INTEGRATING FINGERPRINT BIOMETRICS SYSTEM INTO THE MILITARY POLICE DATABASE: THE CASE OF ZAMBIA ARMY

By

Joseph Kalunga

Thesis submitted to The University of Zambia in fulfillment of the requirement for degree of Master of Engineering in ICT Security

THE UNIVERSITY OF ZAMBIA

LUSAKA

Date 2015

DECLARATION

I, **Joseph Kalunga**, do declare that this dissertation is entirely my own work, except as specified in acknowledgements, and that neither the dissertation nor original work contained herein has been submitted to this or any other institution for the higher degree.

Signed ……

Date……..

Lusaka, Zambia.

NOTICE OF COPYRIGHT

CERTIFICATE OF APPROVAL

This thesis by Joseph Kalunga has been approved as fulfilling the requirements for the award of the Masters of Engineering degree in ICT Security by the University of Zambia.

NAME	SIGNATURE
……	……
Supervisor	
……..……	…….
Internal Examiner 1	
……..	……..
Internal Examiner 2	
……...	……..
External Examiner	

ABSTRACT

This research was conducted so as to investigate the design and development of security applications that integrate fingerprint biometrics system into the military police database for the purpose of improving security in Military organisation. The research opposes manual procedures and modernises military police operations in that fingerprint biometrics authentication systems is incorporated into military police database. Biometric system for human identity authentication is more secure and accuracy as compared to the traditional token based system (access control or Identity Card) method of identification. The study was conducted at Zambia Army Headquarters Military Police Unit in Lusaka Zambia. Prior to the system development, data was collected through unstructured interviews, record inspection and observation. Additionally, other requirements were collected through relevant literature review. After data collection, software requirements were specified; analysed, designed and application was developed using visual studio 2010 on DotNet framework 4.0 with C# object oriented programming language. The backend database used was MySQL relational database management system (RDMBS). The research produced a number of key results including the development of biometric security layer that is able to identify and verify the identity of an individual using enrolled fingerprint template. Other results include the ability to capture service personel data, storage, retrieval and dissemination of information. The developed application performance was evaluated by enrolled ten fingerprints and captured related individual personal information. In each three experiments of enrolled ten fingerprints, the biometric module verified all the thirty fingerprints. Therefore, a biometric error allowance rate of 0.001 percent False Acceptance Rate (FAR) and 0.001 percent False Rejection Rate (FRR) was given. In conclusion, we can describe security as layered and can be improved by introducing biometric human identity authentication. Biometric security layer enhances human identity recognition. Human identity recognition is an essential component of security provision in any organisation. Furthermore, the study shows an integration of fingerprint biometric system into traditional database application is possible and can alleviate problems associated with the manual security procedures.

ACKNOWLEDGEMENT

It is a great pleasure for me to acknowledge the assistance and contributions of many individuals in making this thesis a success.

First and foremost, I would like to thank my supervisor, Dr Simon Tembo, for his assistance, ideas, and feedbacks during the process of writing this thesis. Without his guidance and support, this thesis could not have been completed on time.

Secondly, it is a pleasure to express my thanks to the ARMY COMMAND, the Chief of ICT Brig Gen Mkakangoma, Retired Col Kekelwa, Director IT Col Musonda, The Army Provost Marshal, Maj JB Banda, officers and men who helped me in this research. I deeply appreciate their helpfulness and willingness to provide the useful information for this study.

Lastly, I wish to express my sincere gratitude to my family for their encouragement and moral support.

DEDICATION

To my Mum, Family and friends

Thank you for your encouragement and support.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AFBSMO	Automated fingerprint biometrics system for military organisation.
AFIS	Automated Fingerprint Identification System
ATM	Automated Teller Machine
CCID	Chip Card Interface Device
CIO	Chief Investigation Officer
CN	Crossing Number
COTs	Cost of the Shelf
CTF	Contrast Transfer Function
DARP	Defence Advanced Research Project
DBMS	Database Management System
DLL	Dynamic Link Library
DFDs	Data Flow Diagram
DNA	Deoxyribonucleic Acid
DOD	Department of Defence
ERD	Entity Relationship Diagram
EMV	Euro Pay Master Visa
FA	False Acceptance
FBI	Federal Bureau of Investigation
FR	False Rejection
FZA	MP Form Zambia Army Military Police
GTI	Ground of Tactical Importance
ICT	Information Communication Technology
IDs	Identification Cards
IEDs	Improvised Explosive Devices
K-NN	K-Neighbour Identifier
LDSSA	Long Digital Straight Segment Algorithm
MOD	Ministry of Defence
MP	Military police
MTF	Modulation Transfer Function
NATO	North Atlantic Treaty Organization
NRC	National Registration Card
ODBC	Open Data Connectivity
OCX	Ole Control Extension
OP	Office of the president
OS	Operating System
OpenSSL	Open Source Secure Layer
PIN	Personal Identification Number
POPFNN	Product Based Fuzzy Neural Network
PPI	Pixel per Inch
RGB	Red Green Brown
R&E	Research and development

SDK	System Development Kit
SSL	Secure Sockets Layer
TIS	Top Sharpening Index
SNR	Signal- to-Noise Ratio
TSA	Template Synthesis Algorithm
UML	Unified Modeling Language
SDLC	System Development Methodology
XML	Extensive Mark-up Language

CHAPTER 1. **INTRODUCTION**

1.1 **Background**

Biometrics is an automated method of recognizing a person based on a physiological or behavioural characteristic (Chaudhari, Patnaik, & Patil, 2014). Among the features measured are face, fingerprints, hand geometry, handwriting, iris, retinal, vein, and voice. The term biometrics is derived from the Greek words bios (meaning life) and metron (meaning measurement). Biometrics technologies measure and analyse living human body characteristics for authentication and identification purposes. Any human physiology, chemistry or behaviour can be used as biometric identifier (Chaudhari et.al 2014). However, biometric identifier must poses attributes such as Universality, Uniqueness, Permanence, Measurability, Performance, acceptability and Circumvention (Unar, Seng, & Abbasi, 2014).

The current method for human identity authentication and verification in the Zambia Army is manual. Military Police (MP) uses National Registration Card of the guest to authenticate and verify the true identity of an individual. This approach has security concerns such as impersonation and masquerading as possessing someone else's identity. Biometrics identifiers cannot be easily misplaced, forged, or shared and they are considered to be more reliable for personal recognition than traditional token (e.g. key or card) or knowledge (e.g. password or pin) based method. Additionally, Military police records and employment vetting procedures are manually performed and stored on hard copies. Thus, Manual procedures are associated with a lot of problems such as huge operational cost, technological obsolesce, storage issues, redundancies, inaccuracy, tediousness and labour intensiveness.

With an increase in the use of Information and Communication Technologies (ICT) and global terrorists targeting armed forces citing Boko Haram of Nigeria and Islamic State (ISIL) of Iraq and Syria, there was need to modernise the operation of military police by developing an application that integrates fingerprint biometrics systems into a military police database.

1.2 Aim

The aim of this research was to improve security in the barracks through development and deployment of security applications that integrates fingerprint biometrics into the military police database.

1.3 Description of the Current System

As stated ealier, military police security procedures in Zambia are manual. This entails that human identity identification is done through the usage of code words, service identity cards, national registration cards, driving licence or travelling passports for guests. Service men and women are identified by service IDs which lacks proper security features. Moreover, civilian visitors are required to produce a national registration card for security clearance every time they are visiting military barracks. In the same regard, foreigners may use travelling passports but before that, they must seek national security clearance from the Ministry of Defence (MOD). Furthermore, the current system for office procedures and record keeping is manual. This is to suggest that criminal investigations and analysis of evidence is also done manually. Apart from that, the database for the current serving and retired officers is not in existence. This situation makes it hard for military police to investigate certain cases of fraud and impersonations. Finanally, in terms of employment; criminal vetting procedures for new officers and men is also manual.

1.4 Statements of the Problem

The current system is associated with a lot of problems include: (i) **Cost:** Government is spending huge sums of money on security vetting of new employees joining the system due to manual procedure; (ii) **Technological Obsolete:** The current technologies employed in fingerprinting were old and in some cases obsolete due to introduction of new technologies on the market today; (iii) **Storage Problem:** Vetting forms (FZA MP 12A) have been in use for many years. In case of Zambia, these forms have been accumulated for more than fifty years which have created storage problems; (iv) **Inefficient, Duplication of work, time consuming and Labour intensive:** The current system is deemed inefficient and labour intensive because of high percentage of human intervention, and (v) **Security Concerns:** The current system procedures have a lot of security vulnerabilities in terms of mechanism and

application and (vi) **Identity Verification problem:** the current system has challenges interms of verifying identities of similar personalitities such as resembled and identical twins.

1.5 Motivation

The motivation was derived from the increase in demand of biometrics systems globally and the desire to improve the security situation in the Zambia Army through the development of a military police database that integrates fingerprint biometrics systems into the aforementioned database. Therefore, the developed application is planned to be deployed at sentry posts (gates) and military police headquarters for storage, retrieval and analysis of military police data.

1.6 Research Questions

This research intends to assess the viability of developing computer security application that integrates fingerprint biometrics into military police database for effeciency visitor identity authentication and military police data management. The questions to be answered in this research are: What traditional techniques available for human identity authentication? What success stories in military attributed to this research in the past? What requirements are involved and how are they modelled? What technologies are required in developing an integrated biometric military police database security application?

1.7 Objectives

The specific objectives of the study were to:

a. Understand traditional techniques available for human identity authentication.
b. Describe fingerprint biometrics system and development process.
c. Specify system requirements for integration process of an Automatic Fingerprint Biometrics system into the military police application.
d. Analyse an integrated system requirements and design application models.
e. Develop a C# based security application that integrates a fingerprint biometric system into to military police database.

1.8 Methodology

The study adopted angile software development strategy with system development life cycle (SDLC) methodology. The combination of model includes stages such as requirement gathering, analysis, design, development and product testing for errors. The two models were choosen because of their characteristics centered on clear, fixed and well documented requirements. Additionally, product definations and tehnologies are well known (fingerprint biometrics and c sharp programming language) stable and not dynamic.

1.9 Significancy of the Study

This study is significant because it exploits modalities of infusing modern security technologies in military police procedures and records. The added technology enhances security resilience in terms authenticating and verifying the true identity of visitors accessing sensitive organisation such as Military. The supplementary security layer opposes current manual system. In the current manual systems, National registration card is used as medium to authenticate visitors' identity, but this approach suffers security vulnerabilities mentioned in background to this study. As the result, visitor access control seems to operate autonomously from other military police duties. This development promotes inconsistency and inffecience in discharging military police security duties. Automation of security procedures and functions are therefore, highly recommended.

1.10 Scope of the Study

This thesis covers the software development aspects of fingerprint biometrics systems. It has not included the architecture design of fingerprint sensors and there integration in visual studio libraries as that is provided for in the system development kit (SDK). The developer utilises various algorithms developed by other researchers and modelling them into a computer application using software development techniques.

1.11 Thesis Organisation

This thesis is organised as follows:

1.11.1 **Chapter 1: Introduction**

Chapter one gives the background to the research, defined problem statement, aims, motivation, specific objectives and scope of study.

1.11.2 **Chapter 2: Leterature Review**

Literature review focused on understanding the fundamental details on fingerprinting, technologies, algorithms, parameters and methods of developing the fingerprint biometrics system. The chapter also provided a critique to the existing literature before building the study case.

1.11.3 **Chapter 3: Methodology**

Methodology described the techniques used in the study to achieve the desired results.

1.11.4 **Chapter 4: Analysis**

This chapter provided detailed examination of the elements and structures of the developed system.

1.11.5 **Chapter 5: Design**

This chapter designed architecture, modules and interfaces of the developed system.

1.11.6 **Chapter 6: Development.**

System Development chapter translated the design document into computer language. It involved development of data input forms, reports and routines.

1.11.7 **Chapter 7: Results and Description**

Chapter seven displayed the results of the study.

1.11.8 **Chapter 8: Evaluation**

This evaluated necessity, performance and quality of the developed system.

1.11.9 **Chapter 9: Conclusion and Recommendation**

Chapter nine provided the conclusion and recommendation of the study.

CHAPTER 2. **LITERATURE REVIEW**

2.1 **Introduction**

In this chapter, we will review the literature in an effort to understand the requirements of developing the AFBSMO. We will start with the review of the background information, and then examine the application of biometrics system in military organisations, it thereafter we will review the biological configuration of human fingerprint, approaches to development of a fingerprint biometrics and provide a critique to the existing literature.

2.2 **Background Information**

The background information analyses the biometrics systems with respect to human biological and behavioural characteristics.

2.2.1 **Traditional Identification Techniques**

There are various techniques employed in identifying individuals by both Government and business organisations. Some techniques include the use of national registration cards (NRC's), organisation identification cards (ID's), passports, pin & Chip card, Signature, code words and tickets. These techniques are both token-based (e.g. id card) and knowledge-based (e.g. personal identification number) and they are associated with a lot of security weaknesses (Jain, Hong, & Kulkarni, 1999; Jimoh, Olaniyi, & Adewole, 2011). The reported security vulnerabilities include the £439.4 million UK 2005 financial industry loss attributed to the usage of plastic cards (Apacs, 2006). Other security issues involving traditional mechanisms are failure to authenticate a signature in the foreign country, mutual authentication problems in ticket systems, spoofing, sharable and counterfeiting (Mayes, Markantonakis, & Hancke, 2009). These problems and many others gave birth to smart card technology specifically for financial applications (Apacs, 2006; Boswell, 2009).

Smart cards technology has the ability to provide identity, authentication, data storage, and application processing. This technology has escalated today resulting into a divergence in the application domain. Smart cards have developed into cartel biele (pin and chip system for POS in France), europay paymaster visa (EMV), and contact

less systems, chip card interface device (CCID), USB that allow smart card to connect to the computer, student identification, ticket system, medical care and pay television. Most of it is applicable in pre-paid transactions.

However, the technology has suffered massively in terms of security. There are three categories of security problems on smart Technology (Boswell, 2009). The security vulnerability include: (i) Logical Attacks: exploits that use bugs in the software implementation, (ii) Physical Attacks: exploits that use analysis or modification of the smartcard hardware and (iii) Side Channel Attacks: exploits that use physical phenomena to analyse or modify the smartcard behaviour. These problems most often have resulted in eavesdropping, interruption of service, denial of service and counterfeit (KO & Claytiles, 2011). The vulnerabilities like these ones have paved way for biometric authentication developments (Jimoh et al., 2011).

2.2.2 **Biometrics Systems**

From an evolutionary psychology perspective, humans are biologically similar to animals and each other because they share common ancestors. This view is contrary to the belief of biometrics researchers. Biometrics researchers believe that human biological formations are totally different from each other. Researchers defined biometrics in terms of human behaviour and physiological traits. Biometrics refers to the use of physiological and behavioural characteristics of humans for establishing their identity (Nigam, Vatsa, & Singh, 2015). Any human anatomical or behavioural trait can be used as a biometric identifier provided it poses characteristics which include uniqueness, universality, permanence, collectability, performance, acceptability and circumvention (Jain, Prabhakar, & Pankanti, 2002; Nigam et al., 2015). This view, contradicts the psychological perspective, which differentiate humans by language and brain. Although language and human brain has a behavioural attachment to them, it could be very difficult for scientists to devise the method of identifying an individual using brain and language. However, Chaudhari et,al in his journal of 2014, cited speech as a human behaviour trait in the definition of biometrics system. The most common human behaviour traits used in biometric systems are fingerprint, hand geometry, face, voice, iris, retina, gait palm print and ear (Jain, Nandakumar, & Ross, 2005). Biometrics system can either be based on a single or multiple traits. But a single trait biometrics system is often affected by several

practical problems like noise in sensor, non-universality, lack of distinctiveness in biometric traits, an acceptable error rate and spoof attacks (Fons, & Cantó, 2012; Jain et al., 2005). Additionally, Each biometric system has its benefits and defects in so far as no single biometric can effectively meet all requirements (Chao,Yi-Xian,Xin-Xin, 2006).

2.2.3 **Fingerprint Biometrics**

Among the biometrics technologies, the fingerprint system is the most mature, widely used, acceptable, less costly and considered proof of evidence in the courts of laws all over the world (Jain et al., 2002). Fingerprints are commonly used because their configuration does not change throughout an individual's life time (Chaudhari et al., 2014; Jain et al., 2002). Even when fingerprints temporarily change slightly due to cuts and bruises, they tend to reappear after the healing process (Bhattacharya & Mali, 2013). This makes the fingerprints suitable for human identity verification and authentication (Jain et al., 2002 & Maltoni, Maio,Jain,Prabhakar, 2002). However, the use of fingerprint has remained predominately for police purposes (O'Gorman, 1999). But this will soon change because of a combination of factors that favours the use of fingerprints for the much larger market of personal authentication. These factors include: small and inexpensive fingerprint capture devices, fast and less expensive computing hardware, the explosive growth of network and Internet transactions, increase authentication rate , speed to meet the needs of many applications, and the heightened awareness of the need for ease-of-use essential component of reliable security (O'Gorman, 1999).

The projected increase in utilisation of fingerprint biometrics systems in both security organisations and civil application present some problems. The problems include the costs of building large national fingerprint database, changes in privacy and data protection laws (Maltoni et al, 2002). Other pitfalls include issues of mistaken identity and over dependence on automated identification system. This could possibly happen because people may rely so heavily on fingerprint identification, if fingerprint evidence is not collected, stored, or handled properly it may result in a false identification which people will believe is valid for the reason that they view fingerprinting as highly reliable (Cai, Lu & Liu, 2007; Cappelli, Ferrara, & Maltoni, 2015).

2.2.4 **Biometrics Deployed with other Security Mechanism**

Because of the problems associated with single identifier biometrics systems, some researchers have proposed the employment of multimode biometrics traits and others have proposed for the combination of the biometrics with traditional security mechanisms. Multimodal biometrics systems are more robust than unimodal biometrics (Ak Jain & Kumar, 2010; Zhu & Zhang, 2010). Multimode biometrics systems take advantage of multiple traits to improve the performance in many aspects including accuracy, noise resistance, universality, spoof attacks and reduced performance degradation in large databases (Zhu & Zhang, 2010). Additionally, biometrics has been employed in police forensic references where fingerprints are identified against criminal record in the process called criminal vetting (Lewis, 2014). In travelling industry, biometrics have been used with smart chips embedded passport and facial image(Bolle, Senior, Ratha, & Pankanti, 2006; Lewis, 2014; Y. Liu, 2011). However, even the security system with all those trait combination can be breached somehow (Boswell, 2009; Breebaart, Buhan, De Groot, & Kelkboom, 2011; Martinez-Diaz, Fierrez, Galbally, & Ortega-Garcia, 2011).

2.2.5 **Related Work**

The deployment of a fingerprint system is not new to security organisations. Since the year 1960, police forces have used fingerprints for security vetting and forensic investigation. Many are times when police collect fingerprint impressions from the crime site and taken them to Universities or forensics labs for enhancement and comparison with a National fingerprint database for criminal identification. This technology and many other advanced security tools were developed in the military and then off-loaded to the police for enhancement of urban public safety (Nunn, 2001). Some other governments have developed standard security application for their law enforcement organisations in particular Canada which has developed MorphoBis, the next generation fingerprint biometric system for Calgary Police Service and Edmonton Police Service (Caldwel, 2011).

2.2.5.1 **Biometric System in Military**

Biometric applications have been deployed in military both in peace and war times. The following were success stories of biometric system in military: (i) Identifying

Non-Military Personnel: The US, UK Ministry of Defence and NATO forces used biometrics systems to identify non-military personnel in countries where their troops have been deployed(Gold, 2010), (ii) Identify Prisoners: US Army used biometrics system to identify prisoners of war captured in Iraq and Afghanistan (Gold, 2010; Mansfield-Devine, 2012, 2012; Nunn, 2001), (iii) Actionable Intelligence: Biometrics has been used to detect enemy forces infiltrating ground of Tactical Importance (GTI). In other applications, fingerprints taken from the remains of detonated Improvised Explosive Devices (IEDs) has not only resulted in identifying the bomb maker or person who planted it but, through cross-correlation with other intelligence information, this has led to an understanding of the structure of Insurgents cells(Gold, 2010;Mansfield-Devine, 2012 & Nguyen, 2015), (iv)Tracking Terrorist: The Loyal Netherlands and Denmark Army used biometric in Afghanistan for tracking terrorists and US Government have deployed biometric system at national airports world-wide for the same purpose (Gold, 2014; ("Law enforcement agencies tap into biometrics," 2012; Nguyen, 2015)),(v) Identify Slain Terrorist: US Army deployed face biometrics when identifying Osama bin laden killed in Pakistani (Biometric Technology Today, May 2011),(vi) Biometrics Identity Card:US Marine used faces and fingerprint traits to develop Identity Cards for service men and civilians working for them in Afghanistan and Iraq. In the same vain, the Afghan Government has Introduced biometrics Identity Cards for police and the military (Gold, 2010; Mansfield-Devine, 2012) and,(vii) Protection of Military facilities: Biometrics has been deployed in South East- Asia to protect US Army facilities (Gold, 2010; Mansfield-Devine, 2012). Indian Army also has deployed biometric system to protect their national border (Adams, 2000; Mansfield-Devine, 2012)

2.2.5.2 **Security Vetting Process**

Security vetting is one of the methods security wing employees to ascertain the true identity of applicant enlisted for national service (Prisons, 2014). Vetting process may be conducted for employment, criminal record information, security service record and financial irregularity assessments. In military this role is performed by the police. Automate vetting system process was conducted in Australia, New Zealand government and United States department of homeland security. In new land security vetting is conducted by national Intelligency agency. The need for security vetting in

three nations was feasibly an anti-terrorism measures. This was to ensure that right people were employed to handle classified information (Register et al., 2000).

2.2.5.3 **Military-To-Civilian Technology Transfer**

Most advanced security technologies used in police and civilian applications such as the one installed at national airports world-wide originate from military. The Defence Advanced Research Project Agency (DARPA) and US department of Defence (DOD) developed technologies such as photonics, thermo imaging, and behavioural recognition system like DNA testing which originated from the military under military-to-civilian technology transfer (Nunn, 2001).

2.2.5.4 **Challenges**

The main challenges many Military Establishments are facing in the implementation of biometric systems include privacy issues, mobility of enemies, cost and legal challenges (Gold, 2014).

2.2.6 **Anatomy of Fingerprint**

Fingerprints are the skin patterns formed on the epidermis of the fingertip. These skin patterns are sometimes referred to as fingerprint features and are represented by the collection of ridges and valley on the surface of the fingertip (Hong & Wan, 1998; Nagaty, 2005). Ridges and valleys are broken further into other features called minutiae. Minutiae are the discontinuities in local ridge structure and they are used by investigators to match two fingerprints (Chaudhari et al., 2014; Nagaty, 2005). Scientists discovered about 150 different types of minutiae in the fingerprint (Chaudhari et al., 2014). Among these minutiae types "ridge ending" and "ridge bifurcation" are the most commonly used as all the other types of minutiae are combinations of ridge endings and ridge bifurcations. A ridge ending is defined as the ridge point where a ridge ends abruptly and A ridge bi-furcation is defined as the ridge point where a ridge forks or diverges into branch ridges (Chaohong, 2007, young, 2008 & Maltoni et al, 2003). Researchers' have further classified fingerprint features into secondary classification and Henley classification. Fingerprint classification is important in large databases to simplify searching process (Awad & Baba, 2012; Prabhakar, 2001). Figure 1 Shows fingerprint minutiae type:

Figure 1: Fingerprint Minutiae Type

Source: (Chaudhari et.al 2014)

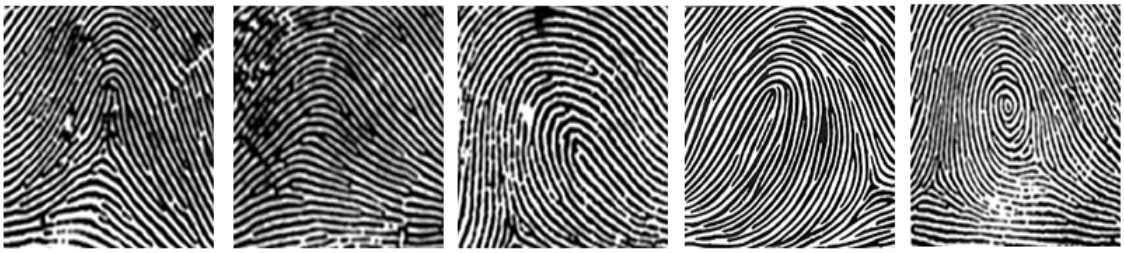
2.2.6.1 **Fingerprint Classification**

Fingerprint features are organised into three levels, level 1 to level 3 (Cappelli & Ferrara, 2012; Chaudhari et al., 2014; F. Liu, Zhao, & Zhang, 2011; Vatsa, Singh, Noore, & Morris, 2011). These levels include: (i) Level-1(global) features: Level 1 feature refers to the global ridge orientations and singular points. This representation relies on the ridge structure, global landmarks and ridge pattern characteristics. The commonly used global fingerprint features are singular points which are discontinuities in the orientation field (Bazen, 2002). There are two types of singular points and these are core and delta (Cao, Pang, Liang, & Tian, 2013; Cappelli &Ferrara, 2012). A core is the uppermost of the innermost curving ridge and a delta point is the junction point where three ridge flows meet. In the Phd thesis, Chaohong, 2007 revealed that fingerprint core and delta are usually used for fingerprint registration and classification in the biometric systems, (ii) Level-2 (local): Local feature includes minutiae details extracted from the ridge skeleton. Level 2 features include ridge ending, ridge bifurcation, bridge and ridge close. Level 2 feature are used for fingerprint matching (Cao et al., 2013; Cappelli & Ferrara, 2012), and Level 3 (Fine-detail) includes intra-ridge details such as width, shape, ridge contours, sweat pores, and creases. Based on the position on the ridges, pores are often divided into two categories: open and closed. A closed pore is entirely enclosed by a ridge, while

an open pore intersects with the valley lying between two ridges. Level3 features are used for matching the templates (Galar, M. et al 2015).

2.2.6.2 **Henley Classification**

Most Fingerprint classification techniques are based on Galton– Henry classification scheme as shown in Figure 2. Henry classification scheme categorizes fingerprint into five major classes which consist of: (i) Arch: Fingerprints that have no Singular Points and ridges flow from one side to the other forming a small bump. In an arche the ridges run from one side to other, making no backwards turns; there is ordinary no delta, but when they is appearance of the delta, no ridge must intervene between the "Inner" and "Outer terminus", (ii) Tented Arch: Fingerprints that have one core and one delta (the delta is under the core), and ridges flows similarly to Arch, but at least one ridge shows a high curvature. The ridge near the middle may have an upwards thrust, arranging themselves as it were on both sides of a spine or axis towards which adjoining ridges coverage, (iii) Right Loop: Fingerprints that have one core and one delta (the delta is under and to the left of the core), and one or more ridges flows from the right side curve back and go out from the same side. In the loop, some of the ridges make backwards turns but without twist. There is one delta, (iv) Left Loop: Fingerprints that have one core and one delta (the delta is under and to the right of the core), and one or more ridges flows from the left side curve back and go out from the same side, and (V) Whorl: Fingerprints that have two cores and two deltas, and at least one of its ridges make a full turn around the centre of the fingerprint. In the whorl, some of the ridges make a turn through at least one complete circuit and there are two deltas. Whorls may contain single or double core (Prabhakar, 2001; Rajanna, Erol, & Bebis, 2010).



(a) (b) (c) (d) (e)

Figure 2: Henley fingerprint classification: (a) Tended Arch, (b) Arch, (c) Right Loop, (d) Left Loop, (e) whorl

Source: Henley, 1900

2.3 **Design and Development**

An important issue in designing a practical *biometric system* is to determine how an individual is going to be recognized. Depending on the application context, a biometric system may be called either a *verification* system or an *identification* system. A verification system authenticates a person's identity by comparing the captured biometric characteristic with her previously captured (enrolled) biometric reference template pre-stored in the system. It conducts one-to-one comparison to confirm whether the claim of identity by an individual is true. The fingerprint verification system can either reject or accept the claimed identity. An identification system recognizes an individual by searching the entire enrolment template database for a match. It conducts one-to-many comparisons to establish if an individual is present in the database and on correct event, returns an identifier of the enrolment reference that match. In most identification system, an application establishes a subject's identity and make pronouncement of the finding (Maltoni, 2005 & Maltoni et al, 2009).

2.3.1 **System Modules**

Biometric system is a computer application like any other specialised application. It is defined as a set of component, objects or modules that interact for the common purpose (Senn, 2009). Any fingerprint biometrics must include the following modules: (i) Fingerprint Capture: a digital representation of biometric characteristic needs to be sensed and captured. A biometric sensor, such as a fingerprint scanner, is one of the central pieces of a biometric capture module, (ii) Feature extraction: in order to facilitate matching or comparison, the raw digital representation (sample) is usually further processed by the feature extractor to generate a compact but expressive representation, called a feature set, (iii) Template creation: the template creation module organizes one or more feature sets into an enrollment template that will be saved in some persistent storage. The enrollment template is sometimes also referred to as a reference, (iv) Pre-selection and matching: the pre-selection (or filtering) stage is primarily used in an identification system when the number of enrolled template is large. Its role is to reduce the effective size of the template database so that the input is matched to a relatively small number. The matching stage also know as a matcher takes a feature set and an enrollment template as inputs and computes the similarity

between them in terms of a matching score, also known as similarity score. The matching score is compared using system threshold to make the final decision. If the match score is higher than the threshold, the person is recognized, otherwise not and, (v) Data storage: data sorage is dedicated for templates storage and other demographic information. Depending on the application, the template may be stored on either internal or external storage devices (Yoon 2014, pp.16; Maltoni, 2005; Prabhakar, 2001).

2.3.2 **Approach to Development**

The fingerprint biometric system could be developed using one or combination of the following methods:

2.3.2.1 **Structural Approach**

Structural approach concept use operators to recognise ridge characteristics and impart knowledge in the computer system to manipulate and compare fingerprints in order match stored parameters with that being extracted from the fingerprint scanner. Fingerprint patterns are extracted and matched using terminal symbols and production rules (Kumar & Wu, 2012; Lumini, Maio, & Maltoni, 1997).

2.3.2.2 **Ridge Orientation Approach**

Ridge orientation method is based on calculation of orientation field of fingerprint minutiae. Feature enhancement algorithms are applied using image processing techniques to enhance ridge fingerprint structures. Ridge orientation approach is minutiae based method and is characterised by good quality image. Minutiae based fingerprint biometrics require good quality image because it is so sensitive to noise. Ridge orientation approach parameters include ridge count, ridge length, ridge curvature direction and ridge type (Maltoni, 2005).

2.3.2.3 **Pixel-Level Approach**

The pixel-level approach uses the fact that fingerprint is made up with white and black bounded number of pixels. The parameters are the widths of black and white boundary of scanned fingerprint image (Lin, Chen, & Tseng, 2011).

2.3.2.4 **Filtering and Wavelet Approach**

Filtering and wavelet development approach is a one-step method using Gabor filters for directly extracting fingerprint features for a small-scale fingerprint recognition system. The method uses k-NN classifiers. Block wise Fourier transform is multiplied by its power spectrum raised to a power, thus magnifying the dominant orientation (Montoya Zegarra et al., 2009).

2.3.2.5 **Geometric Approach**

Geometric Techniques is the type of minutiae fingerprint recognition system, in which the fingerprint's orientation field is reconstructed from minutiae and further utilized in the matching stage to enhance the system's performance and accuracy matching levels. Each minutia is defined by the type, relative topological in relationship with other minutiae using nearest neighbour concept (Montoya Zegarra et al., 2009; Rajanna et al., 2010).

2.3.2.6 **Singularity Approach**

Singularity fingerprint recognition development approach is based on core and delta singularity point detection. The singular point extraction is performed using three sequential steps: directional image extraction, Poincare indexes computation and core and delta extraction. The approach has shown a good accuracy level in the singularity points detection and extraction and a low computational cost (Danese, Giachero, Leporati, & Nazzicari, 2011).

2.3.3 **Design and Development Parameters**

A fingerprint biometrics system design is centred on digital fingerprint image processing concepts.The main parameters characterizing a digital fingerprint image are as follows.

2.3.3.1 **Rectangular Area**

The size of the rectangular area sensed by a fingerprint scanner is a fundamental parameter. The larger the rectangular area becomes, the more ridges and valleys features captured. Furthermore, the more distinctiveness a fingerprint image and template becomes. A rectagular area greater than or equal to 1 × 1 a square inches is prescribed by FBI as standard specifications and thus permits a full plain fingerprint

impression. In most recent fingerprint scanners, rectangular area is sacrificed to reduce on cost. However, a smaller area scanner do not allow whole fingerprint to be captured, and users may encounter difficulties in representing the similar portion of the finger. This may result in a small overlap between different acquisitions of the same finger, leading to false acceptace and non-matching errors. (Moltoni et al, 2009& Liu et al., 2011).

2.3.3.2 **Resolution**

Resolution indicates the number of dots or pixels per inch (dpi). 500 dpi is the minimum resolution for FBI-compliant scanners and is met by many commercial devices; 250 to 300 dpi is probably the minimum resolution that allows the extraction algorithms to locate the minutiae in fingerprint patterns (Hong, & Wan, 1998 & Maltoni, 2005).

2.3.3.3 **Number of Pixels**

The number of pixels in a fingerprint image can be simply derived by the resolution and the fingerprint area: a scanner working at R dpi over an area of *Height (h)* $\times$*Width (W) inches*^{2} *has R.h×R.W pixels* (Maltoni, 2005). If the area is expresses in mm^2, the formulae must include mm to inch conversions and, therefore, the number of pixels may be computed as given in equation 1.

$$\textit{No of Pixels} = \quad R.\left(\frac{h}{25.4}\right) \times R.\left(\frac{w}{25.4}\right) \ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

2.3.3.4 **Geometric Accuracy**

This is usually determined by the maximum geometric distortion introduced by the acquisition device. The geometric distortion can be measured as the absolute value of the difference between the actual distance between two points on a calibrated target and the distance between those same two points as measured on the scanned image of that target. Some of the optical fingerprint scanners introduce geometric distortion which, if not compensated, alters the fingerprint pattern depending on the relative position of the finger on the sensor surface (Holder & Robinson, 2010; Maltoni, 2005).

2.3.3.5 **Gray-level Quantization and Gray Range**

This denotes the number of bits used to encode the intensity value of each pixel. The FBI standard for pixel bit depth is 8 bits, which yields 256 levels of gray image (Maltoni, 2005).

2.3.3.6 **Spatial Frequency Response**

The spatial frequency response denotes the ability of an acquisition device to transfer the details of the original pattern to the output image for different frequencies. It is well-know that the fine details corresponding to the high frequencies tend to be smoothed out when a signal is digitally sampled. Spatial frequency response is usually measured through Modulation Transfer Function (MTF) or Contrast Transfer Function (CTF) as explained in Nill (2005); a specific measure for fingerprint scanners, called Top Sharpening Index (TSI), was introduced by Ferrara, Franco, and Maltoni (2007).

2.3.3.7 **Image Quality**

It is not easy to precisely define the quality of a fingerprint image, and it is even more difficult to decouple the fingerprint image quality from the intrinsic finger quality or status. In fact, when the ridge prominence is very low (especially for manual workers and elderly people), when the fingers are too moist or too dry, or when they are incorrectly presented, most of the scanners produce poor quality images (Maltoni, 2005)

2.3.3.8 **Signal-to-Noise Ratio (SNR)**

The signal to noise ratio quantifies the magnitude of the noise with respect to the magnitude of the signal. The signal magnitude is related to the gray-range in the output image while the noise can be defined as the standard deviation of the gray-levels in uniform gray patches.The above described parameters and their specification are designed fingerprint standards for American security intelligence body Federal Beraul of intelligency (FBI) as extracted from fingerprint biometrics handbook (Moltoni el al, 2009). For complete reference table of FBI Recommendation check appendix D table D.2.

2.4 **Feature Enhancement**

The performance of minutiae extraction algorithms and other fingerprint recognition techniques relies heavily on the quality of the input fingerprint images. In an ideal fingerprint image, ridges and valleys alternate and flow in a locally constant direction. In such situations, the ridges can be easily detected and minutiae can be precisely located in the image. However, in practice, due to skin conditions (e.g., wet or dry, cuts, and bruises), sensor noise, incorrect finger pressure, and inherently low-quality fingers (e.g., elderly people, manual workers), a significant percentage of fingerprint images approximately 10 percentage, according to experience is of poor quality (Jain et.al 2006).

2.4.1 **Pixel-wise Enhancement**

In pixel-wise image processing operation, the new value of each pixel only depends on its previous value and some global parameters (but not on the value of the neighboring pixels). Pixel-wise techniques do not produce satisfying and definitive results for fingerprint image enhancement. However, contrast stretching, histogram manipulation, normalization and Wiener filtering have been shown to be effective as initial processing steps in a more sophisticated fingerprint enhancement algorithm. The normalization approach determines the new intensity value of each pixel in an image as illustrated in the formular below:

$$\mathbf{I}'[x,y] = \begin{cases} m_0 + \sqrt{(\mathbf{I}[x,y]-m)^2 \cdot v_0 / v} & \text{if } \mathbf{I}[x,y] > m \\ m_0 - \sqrt{(\mathbf{I}[x,y]-m)^2 \cdot v_0 / v} & \text{otherwise,} \end{cases} \quad \text{..... (2)}$$

Where*m* and *v* are the image mean and variance and m_0 and v_0 are the desired mean and variance after the normalization (Hong, Wan& Jain, 1998, 2000 and Jain et.al 2006). Figure 3 (a) shows original image which is in enhance in Figure 3 (b)

(a) **(b)**

Figure 3: Fingerprint Image (a) Original (b) Enhanced
Source: Jain et.al, 2006

2.4.2 **Contextual Filtering**

The most widely used technique for fingerprint image enhancement is based on *contextual filters*. In conventional image filtering, only a single filter is used for convolution throughout the image. In contextual filtering, the filter characteristics change according to the local context. Usually, a set of filters is pre-computed and one of them is selected for each image region. In fingerprint enhancement, the context is often defined by the local ridge orientation and local ridge frequency. In fact, the sinusoidal-shaped wave of ridges and valleys is mainly defined by a local orientation and frequency that varies slowly across the fingerprint area. An appropriate filter that is tuned to the local ridge frequency and orientation can efficiently remove undesired noise and preserve the true ridge and valley structure. Several types of contextual filters have been proposed in the literature for fingerprint enhancement. Although they have different definitions, the intended behavior is almost the same and includes: (1) provides a low-pass (averaging) effect along the ridge direction with the aim of linking small gaps and filling impurities due to pores or noise; (2) performs a bandpass (differentiating) effect in the direction orthogonal to the ridges to increase the discrimination between ridges and valleys and to separate parallel linked ridges (Cappelli & Ferrara, 2012; Jain et.al, 2006 ; Khalil, Mohamad, Khan, & Al-Nuzaili, 2010).

2.4.3 **Minutiae Based Fingerprint Matching**

Based on the features used in fingerprint matching, most existing algorithms can be classified into two categories: minutiae-based approaches and global feature-based approaches. It is widely believed that minutiae are the most discriminating and

reliable features in fingerprints. Many matching methods based on minutiae have been proposed (Maltoni et al, 2009). Since the relative transformation between two fingerprints is unknown in advance, the correspondence between minutiae is very ambiguous. Many researchers have tried to attach local features to minutiae to reduce ambiguity. These local features include ridge information (Jain et al., 1997a; He et al., 2003), local orientation features sampled around the minutiae (Tico and Kuosmanen, 2000) and local minutiae structure features (Chen et al., 2006; Ratha et al., 2000). Recently, He et al. (2006) proposed a global comprehensive similarity-based fingerprint matching algorithm, in which a minutiae-simplex, including a pair of minutiae as well as their associated textures, were employed to achieve fingerprint matching. This approach represent fingerprint as a graph, in which the comprehensive minutiae acted as the vertex set and the local binary minutiae relations were used to provide the edge set. Some researchers combined other local features to increase the discriminative ability between minutiae (Cao et.al 2012).

2.5 **Critique**

In conclusion, the researcher critically evaluated the weaknesses, strength, logical links and practical implementation of theories in existing literature. The following were discovered:

- Fingerprint Biometric system design lack prescribe standard. This was evident in many algorithm that researcher have recommended.
- According to the definition of biometric system which is defined as "Automated method for verifying or identifying living individuals based on physiological or behavioural characteristics "(Abhyankar & Schuckers, 2009). This definition is contrary to what the author reported in biometric today of May, 2011 where it stated "The US Military used face biometrics recognition to identify slain al-Qaida leader Osman bin Laden". Biometric system does not apply to dead human beings.
- Biometrics systems developments forgo privacy and legal public concern.
- The literature is silent on the provisions for disabled people, people who have lost two hands. In the developed system, this requirement is taken care of using the national registration card details and passport size photo of the disabled.

- The literature is also silent on legal framework governing the development of biometric system.

2.6 Summary

This chapter reviewed an existing literature in order to understand integration process of fingerprint identity authentication system into military police database. Basically, biometrics data (literature) was reviewed with the view to learn process of developing a biometric fingerprint system which was required in an integration process. Various image processing techniques were highlighted and reviewed. Literature critique was done from which the case was built. The built case was to develop a computer application that integrate military police database with fingerprint biometrics authentication system. The next chapter covers methodology and material used in the study.

CHAPTER 3. **METHODOLOGY**

3.1 **Introduction**

Methodology chapter refers to the strartegies that is used to structure, plan, and control the process of developing an integrated fingerprint biometrics system for military organisation. There are wide variety of such strategies and have evolved over the years of which each of them has its own recognized strengths and weaknesses. In this study, we have used angile method with water fall system development life cycle methodology. Angile is used in this research because the requirements and solutions of developed system evolve through collaboration between self- organizing cross-functional teams.

3.2 **Study Area**

The research was conducted at Zambia Army Headquarters Military Police Unit department of administration branch. For research timeline and project schedule refer to appendix section B.

3.3 **Study Population**

The target population included the provost marshall, military police officers, military intelligence officers, sentries and military legal officers.

3.4 **Research Strategy**

The research strategy for understanding the integration process of fingerprint biometrics system into military police database is action research. The action research strategy is chosen because it is centred on the problem solving in an organisation where the researcher operates or works. The key to understand action research is to realise from onset that the researcher is involved in the research not just as a research observer but as participant (Biggam, 2008). This research approach provides better understanding of system requirements than any other methods since the researcher knows some of the procedures in an organisation. On top of information that the researcher already knows or has, other sources such as military police records were inspected. For details on these forms check appendix section C.1 and C.2. Literature

review was also used to acquire information on the developed fingerprint biometric system.

3.5 Data Collection

Data collection was conducted to provide the solution for objective 1 to 4 of our study and the following methods were employed:

3.5.1 Unstructured Interview

Unstructured interviews were also used in our action research. An unstructured interview is conversation without specific format. This method was employed for security reasons as we were dealing with military information which is security classified in nature. In unstructured interviews, respondents may be more likely to discuss sensitive and painful experiences if they feel the interviewer is sympathetic and understanding (Senn, 1989:115-116). However, the results of unstructured interviews may be misleading because some respondents may not be the rightful people to give certain information. Nevertheless, unstructured interview was a primary source of information and was tactically executed so as to obtain relevant information. The following were conducted:

a. Step one made enquiries about organisation security rules, standing procedures and laws.
b. Step two made enquiries about organisation identification process of visitors, dependants and military personnel in plain clothes.
c. Step three made an enquiry on various problems military police officers were facing in provision of security, enforcement of law and order.
d. Step four made consultative enquiry on problems faced in areas of record keeping.
e. Step five provided discussion on how military police conducted operations with an intention to learn processes, key player and statistics of products.
f. Step six asked about departments, roles and duties each actor was performing.
g. Step seven discussed service identity cards
h. Step eight asked about new measures for improving security in barracks.

3.5.2 **Record Inspection**

Record inspection was conducted which included the review of written policy manual, government forms, regulations, and standard operational procedures used in organisation as guide for officers and men. This method was employed to understand the framework or rather operating procedures for the military police. Record inspection was used because it provides more operation details than any other methods (senn, 1998: 117-118). In record inspection the following were conducted:

a. Organisation security rule, standing operational orders and roles.
b. Document appointments and roles.
c. Disciplinary record.
d. Filling system.
e. Government security forms.
f. Manual man number registers.

3.5.3 **Observation**

Observation data collection method was used to learn and verify phenomena procedures and operation of the military police. The main strength of observation is that it provides direct access to the social occurrences under consideration. Instead of depending on some kind of self-report, such as asking people what they would do in a certain situations, you actually observe and record their behaviour in that situation. This method was used to capture classified information. However, the disadvantage of observation method is that it is extremely time consuming and resource intensive (Senn, 1998: 119-120). During observation method, particular attention was paid to the following items in order of security importance:

a. **Step one;** observed organisation security rules, standing procedures and laws.
b. **Step two;** observed identification process of visitors, dependants and military personnel in plane cloths.
c. **Step three;** observed the process of conducting criminal vetting for new employees.
d. **Step four;** observed how military police conducted operations with an intention to identify key players.
e. **Step five;** observed the roles, departments and duties of each actor.

f. **Step six;** observed security techniques or mechanisms (forms of identification).
g. **Step seven;** observed how records were managed at military police headquarters.

3.5.4 **Approaches for Literature Review**

Literature review was also used as a data collection method. It focused on understanding the development aspect of fingerprint biometrics system for military organisation. This covers review of background information, the employment of fingerprint biometrics in security organisation, reviewed the physiological configuration of fingerprint biometrics systems, specified fingerprint image parameters as specified by FBI, reviewed Techniques and methods used in feature detection and orientation field. Additionally, literature review covered methods for development of an automated fingerprint verification system and provide the critique of existing literature.

3.6 **Software Development Approach (Overall Methodology)**

The research adopted traditional System Development Life Cycle (SDLC) methodology. The SDLC model is a sequential design process, used in software development in which advancement is seen as walked progressively through system development phases (Senn, 1999). SDLC stages include investigation, analysis, design, development, evaluation and maintenance. The SDLC methodology was used because it is easy to measure the progression of the project and the model gives clear system objectives. However, SDLC methodology suffers from backward incompatibility and difficult to respond to changes (senn, 1998 and Langer, 2008).

3.6.1 **Approach for Investigation**

The existing system was studied and evaluated. Prior to detailed investigation, a feasibility study was conducted aimed at exploring available technologies (technical feasibility), costs (economic feasibility) and operational issues (operational feasibility). Tools employed included unstructured interviews, observation and record inspection as described in data collection methodologies above. The output was user requirement elicitation.

3.6.2 **Approach for Analysis**

Analysis in this study involved overall system description which included product view, project philosophy and description of military police procedures. The section also analysed architectural design of the new system which include biometrics and security vetting system. Analysis in a nutshell detailed the requirement engineering aspect of the research. Requirement engineering phase included problem domain definition using rich picture, behaviour modelling; using use case diagrams, requirement elicitation; using derived requirement model that resulted in functional and non-functional requirements specification. The process modelling was conducted using activity diagrams. Ultimately, analysis stage gives way to system design stage. Figure 4 shows a system development life cycle (SDLC).

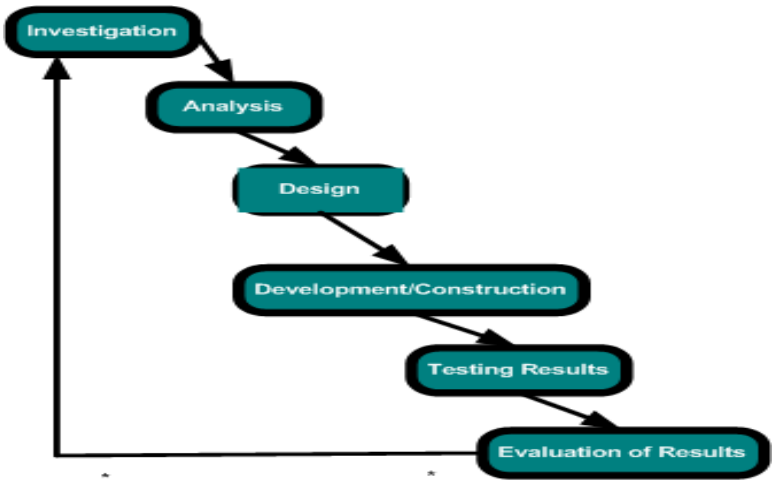


Figure 4: System Development Life Cycle Methodology

3.6.3 **Approaches for Design**

The system design describes the operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, processing logic, and external interfaces of AFBSMO. The data processing model employed during database design stage included schema diagram and entity relational ship diagram, While class diagram was used for business logic or application design. The design was performed in the following manner:

a. Step one- hardware design.
b. Step two- software designs.

c. Step three- Process design.

3.6.4 **Approaches for Development /Construction**

The Construction and Assembly phase focused on transforming the design into a working system that satisfies all the requirements. Any special procedures for data conversion and/or data warehousing was also be developed and tested during this phase. Various technological implementations had been explained in this stage. Application developments were executed in the following order:

a. **Step One** Software Installation:
    - Installation of visual studio.
    - Installation of DBMS (WAMP Server).
    - Installation of middleware program
b. **Step Two** Database Development:
    - Database creation.
    - Creation of table.
    - Creation of constraints and relationships
c. **Step Three** Application Development
    - Creation of Data Entry forms.
    - Coding
    - Creation of reports and coding
    - Debugging

3.6.5 **Approach for Testing**

System testing is categorised into two software and hardware testing. Software or hardware testing was conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. The testing was conducted in the following order:

a. Unit testing.
b. Modular testing.
c. Application testing.

3.6.6 **Approach for Software Evaluation**

Evaluation stage assessed the necessity, quality and matters addressed by the developed product. The aim of system evaluation was to collect information about the developed product as a whole, performance and any other details that are helpful in guiding success of the developed application.

3.6.7 **Approach for determining Main Module**

The approach for determining main module involved understanding the main components of the project and grouping them into modules (i.e. identity verification and military police modules). The modular application development approach was adopted. An understanding of the relationships and inter-modular communication was important. Inter-modular data flows were elaborated by procedures, Data Flow Diagrams (DFDs), class diagram and database schemas diagram.

3.7 **Approach for Biometric Fingerprint Image Processing**

The fingerprint image processing methodology is based on Template Synthesis and includes stages such as image capturing, Image normalisation, Segmentation, threshold value Calculations, Application of Gabor filter, Image binarisation, Thinning, Minutiae extraction, Template generation and apply matching algorithm as shown in Figure 5. For further reading on minutiae based fingerprint biometrics system check (Chaudhari et al., 2014).

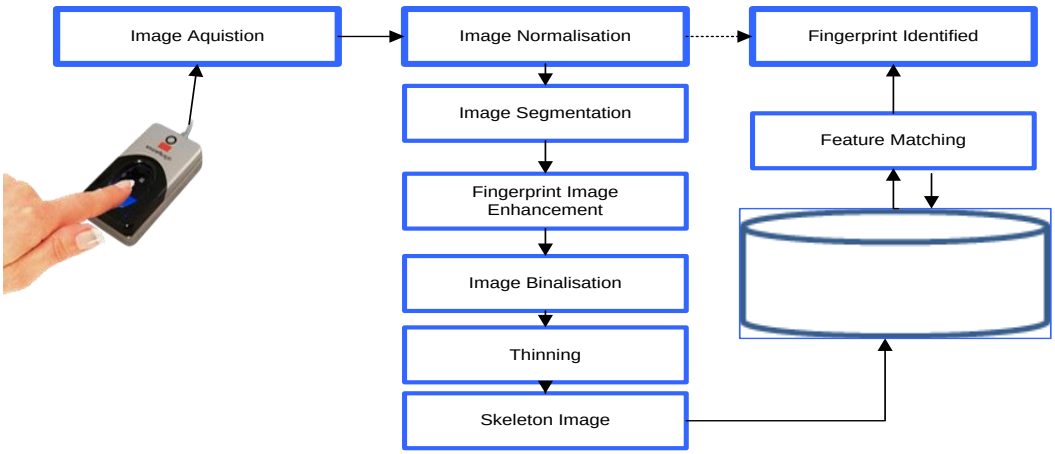


Figure 5:Fingerprint Image Processing Methodology

3.8 **Approaches for Database Development**

Database development approach involved synthetic break down of the requirements into relation. A relation consists of one or more attributes. An attribute is a property of an objector relation type. Relations were created as a data storage structure in the relational database management system (DBMS) and relationship was created through the foreign key attribute. Relations were grouped together to form a schema.

3.9 **Materials**

Materials used in this study are categorized into hardware and software's. Hardware category involves physical components such as fingerprint sensor and digital camera. Software components include visual studio, SDK kit and other software applications. The design specification of both hardware and software are given in table 1 below. For detailed specification and costing check appendix section B.

Table 1: Development Tools Design Specifications

Name of the Requirement	Specification
Hardware - Laptop	- Hard disk 80GB Minimum - Processor 3.2 MHz Minimum. Preferable core i3 and above - RAM 4.0 GB - Graphics frequency 3.30 MHz - 64-bit Operating System
Fingerprint Scanner	- Image resolution 500 pixels per inch. - Image area 9.75mm X 0.41mm/ 192 X 8 pixel - ISO / IEC 7816 T=0 and T=1 - Up to 8Mhz smart cards, and a 412 Kbit/s communication speed
Digital Camera	- 18.0 Megapixels - 18-55mm lenses - Speed 3frs - Full-high definition
Software - Visual Studio	Version 2010
- Wamp Server	Version 2.2
- Microsoft Project	Version 2003
- Microsoft Visual	Version 2003

3.10 **Summary**

This chapter discussed research approaches used in this study. Theses approaches include the data collection, analysis and system development methods. Furthermore, the chapter tabulated various materials and their specifications used in the study.

CHAPTER 4. **ANALYSIS**

4.1 Introduction

Analysis Chapter determined system requirements for application development. The heart of system analysis is a detailed understanding of all important facets of system under study. System analysis is defined as the software development phase that provides detailed examination of the elements or structure of new and old system. System structures include requirements and their stipulated environmental conditions under which they operate. Requirements are categorised into business, architectural and integration aspect.

4.2 Fact Finding Results

A lot of problems were highlighted as a result of the investigation (fact finding) conducted. Most of it was security related concern. The security concerns were compounded by the global increase in terrorism activities targeting security forces, and the desire to computerise military police operations aimed at improving record keeping, visitor authentication and fingerprint evidence processing.

4.2.1 **Problems of the Current System**

The current military police procedures in the Zambia Army are manual and therefore, the organisation was facing the following difficulties:

4.2.1.1 **Huge Operation Cost for new Employees Criminal Vetting**

The current security vetting process of new employee is costly in terms of material requirements and time. It demands for the large usage of paper, ink and stamp. Additionally, the current process is dirty and slow because of procedure. Criminal vetting process is currently conducted by military police in conjuction with civil police. Civil police is collaborated in this exercise because they hold national criminal record register. The national criminal record register is manual and hence costly in terms of time to complete vetting exercise.

4.2.1.2 **Technological Obsolete**

The current technologies employed in fingerprinting are old and in some cases may be declared obsolete because of current introduction of new technologies on the market.

For example, the vetting process was being conducted using ink, stamp and prescribed government forms. Fingerprint evidence analysis was being done by human comparison of fingerprint features between the evidence print and the actual print taken from a suspect for that purpose. Some of the technologies being used are old and are not mentioned here for national security reasons. Additionally, there was a concern of technological obsolete of some items being used by military police because manufacturers had stopped production due to the new innovation or creeping into the market demand.

4.2.1.3 **Storage Problem**

Storage problems were also reported due to large volume of papers which was being used. It was established that a lot of files were created and the documents were kept properly in those files. Military organisation is much organised in terms of record keeping but the study discovered that the organisation was overwhelmed with the storage space. The filling system was traced way back to 1964. The oldest file was in February 1964. Some files were taken to archive. A classic example is vetting form (FZA 12A) check appendix section B.

4.2.1.4 **Labour Intensive**

Labour intensity is high in any manual system. The current system was deemed inefficient and labour intensive because of high percentage of human intervention. Human intervation resulted in Inefficient, Duplication of work, time consuming and Labour intensive. It was also discovered that manual intervention resulted in resource mismanagement in terms of human deployments. A lot of people were attached to a job which can be done by a single person.

4.2.1.5 **Security Vulnerabilities**

The current system procedures have a lot of security vulnerabilities in terms of mechanism and application. Security concerns were mostly discovered in the manner of identifying visitors and dependants living in military cantonment. Visitors were being identified using id NRC’s or travelling documents. This approach has security concerns such as impersonation and masquerading as possessing someone else’s identity. The study also discovered no proper record and identification parameters of service personal dependants and biological children living in the military cantonment

or barracks. This in itself presented security concern to the Zambia Army. Other security concerns were traced from the banquet hall constructed at the heart of Army Headquarters offices which host civil functions such as wedding and other social meetings. This building presented a security issue at the Zambia Army Headquarters because of social attachment to it.

4.3 **Problem Domain Modelling**

The current problem domain was modelled using soft system methodology (SSM) called rich picture. Rich picture represent traditions, customs, believes, wishes, emotions, misunderstanding, prepositions, possible occurrences, fear and dreams of people in an organisation. The rich picture is a flexible soft system development tool that gives developers chance to present problem domain in pictorial form according to the way they have understood the problem domain (Horan, 2000). Figure B.3 Appendix shows the rich picture representing the problem domain, procedures, emotion, behavioural, traits, wishes, expectation of people in military organisation towards terrorism and other related crimes.

4.4 **Requirement of the New System**

The requirements for developing a new system were divided into fingerprint biometrics and military police database requirements. The two requirement type was integrated to form an automated integrated fingerprint biometric system for military organisation (AFBSMO).The integration of biometric application into military police database provides the military organisation with the required security resilience. On the other hand, AFBSMO would improve military police service delivery. But before then, the product has to be described and modelled in the suitable way befitting software development processes. This includes providing documentation of the developed system as it transverse from one development stage to another.

4.5 **Product Perspective**

The automated fingerprint biometrics system for the military organisation is a new system that was intended to replace the current manual human identification and security vetting systems in the Zambia Army. The context diagram in Figure 6 illustrates the external entities and system interfaces for release. The system was

expected to evolve over several releases, ultimately deployed at various barracks and formations.

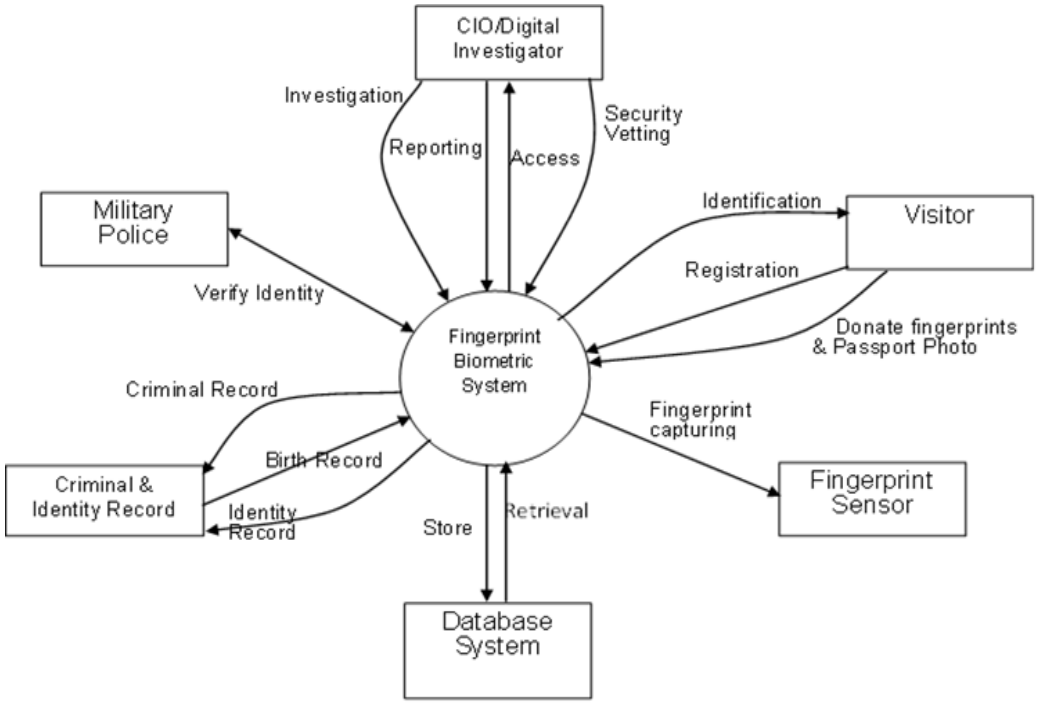


Figure 6:AFBSMO context diagram

4.5.1 **Philosophy and Process**

The philosophy behind the developed new system was to amend the security weaknesses identified in the current manual system. The current system is associated with various problems indicated in fact finding results of this chapter. These problems should be alleviated in order to improve institutional security provision and service delivery. Institutional security provision and service delivery can only be improved through business process refinement and automation. In this research input variables were sourced from Army form (FZA MP 12A see Appendix). The FZA MP 12A form contains information such as applicant details, fingerprint template, recording officer information, criminal data and human descriptive information. Examining this information could give birth to three security layers with special application to the military organisation. The first layer is the fingerprint biometric systems that could be deployed at gates for human identification and access control. The second layer is the production of automated service identification card. The last layer involved collection

of security information from the Guests. For the new system processes, actors and procedure data flow refer to Figure 7.

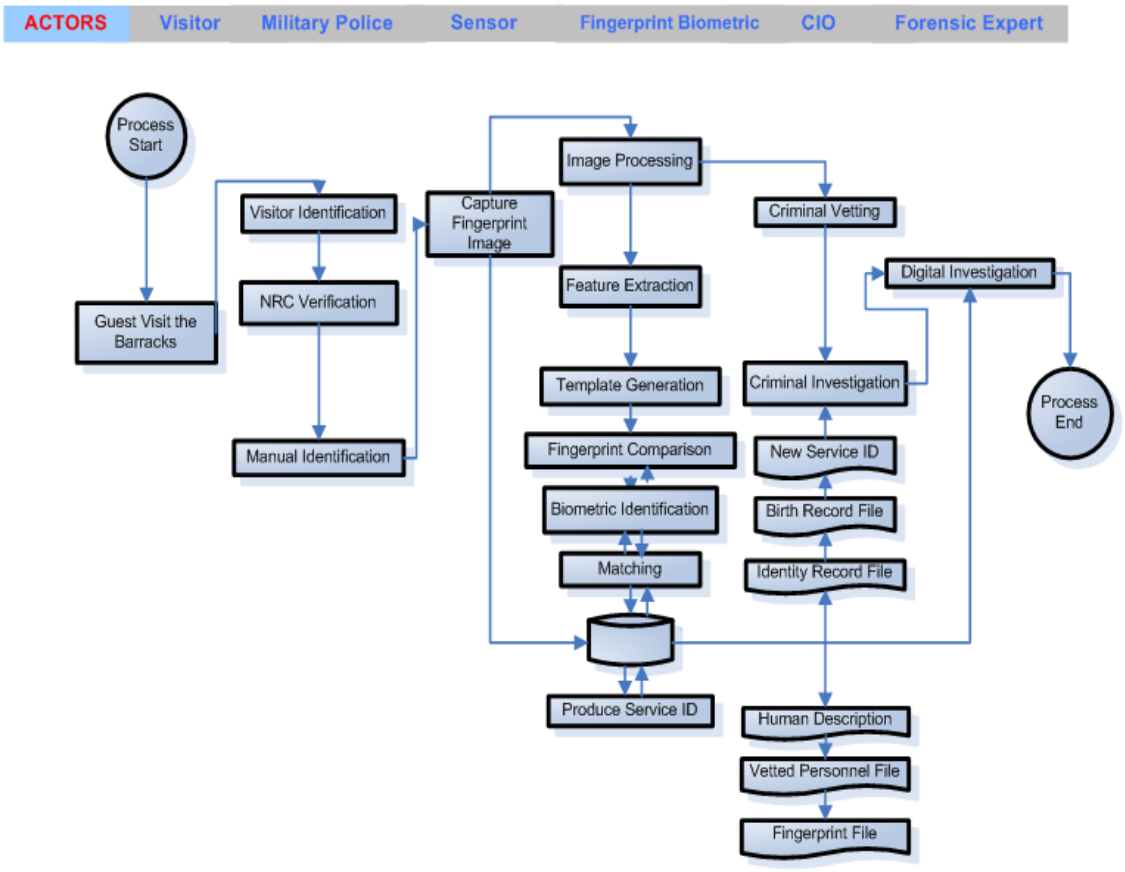


Figure 7: New System Processes, actors and procedure flow

4.5.2 **Military Police Database**

The database for military police is used for storage of military police data both in peace time and war times. It consists of tables, queries and store procedures dedicated for police duties. Police database was designed to depict manual interventions according to the roles and duties of Regimental Military Police (RMP).

4.5.2.1 **Roles of Military Police**

The Military Police are required to provide tactical military police support to the Zambia Army in military operations. When deployed, some of the roles the regimental military police performs include:

- General criminal investigation in peace time and war crime investigation in war time.
- Handling and collecting criminal evidence.
- Reconnaissance patrols.
- Prisoner handling.
- Search operations and road blocks.
- General policing duties within operational bases.
- Foreign police and military training
- Provide close protection operatives for senior military personnel on operations

4.5.2.2 **Specific Military Police Functions in Peace Time**

The military police (MP) unit is responsible for maintaining law and order within the military. In peace time, military police are deployed at entry point (gates) to control traffic and provide security of the military offices, personnel, installations, military stock both inside and outside the barracks. Civilian Guests are compelled to disclose their true identity which is verified by details on their identity card or travelling passport. Foreigners are required to obtain security clearance from Ministry of Defence (MOD) each time they intend to visit a barrack. The technique of verifying the identity of an individual using NRC or passport is undesirable for the military organisation because of the reported security flaws in the mechanisms and technique.

Military police also perform criminal vetting in conjunction with civil police on recruits and officer cadet joining the service. In vetting process the total of ten (10) fingerprints are captured from each applicant and manually recorded on FZA 12A forms. The tools employed in this process include Army form FZA 12A, ink, tipexand stamp. However, this technology is old, dirty, costly, time consuming and tedious in operation.

Another function of MP in peace time is to conduct digital forensic investigations. Digital forensics is sometimes referred to as digital forensic science and is defined as a branch of forensic science encompassing the recovery and investigation of material found in digital devices, often in relation to computer crime. Finally, military police also keeps the database of disciplinary record of service personnel, offences

committed and identity information of both officers in active service and reserve forces.

4.5.3 **Fingerprint Biometrics System in Military**

The process of human identification in the military goes beyond the normal fingerprint biometric systems. The development and usage of biometrics systems in any organisation is dependant on the policy. The biometric system developed for military is totally different from the civil or domestic one. This is due to the fact that the nature of the military organisation demands extreme security resilience and accurate mechanisms. Generally, biometrics systems suffers from the problems of false acceptance (FA) and false rejection (FR) which may be caused by orientation of finger on the sensor, the physical condition of the sensor and the physical state of the finger(e.g. finger may be greased, burnt, masked or scratched on the fore skin). Therefore, as a result of these challenges, redundancy fingerprints are created. Instead of capturing only a single print, military biometric system capture ten fingerprints which could be easily referred to in order to clarify any issue. The same ten fingerprints are reused for criminal vetting. On top of the 10 prints left and right thumb are captured simultaneously and stored as a single image. Additionally, left and right palm print are also captured and stored for extended identification. Ultimately, a fingerprint biometrics system developed for military organisation is usually large in scope and demand high levels of accuracy.

4.6 **Architectural Perspective**

The AFBSMO system architecture consist of three major components namely fingerprint sensor (scanner), business logic and relational database. These constituents consist of other smaller components which are related and perform specialised functions. The components of AFBSMO are integrated and linked to each other by functionality. The Architectural design of AFBSMO is shown in Figure 8.

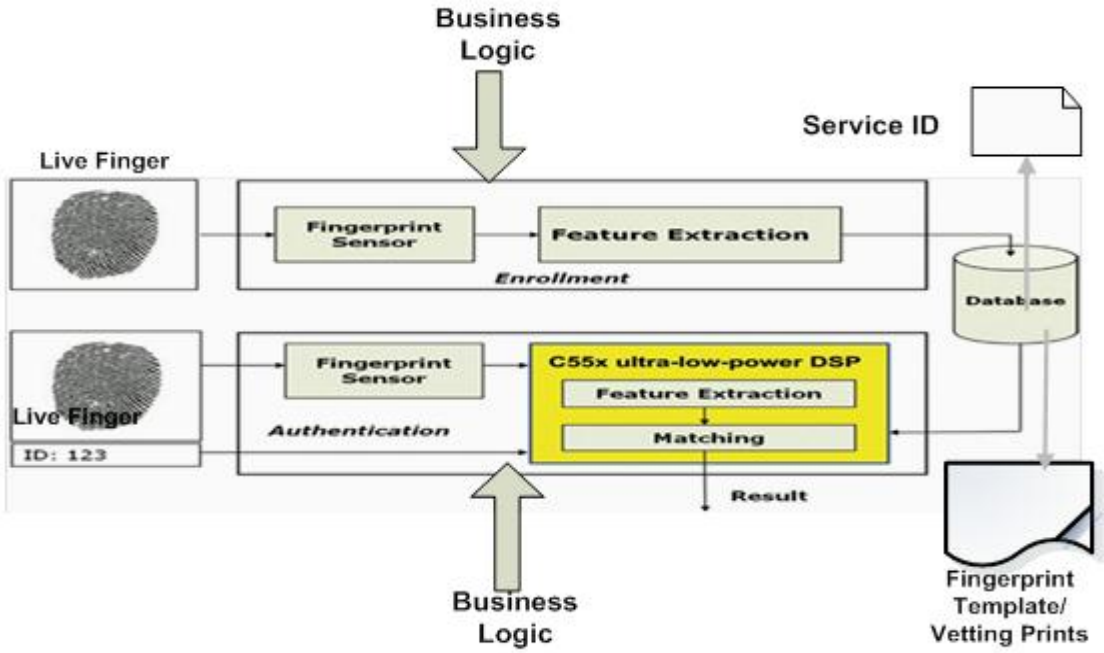


Figure 8:Fingerprint Biometrics Design

4.6.1 **Fingerprint Scanner**

Fingerprint scanner is the device that is used to capture fingerprint images from a live human finger. The fingerprint scanner is used as a principal security device in fingerprint biometrics system for live human identification and authentication.

4.6.2 **Business Logic**

Business logic consists of many algorithms that encode real-world problem in to determining how data can be created, extracted, displayed, stored, and changed. An algorithm describes the processes or set of rules which must be adhered to in performing certain action. Algorithms are combined together in business logic and function as a system. Business logic includes the following procedures:

4.6.2.1 **Fingerprint Image Processing**

The developed fingerprint biometrics system is based on Template Synthesis which is sometime referred to as super template. The super template is generated by the following stages:

(i) **Capturing**; Fingerprint capture process allows the operator to capture a person's fingerprint, printing it on visual studio runtime form, and storing

the image on the database table. The fingerprint scanner is used in this process.

(ii) **Normalisation**; Normalization operation acts on the input fingerprint image to standardize image pixel intensity values. In this study the research used image brightness normaliser code. Normalisation is the pre-processing stage in fingerprint template generation and is defined by variance analysis as illustrated in equation 3:

$$N(\mathrm{i,j}) = \begin{cases} \mathrm{M_0} + \sqrt{\frac{\mathrm{v_0}\ (\mathrm{I(i,j)-M})^2}{\mathrm{v_0}}} & \text{if } \mathrm{I(i,j)} > M \\ \mathrm{M_0} - \sqrt{\frac{\mathrm{v(I(i,j)-M)^2}}{\mathrm{v_0}}} & \text{otherwise} \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

Where:

M and V are the estimated mean and variance of image I (i; j), respectively, and M_0 and V_0 are the desired mean and variance values, respectively.

(iii) **Segmentation**. Segmentation is performed to separate foreground from background region. The research implemented theotsu Variance thresholding method on normalized fingerprint images. When minutiae extraction algorithms are applied directly to fingerprint image contains the background, the operation tend to generate noise result in false minutiae. The grey-level variance for a block of size W×W was calculated as shown in equation 4:

$$\mathrm{v(k)} = \frac{1}{\mathrm{w^2}} \sum_{\mathrm{i=0}}^{\mathrm{w-1}} \sum_{\mathrm{j=0}}^{\mathrm{w-1}} (\mathrm{I(i,j)} - \mathrm{M\ (k)})^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4)$$

Where:

V (k) is the variance for block k, image I (i, j) is the grey-level value at pixel (i, j), and M (k) is the mean grey-level value for the block k.

(iv) **Enhancement**. The configuration of parallel ridges and valleys with well-defined frequency and orientation in a fingerprint image provide useful information which helps in removing undesired noise. Gabor filter was applied because it has both frequency-selective and orientation-selective properties and have optimal joint resolution in both spatial and frequency domains. Therefore, it is appropriate to use Gabor filters as band pass filters to remove the noise and preserve true ridge and valley structures.

(v) **Binalisation**. Binalisation is the final stage in fingerprint pre-processing stages. It converts a grey level image into a binary image by improving the contrast between the ridges and valleys in a fingerprint image, and consequently facilitates the extraction of Minutiae. BW(x, y) represent the binary image which could be obtained by the equation 5:

$$BW(x,y) = \begin{cases} 1 & \text{if } I(x,y) \geq Tp \\ 0, & \end{cases}$$

Otherwise…… (5).

(vi) **Thinning**. Thinning process is a morphological operation that successively erodes away the foreground pixels in binary image until they are one pixel wide. The Thinning operation falls under feature enhancement stage of fingerprint image processing. Its main purpose is to create skeletonised binary image. Image filter is applied to digitalised image.

(vii) **Minutiae Detection**. The skeleton image is then used in the subsequent extraction of minutiae using cross numbering (CN) method proposed by Rutovitz. CN is computed by equation 6:

$$CN = \frac{1}{2}\sum_{i=1}^{8} I\, p_i - p_{i-1} I, p_9 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6)$$

After the CN is computed , the pixelsare is classified according to the property of its CN value (Chaudhari et al., 2014).

4.6.2.2 **Enrolment Stage**

During the enrolment stage, biometric data are obtained, linked with identity, and encoded for storage, retrieval and matching. Fingerprint scanners are used to collect data and verify identities (Clodfelter, 2010).

4.6.2.3 **Identification**

Identification is conducted to verify the live scanned fingerprint from an individual or subject. The matching algorithm compares a current fingerprint image against the previous enrolled print, checking whether they come from the same finger.

4.6.2.4 **Database Operations**

Business logic also contained a database operation which is sometimes referred to as relational operation. Database procedure or operations is a collection of database

tasks defined by end users or application code, for example, a batch job or Extraction algorithm, Transformation, and Loading (ETL) processing. The basic database operation includes insert, update, delete and search operations which were implemented in the study. Other database relation operation includes Union, Selection and append.

4.6.3 **Database**

The database component of AFBSMO is responsible for data storage. The database consists of tables which are physical structures responsible for data storage. A table is broken further into attributes. From the above explanation, the table can be defined as the collection of related attributes while relational database may be defined as the collection of relations.

4.6.4 **Behavioural Modelling**

The use-case diagram was used to envisage the behaviours of various objects in AFBSMO. Use-case diagram depicts graphically system actors, goals and their dependencies (Pooley & Stevens, 1999). The use-cases include; verify Identity, Enrol Fingerprint, Identification, Criminal Vetting, Criminal Investigation, Fingerprint Donor, Forensic Investigation and Fingerprint Analysis as shown in Figure 9:

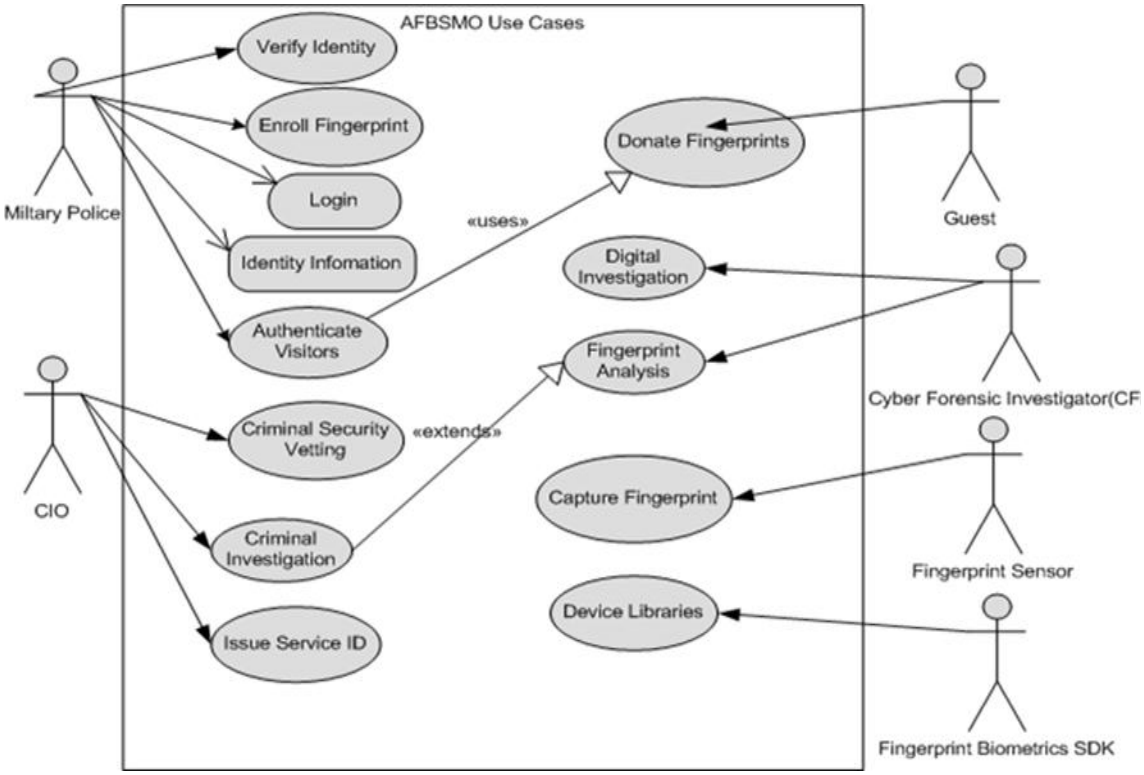


Figure 9: Use Case Diagram

4.6.5 **Use Case Description**

Use cases were described by stating the associated actor or actors, preconditions, description, exceptions and post-conditions as follows:

4.6.5.1 **Verify Identity**

Actors: The main actors are Military Police and Guest.

Preconditions: Guests must visit the barrack and submit personal identity (NRC or passport)

Description: Military police is in charge of security in the barracks. When the guest visits the barracks they are asked to identify themselves. After identification process, Guests are told to produce their NRC or passport for the officer to verify the identity**.**

Exceptions: Service Personnel, Member of public not in possession with NRC and foreigners without security clearance.

Post-conditions: Access Granted/institute criminal investigation/report the matter to civil police.

4.6.5.2 **Enrol Identity**

Actors: The main actors are Military Police and Guest.

Preconditions: Guest must visit the barrack and submit personal identity (NRC or passport) and should be Guests visiting the barracks for the first time.

Description: Members of public without NRC and Foreigners without security clearance.

Exceptions: Members of public without NRC and Foreigners without security clearance.

Post-conditions: Authentication and Verification.

4.6.5.3 **Identification**

Actors: The main actors are Military Police and biometric system.

Preconditions: Guest must visit the barrack and submit personal identity (NRC or passport), fingerprint donated and should have been enrolled in the biometric system. Fingerprint sensor must be connected to the computer.

Description: Military police enrol fingerprint of Guests on the biometric system. The Guest fingerprint is registered in the database. In the next visit the system will recognise the enrolled Guest.

Exceptions: Service Personnel, Member of Public not in Possession with NRC and Foreigners without Security Clearance.

Post-conditions: Access granted/Access denied.

4.6.5.4 **Fingerprint Donation (Acquisition)**

Actors: The main actors are Military Police, fingerprint sensor, Guest (Job Applicant), CIO and biometric system.

Preconditions: Guest must visit the barrack and submit personal identity (NRC or passport) and should have been enrolled in the biometric system. Fingerprint sensor must be connected to the computer.

Description: Fingerprint print may be acquired for biometric usage or criminal vetting usage.

Exceptions: Member of public without NRC and Foreigner Guests without security clearance from the Ministry of Defence.

Post-conditions: Fingerprint image for criminal vetting and human biometrics identification key (fingerprint template).

4.6.5.5 **Digital Investigation**

Actors: The main actors are forensic investigator and biometrics system.

Preconditions: Cybercrime committed.

Description: Forensic investigator detects the occurrence cyber-attack against the security information.

Exceptions: N/A.

Post-conditions: Arrests, fix security vulnerabilities, institute criminal proceedings.

4.6.5.6 **Criminal Investigator**

Actors: The main actor is CIO.

Preconditions: Crime committed and reported.

Description: CIO investigates the reported crime.

Exceptions: Where no crime was reported.

Post-conditions: Institute Criminal Proceedings.

4.6.5.7 **Security Vetting**

Actors: The main actors are CIO, job applicant and biometric system.

Preconditions: Guest must visit the barrack and submit personal identity (NRC or passport) and should have been enrolled in the biometric system. Fingerprint sensor must be connected to the computer.

Description: Ten (10) fingerprints are captured and compared with those fingerprints of people who committed criminal offence before.

Exceptions: Member of public without NRC and foreigners.

Post-conditions: Criminal status of job applicant is established. Applicant Accepted/ Rejected.

4.6.5.8 **Capture Fingerprint**

Actors: The main actor is a fingerprint sensor.

Preconditions: Sensor must be connected to the computer and image display environment set.

Description: Capture fingerprint both for Biometrics and for criminal vetting

Exceptions: Member of public of public below NRC.

Post-conditions: Image processing, refinement and storage

4.6.6 **Requirements Modelling**

Requirement modelling prototypical user requirement based on external design. The researcher used software engineering principles to model the developed products. Requirements prototypes were used to discover and clarify the functional and data requirements for software and business systems. Additionally, the requirements models are usually used as specifications for the designers and programmers of the system applications.

4.6.6.1 **Requirement Specification**

Possible system requirements were extracted from military police procedures, operations, duties, experiences of researchers working in military organisation

(knowledge transfer) and the reviewed literature. Other sources of requirements are given in introduction section of this chapter. Every piece of requirements was tested against qualification strategy model as shown in Figure 10. The Specific Derived requirements and their qualifying strategies are given in Table 2.

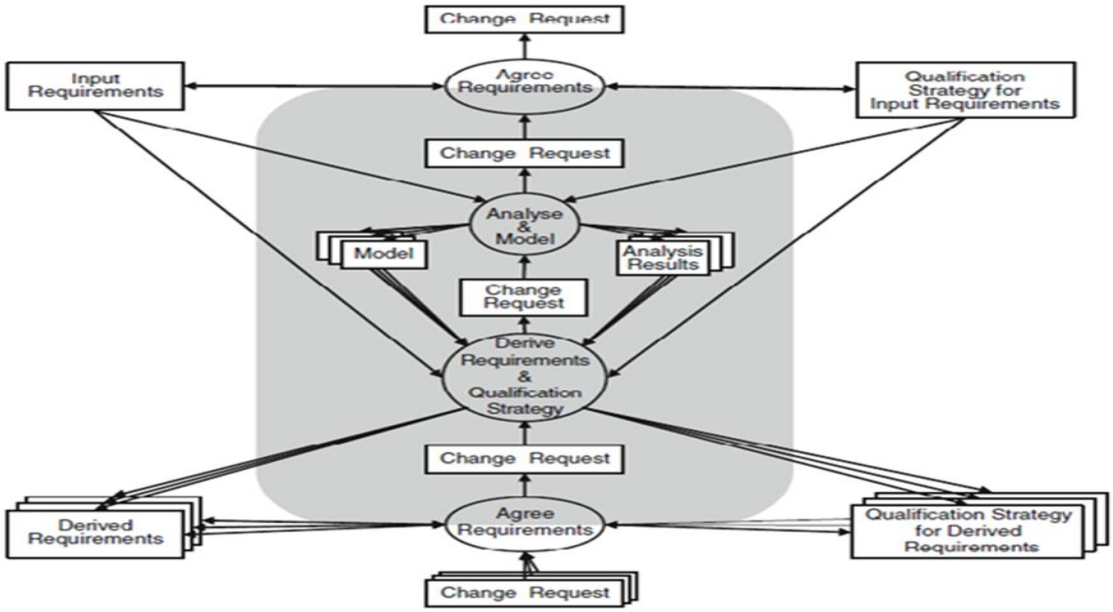


Figure 10:Model for requirement analysis

Table 2: Derived requirements and their Qualifying Strategies

Serial No	Specific Requirement Name	Qualifying Strategies
1	Capturing particulars of a Guest	Strategy to identify a Guest
2	Guest time in and time out	Strategy to establish Guest visit time
3	Latent print	Strategy to use fingerprint optical scanner.
4	Ten (10) fingerprint	Strategy to identify a criminal
5	Enrol fingerprint	Strategy to extract minutiae
6	Verify identity	Strategy to verify a Guest
7	Develop Minutiae based fingerprint	Strategy to improve accuracy
8	Capture passport size photo	Strategy to automate and control Service ID issue
9	Fingerprint classification	Strategy for searching
10	User levels of access	Strategy to escalate Privileges
11	Capture fingerprint image	Strategy to acquire fingerprint for vetting
12	Crime name	Strategy to establish the committed name of crime
13	Type of crime	Strategy to define punishment
14	Punishment	Strategy to define jail term

15	Create database	Strategy for storage
16	Fingerprint scanner	Strategy to capture fingerprint
18	Security of the system	Strategy to secure the system

4.6.6.2 **Functional and Non Functional Requirement**

System requirements can be categorised into two types: functional and non-functional requirements. Functional Requirement outlines functions of the system and its components. Typically, functional requirements specify behaviour of the system. The behaviour of the system includes the set of inputs, business logic and outputs. **N**on-functional requirements describe how the system works and that include specifying system quality characteristics or quality attributes. Non-functional requirements are sometime referred to as technical requirement. Non-functional requirement include system reliability, security, maintainability, usability, scalability and accuracy. Requirements were graded according to priorities or importance. The study graded requirements as: (i) 1 = must implement, (ii) 2=should implement (time allowing) and (iii) 3=be nice to have. The requirements and respective grades are shown in table 3 and 4 below.

Table 3: Functional Requirement Table

Functional Requirements Table			
No	**Description**	**Referred Use Cases**	**Priority**
1.1	Capture fingerprint	Capture fingerprint	1
1.2	Fingerprint Registration	Fingerprint sensor/ fingerprint SDK	1
1.3	Fingerprint verification	Fingerprint sensor/ fingerprint SDK	1
1.4	Verify user	Fingerprint sensor	2
1.5	Capture human identity information	Identity Information	2
1.6	Capture human description	Identity Information	3
1.7	Capture human criminal record	Investigate crime & digital forensic	3
1.8	Capture Guest Time In and Time Out	Registration & verify identity	2
1.9	Capture ten (10) fingerprint	Fingerprint capture & security vetting	1

1.10	Display number on minutiae in the fingerprint	Fingerprint capture.	3
1.11	Capture passport size photo of Guest and service men	Identification information.	1
1.12	fingerprint classification	Fingerprint capture & identity data	3
1.13	Capture user access level	Identification information	3
1.14	Crime details	Investigate crime & digital forensic	3
1.15	Crime type	Investigate crime & digital forensic	3
1.16	Image Refinement	Investigate crime & digital forensic	1
1.17	Create Login form	Login form	1

Table 4:Non-Functional Requirement Table

No	Requirement	Description	Priority
1.1	Reliability	Archive up time of 99.9%	1
1.2	Security	Control against unauthorized access, changes, deletion & hacking	1
1.3	Maintainability	Easy to make changes to existing and add new features	3
1.4	Usability	Achieve friendless	2
1.5	Scalability	Support many users	2
1.6	Performance	Fast information retrieval and processing power	1
1.7	Accuracy	Reduce false Reject/Acceptance rate up to 99.9%	1

4.6.7 **Activity Diagram**

An activity diagram visually presents a series of actions and flow of control in a system similar to a flowchart and a data flow diagram. Activity diagrams were used to

display the activity and actions in the use case diagrams. Furthermore, activity diagrams are often used in business process modelling to describe the business logic. In both cases an activity diagram will have a beginning and an end. This paragraph describes some activities performed in AFBSMO.

4.6.7.1 **Main Activity Diagram**

The automated fingerprint biometric system for military organisation consists of many activities. Firstly, guests visiting military barracks must identify themselves before military police officers guarding the main gate or entry points who in turn verify visitor identity through checking details appearing on guest NRC or passport. This process is conducted manually. The critical attributes verified are names and photo appearing on NRC or travelling passport. Figure 11 shows the main activity diagram of an AFBSMO.

Figure 11: AFBSMO Main Activity Diagram

4.6.7.2 **Verification Activity Diagram**

Verification use-case activity analysed the process which takes place when the sightseer visit the barracks or restricted area. The guest identifies their individuality to

military police that is deployed at the gates. The activities and action are already describe but amplified in Figure 12.

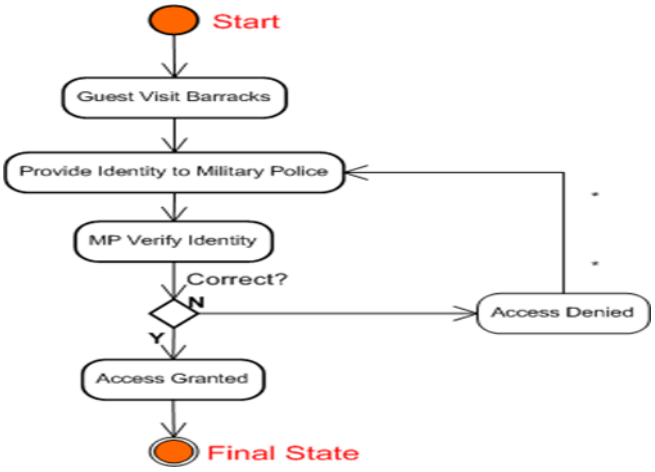


Figure 12:Activity diagram for verification process

4.6.7.3 **Fingerprint Donor Activity Diagram**

Fingerprint capturing activity diagram shows action takes place when a person provides fingerprints to the military. Activities and action involved in fingerprint donor activity diagram include pressing the finger on the scanner and image acquisition. The acquired image is divided into two (2) streams according to the usage. The image captured may be used either for biometrics or criminal investigation/Service Identity card production. Biometric images are processed further before storage while the criminal investigation image is saved directly in the relational database table. Criminal investigation images can be used further in digital forensics. Fingerprint donor activity diagram is shown in Figure 13.

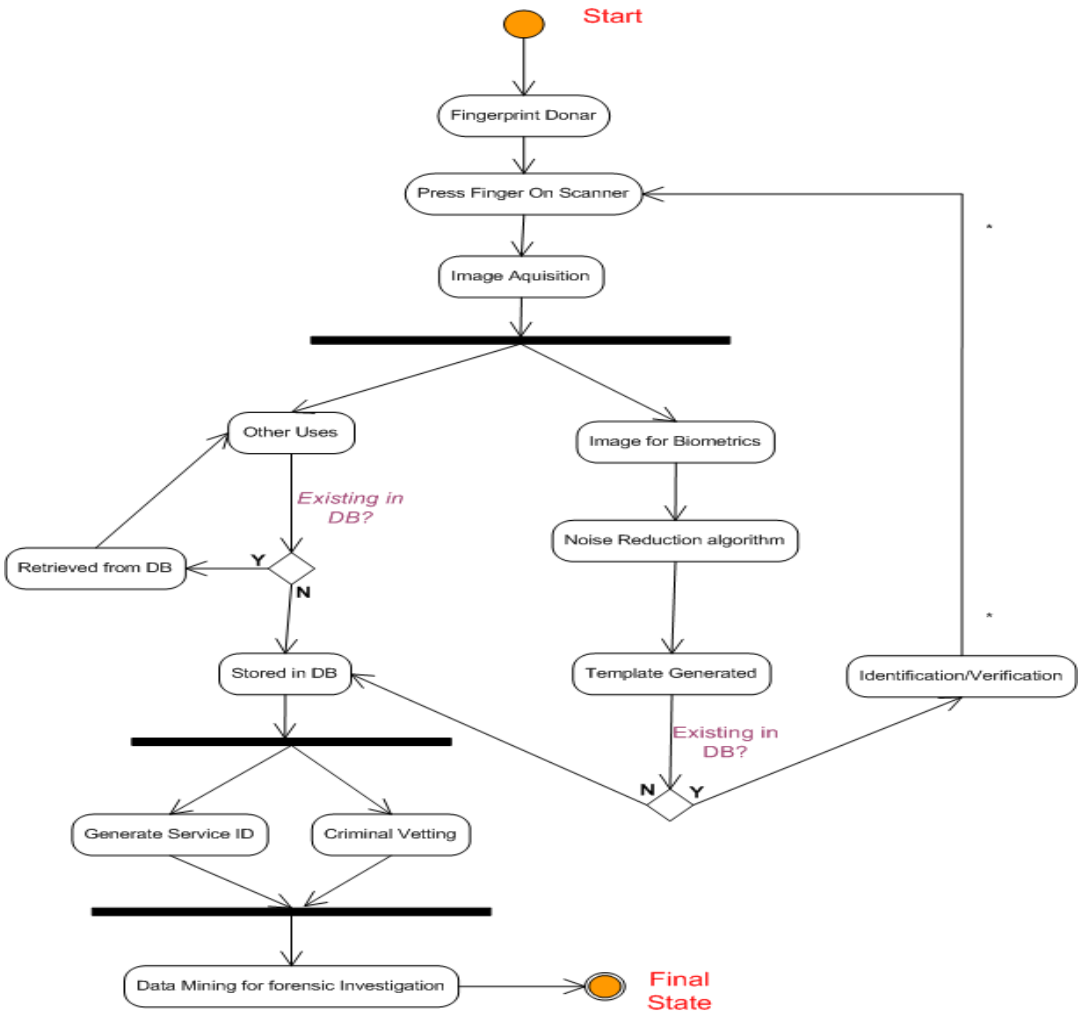


Figure 13:Fingerprint Donor activity diagram

4.6.7.4 **Fingerprint Analysis Activity Diagram**

The fingerprint analysis activity diagram includes activities and actions performed by CIO and that of biometrics system. Image capture for biometric usage normally passes through pre and post image processing stage. The two stages are important because fingerprint images captured from fingerprint sensor are normally noise and include false minutiae as shown in activity diagram Figure 14:

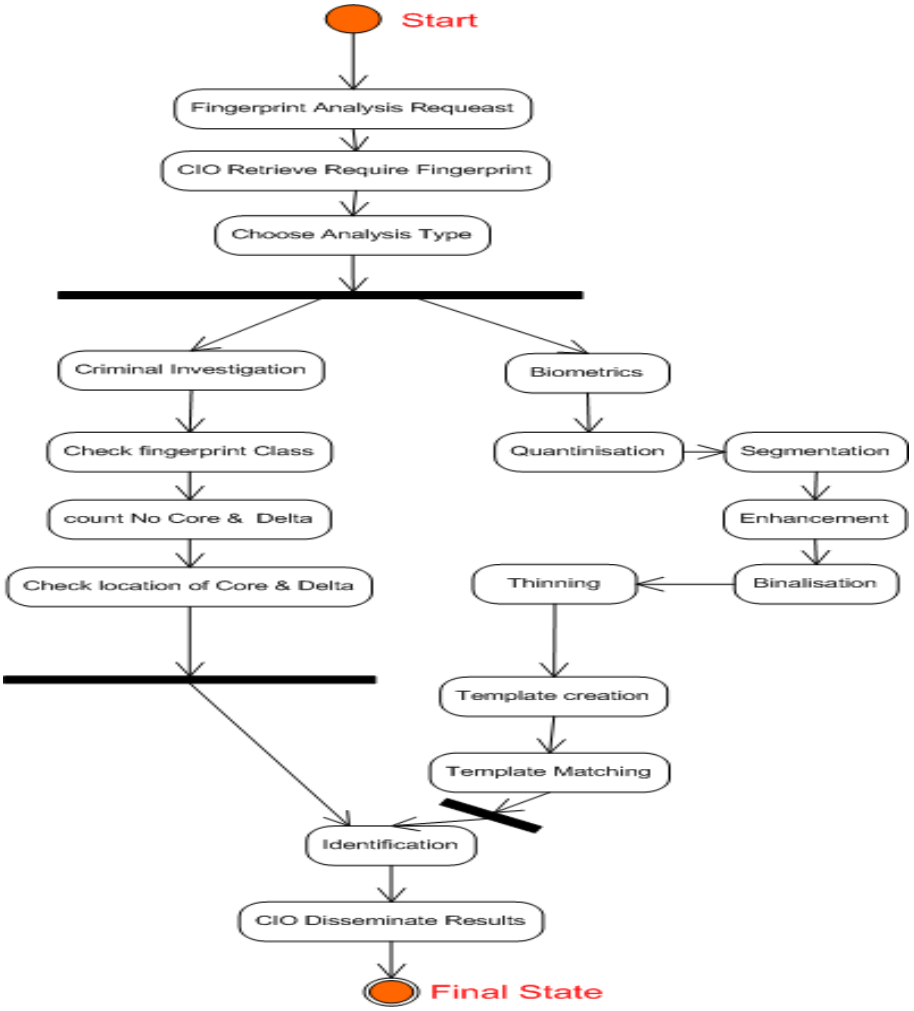


Figure 14:Fingerprint Analysis activity diagram

4.7 Object Oriented Analysis

Object Oriented Analysis model real-world entities using class diagram. The class diagram is described as a static diagram in this study. It represents the static view of the developed application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram is used to describe the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely employed in the software modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram (Langer, 2008: 248-252). Figure 15 shows class diagram of AFBSMO.

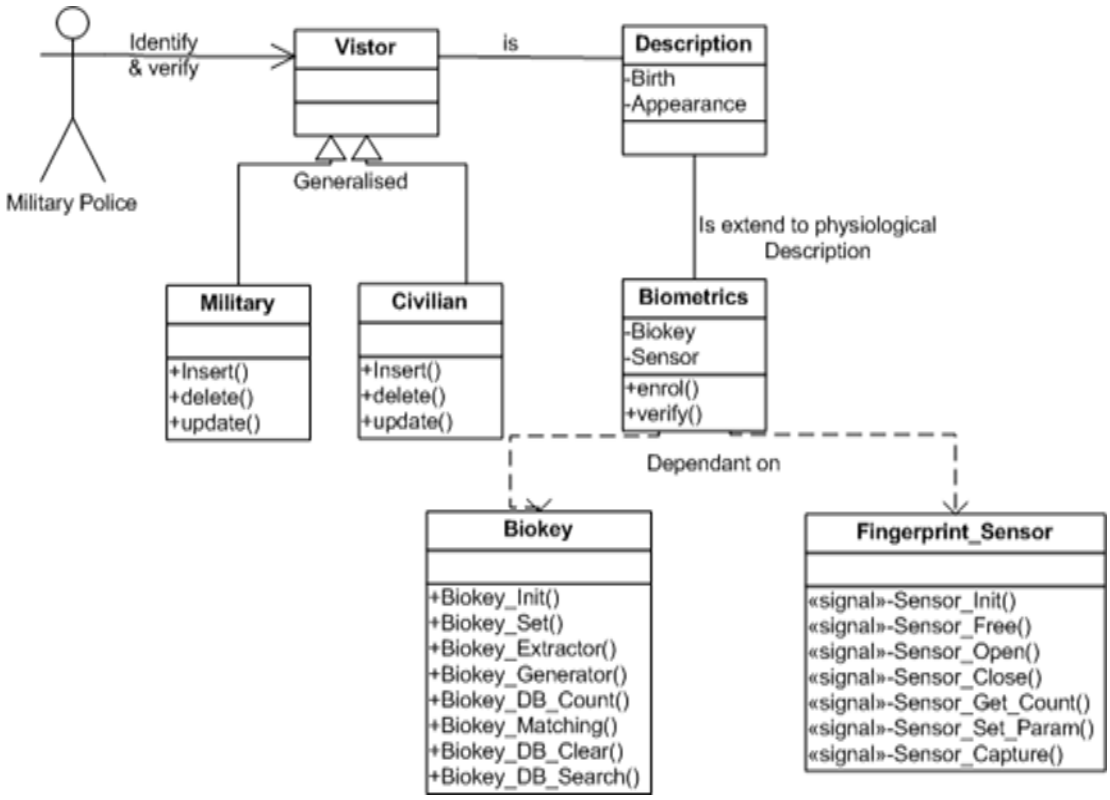


Figure 15:Snap Class Diagram

4.7.1 **Chapter Summary**

Analysis involved the study of procedures or business processes in order to identify goals and requirements desired to be implemented in the system. Analysis and design, as scientific methods, always go hand in hand; they complement each another. Every design process is built on results of a preceding analysis, and every analysis requires a subsequent designed to verify the correctness of results.

CHAPTER 5. **DESIGN**

5.1 **Introduction**

Systems design is the method of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. This process is devoted to explain how the requirements specified in analysis stage were translated into the designed aspect of AFBSMO. Systems design in many cases is seen as the application of systems theory to product development.

5.2 **Application Architectural Design**

Architectural design of the developed system is based on the three-tier model. The three-tier model consists of presentation, middle and the data service tiers. Tiers in computer science are sometimes referred to as layers. The three layers in this architectural design are functionally independent but inter-linked and collaborative in processing and resource allocation. The architectural design shown in Figure 16 is logical representation of the developed AFBSMO.

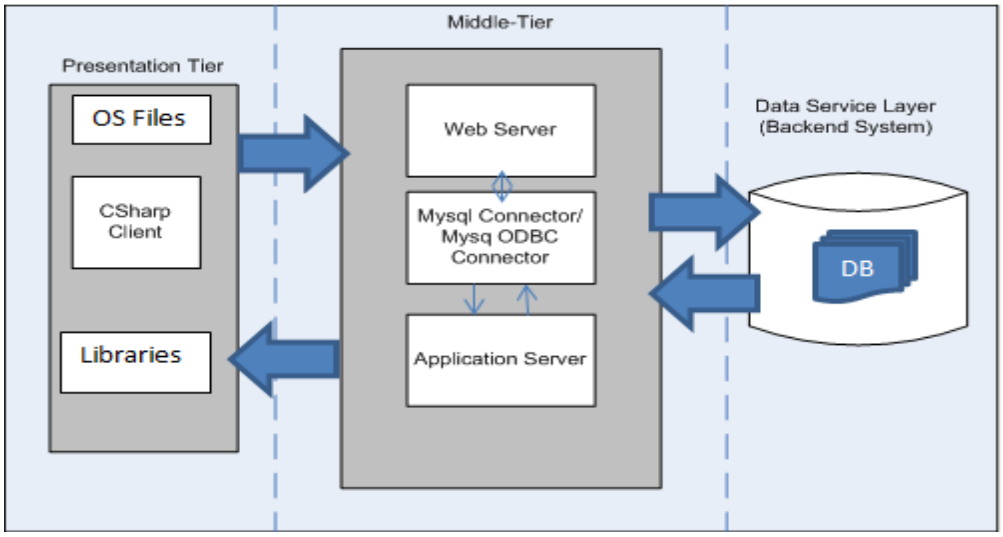


Figure 16:AFBSMO Applications Architectural Design

5.2.1 **Presentation-Tier**

Presentation-tier includes C# client (computer running C# compiler) which integrates Operating system (OS) files and other libraries (dll files) to communicate with other devices and applications. This study used visual studio 2010 environment with C#

object oriented programming language and digital Personal Fingerprint System Developer Kit (SDK). C# environment communicate with OS and database through other components illustrated in architectural design shown in Figure above.

5.2.2 **Middle-Tier**

The middle-tier includes web server, MySQL Connector and application server. Middle-tier is sometimes referred to as middleware. Middleware software is adaptor applications software that acts as a bridge between an operating system or database and applications, especially in client server architecture. The web server used in this study was WAMP Server which is the database server for MySQL. MySQL database does not communicate directly with visual studio application instead it communicates through the middleware programs such as MySQL Connector, SQL Connector and ODBC Connector as shown in Figure 5.2. In desktop application an application server works as a compiler. It converts computer program written in high level language into low level language for computer to interpret and understand.

5.2.3 **Data Tier (Data Service Layer)**

The data layer interacts with persistent data usually stored in a database or in permanent storage. This is the actual DBMS access layer. It can be accessed through the business services layer and on occasion by the user services layer. This layer consists of data access components (rather than raw DBMS connections) to aid in resource sharing.

5.2.4 **Benefits of Three Tiers Architectural Design**

This project Architecture design provided the following benefits.

- Scalability—Each tier can scale horizontally. For example, you can load-balance the Presentation tier among three servers to satisfy more Web requests without adding servers to the Application and Data tiers.
- Performance—Because the Presentation tier can cache requests, network utilization is minimized, and the load is reduced on the Application and Data tiers. If needed, you can load-balance any tier.
- Availability—If the Application tier server is down and caching is sufficient, the Presentation tier can process Web requests using the cache.

However, this design introduces some process communication complexity because modular communication is done through other programs. This approach makes the process of program debugging difficult because the programmer may not effectively trace which application error occurred.

5.3 **Hardware Design**

Hardware design specified how hardware components were integrated in the developed system.

5.3.1 **Architectural Design**

The AFBSMO hardware design architecture consists of three (3) major components namely personal computer, digital fingerprint scanner and digital camera. Personal computer acts as a server, hosts the drivers for peripheral devices and software development application. Fingerprint scanner is connected to computer through USB port. Fingerprint scanner is used to capture and verify human finger. Digital camera connects to the computer system using USB port. Both digital camera and fingerprint scanner interact with the computer system through a middleware program called drivers. Figure 17 shows hardware architectural design of AFBSMO.

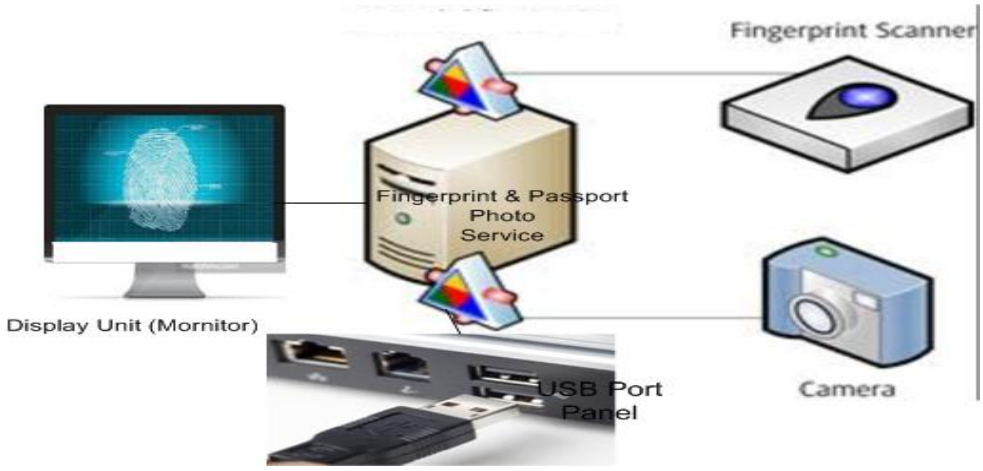


Figure 17:Hardware Architectural Design

5.4 **Software Design**

Software Design discuss all internal software components, including Cost of the shelf (COTS) and their configuration. Provide detailed design for all software components being built including software integration.

5.4.1 **System Modules**

This project is divided into two main modules. The module for biometrics system which is used for human identification and the module for military police administration duties. Military police administration modules include classes used for security vetting and general police duties as stated earlier. Figure 18 shows various classes found in the developed system.

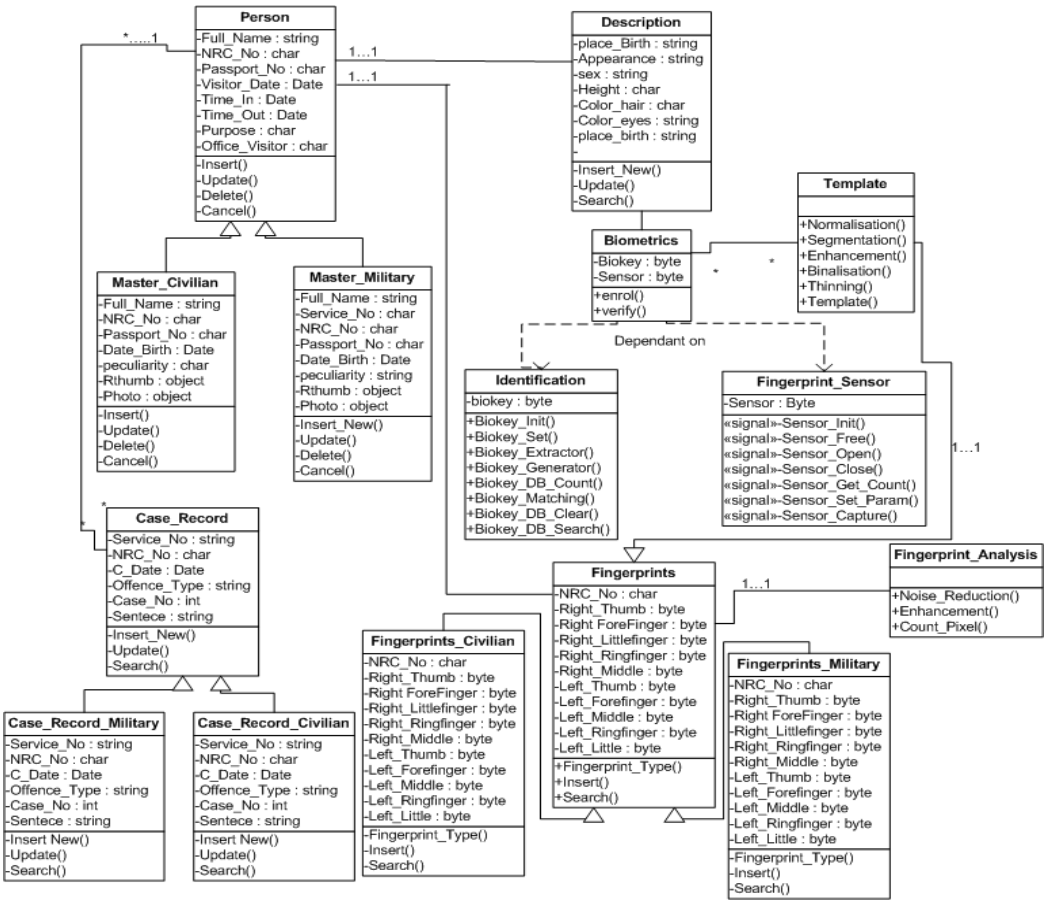


Figure 18: Detailed Class Diagram

5.4.2 **Software Integration**

Software integration section discussed various applications and their architectural design that enables AFBSMO to communicate or collaborate with other software

application. The applications referred to include MySQL database Server and middle applications. Each of this application software performs specialised function.

5.4.2.1 **MySQL Database**

MySQL is the world second most widely used relational database management system (Solid IT, 2015). It is open source, scalable and support a lot of data format. MySQL was used in development process as the relational database management system (RDMS). MySQL database architecture is layered. The topmost layer contains services mostly network-based client/server amenities which include connection handling, authentication, security, and so forth. The second layer habor the brain of MySQL relational database. It include threads controls, query parsing, analysis, optimization, caching, and all the built-in functions such as dates, times, month, and encryption. Any functionality provided across storage engines exists at this level. Layer 2 also performs handshark process with Layer 3 processes to handle data storage mechanisms. For architectural design refer to Figures 19.

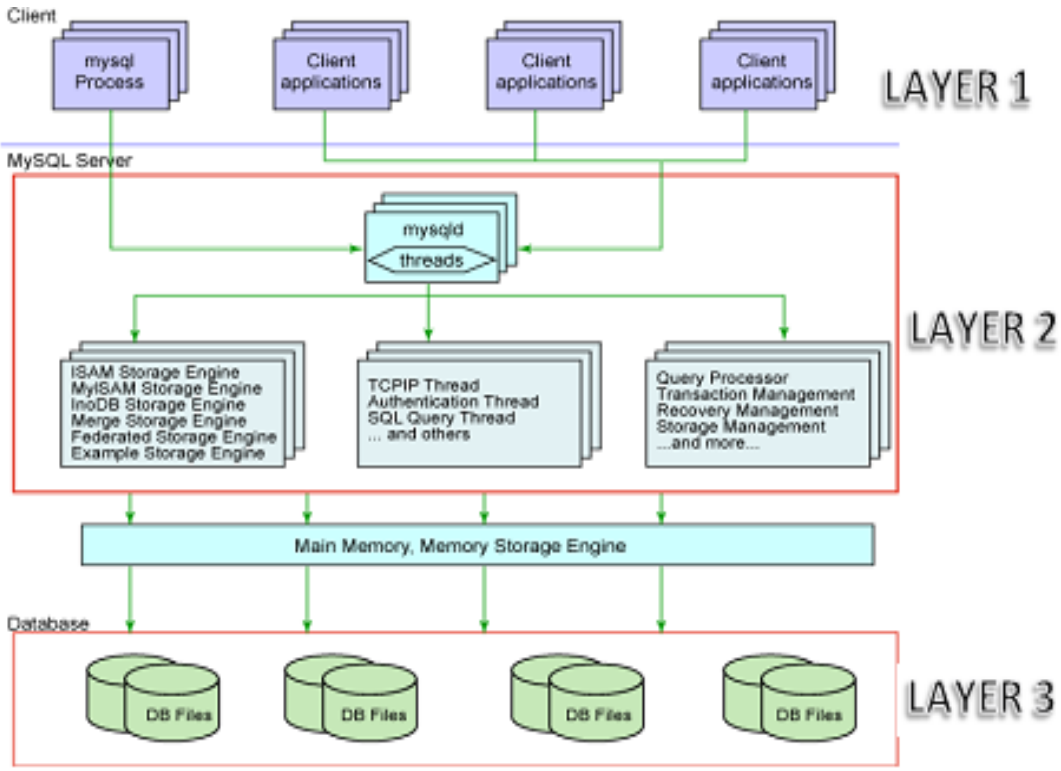


Figure 1: MySQL layered Architecture

Source: (William, 2011)

5.4.2.2 **Middleware**

Visual studio is Microsoft developer suit. By default Visual studio was created to support SQL database which is also Microsoft appropriatory software product. MySQL database is an open source database and does not communicate directly with visual studio development suit. The communication is facilitated by means of employing a third hand tool called MySQL Connector. MySQL Connector works exactly like Open Source Data Connectivity (ODBC) adaptor. The later could not be used during development because its driver failed to detect the existence of MySQL database on windows 7 64 bit.

5.4.2.3 **SDK Architecture**

The SDK Architecture consist of four basic components include application program, ActiveX Control OCX, device drivers and the fingerprint sensor as shown in Figure 20. The four components provide interaction and communication between internal and external interfaces. Internal interfaces refer to other components in computer system while external interfaces refer to devices connected to computer system.

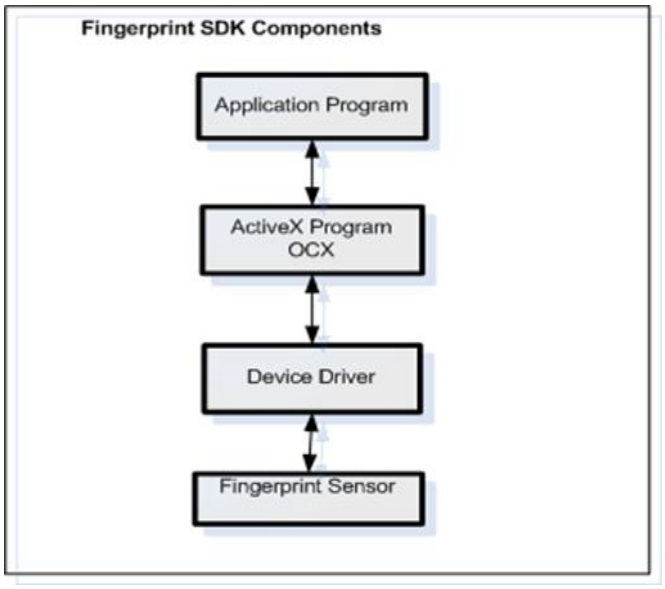


Figure 19: SDK Architecture

5.5 **Biometric System Data Flow Design**

An important issue in designing a practical *biometric system* is to determine how an individual is going to be recognized. Depending on the application context, a biometric system may be classified as verification or an *identification* system (Maltoni et.al, 2009). Biometrics information processes were designed using Data flow Diagram. The Data Flow Diagram (DFD) is a graphical representation of the flow of data through an information system. DFD represented information processes from the data point of view. DFDs were also used to visualize how the system would operate and accomplish specific task.

5.5.1 **Enrolment**

Operator enrolment is a procedure that is responsible for registering individuals in the biometric system storage. During this process, the biometric characteristic of a subject is first captured by a biometric scanner to produce a sample. A quality check is often performed to ensure that the acquired sample can be reliably processed by successive stages. A feature extraction module is then used to produce a feature set. The template creation module uses the feature set to produce an enrolment template. Some systems collect multiple samples of a user and then either select the best image (or feature set) or fuse multiple images (or feature sets) to create a composite template. The enrolment process then takes the enrolment template and stores it in the system storage together with the demographic information about the user (such as an identifier, name, gender, height, etc.). Figure 21 shows how data flows during fingerprint Enrolment process.

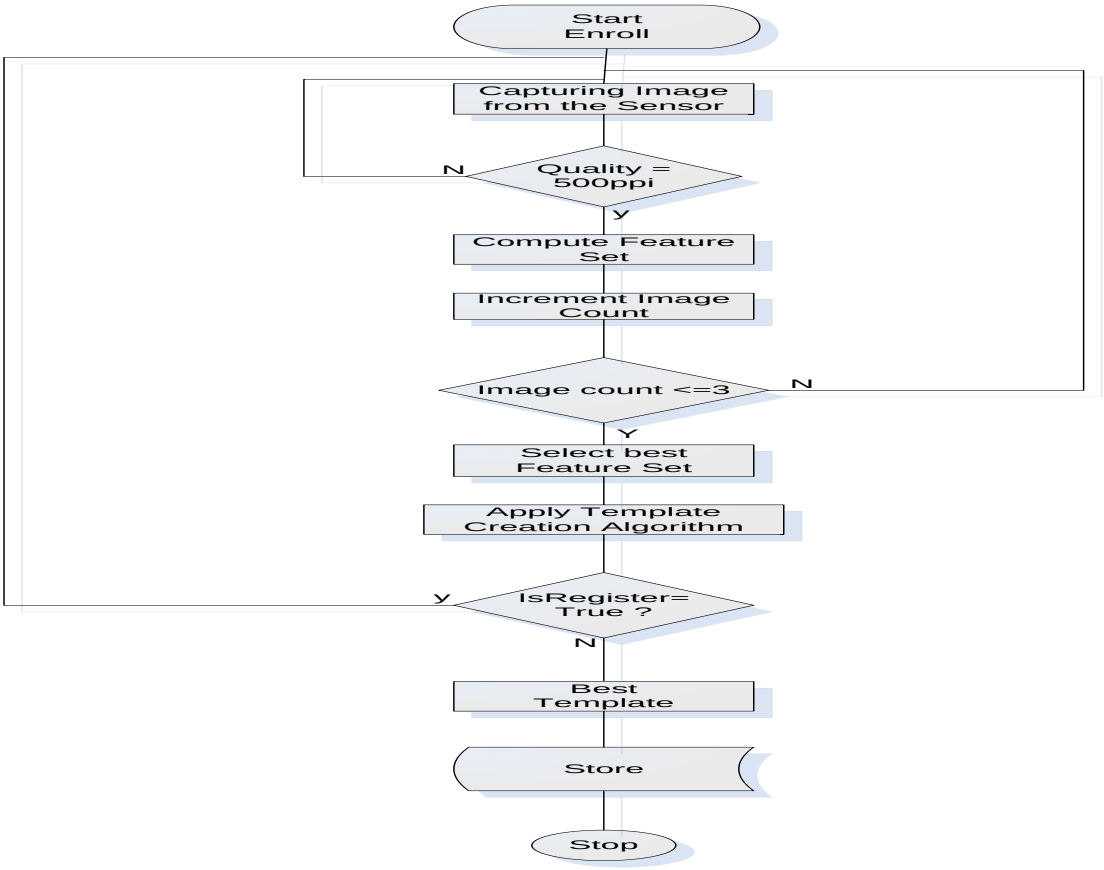


Figure 20:Enrollment DFD

5.5.2 **Verification**

Verification procedure is liable for confirming individual identity. During the recognition stage, an identifier of the subject (such as username or PIN [Personal Identification Number]) is provided to claim an identity. The Biometric Scanner Captures the characteristic of the subject and converts it to a sample, which is further processed by the feature extraction algorithm to produce a feature set (characteristics). The resulting feature set is fed to the matcher, where it is compared against the enrolment template(s) of that subject (retrieved from the system storage based on the subject's identifier). The verification process produces a match/non-match decision. Figure 22 shows verification process DFD.

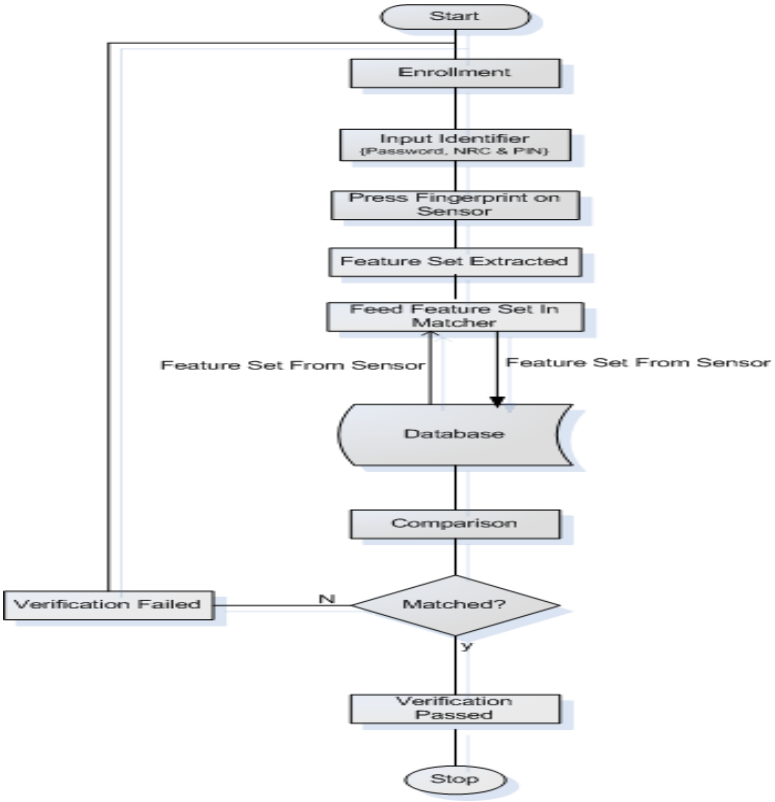


Figure 21: Verification

5.5.3 **Identification**

In identification process, the user does not explicitly claim an identity and the system compares the feature set (extracted from the captured biometric sample) against the templates of all (or a subset of) the subjects in the system storage. The output is a candidate list that may be empty (if no match is found) or contain one (or more) identifier(s) of matching enrolment templates. Since identification of large image in the databases is computationally expensive, a pre-selection stage is often used to filter the number of enrolment templates that have to be matched against the input feature set. Figure 23 shows the biometrics identification process. The comparison is made between a finger on the sensor with the minutiae points (template) stored in the database. If there is a match identification process is successful or otherwise the opposite is the case.

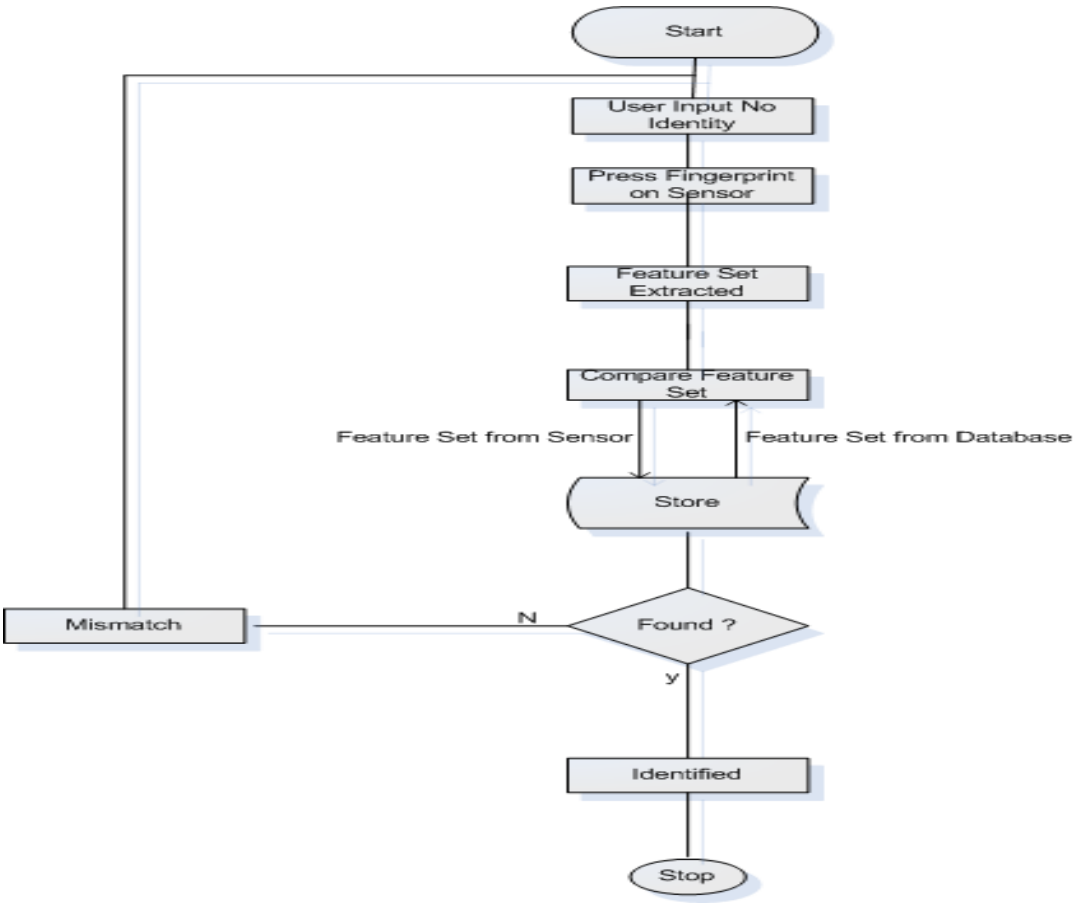


Figure 22:Identification Process DFD

5.5.4 **Template Generation**

Template generation is sometimes referred to as template synthesis or bio key generation. Firstly, the template generation is initialized by capturing new image. Minutiae based fingerprint biometrics is highly sensitive to noise. Therefore, the image quality is tested for acceptable biometrics quality levels as shown Figure 24. If the image captured is below quality levels the error is reported to advise the user of the need of new image otherwise template generation algorithm is applied. Template generation algorithm consider extracted prominent feature, their spatial coordinates and count the scores (points).

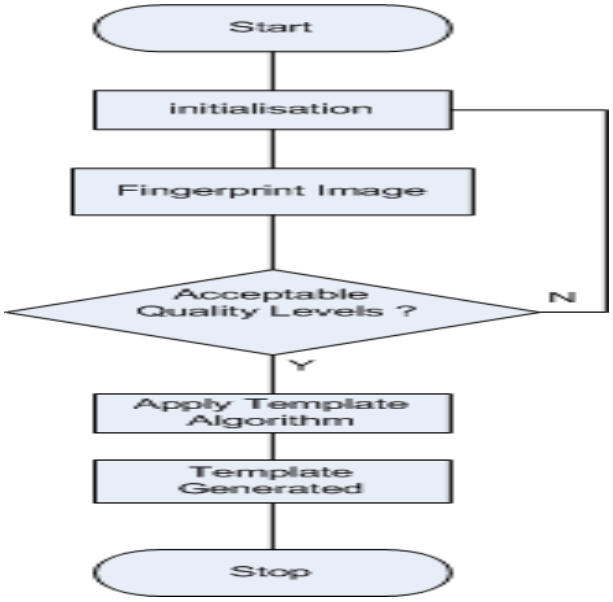


Figure 23:Template Generation DFD

5.5.5 **Security Vetting**

Zambian law requires individuals joining the service to undergo **security vetting** in order to gain access to government information. Government information in this case is referred to such information used civil Police, military and office of the president (OP). Vetting is intended to assure government bodies that the individuals employed in public service have not been involved in espionage, terrorism, sabotage or actions intended to overthrow or undermine democracy by political, industrial or violent means. It also assures the government department (service bodies) that the recruited individuals have not been a member of, or associated with, any organisation which has advocated such activities or has demonstrated a lack of reliability through dishonesty, lack of integrity or behaviour. Finally the process assures the department that the individual will not be subject to pressure or improper influence through past behaviour or personal circumstances. The data flow diagram of vetting process is shown in Figure 25.

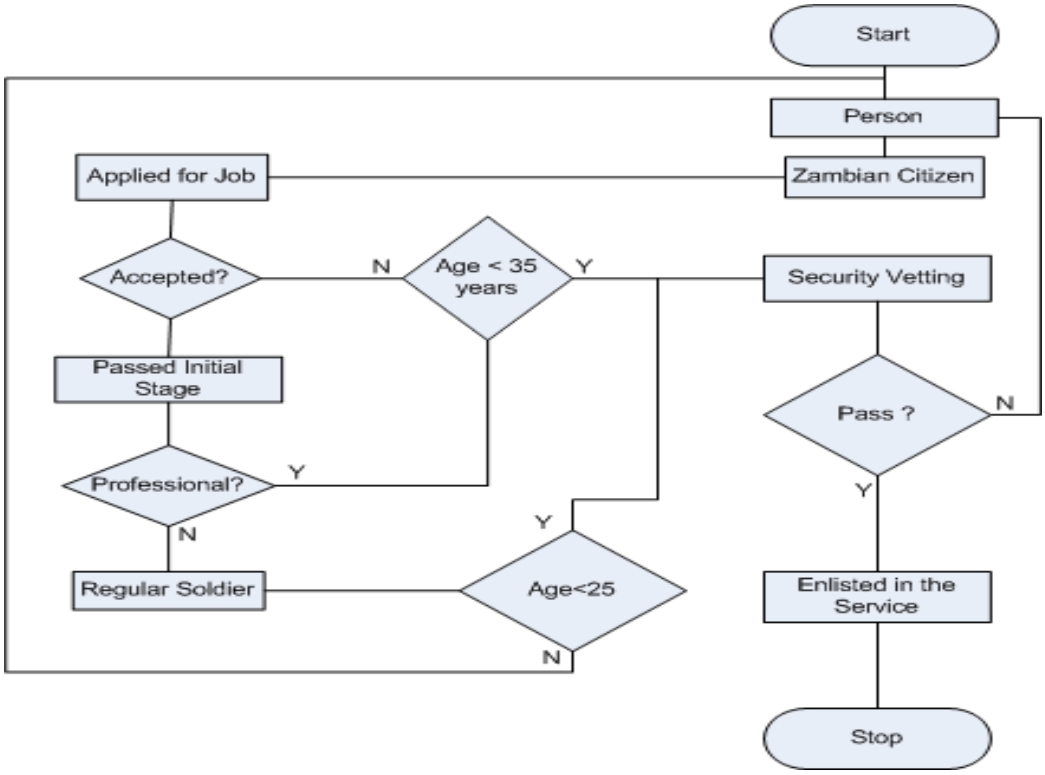


Figure 24:Security Vetting DFD

5.6 **Database Design**

Database design is the process of producing a detailed data model of the project data base structures. The database design included the relational schema and entity relationship database design.

5.6.1 **Relational Schema**

The relational schema describes the names and structure of tables in the relational database (Navathe, 2011). The names of the entities together with their associated attribute are defined. The schema help developers to remove duplicate data and enforce integrate in the database. In nutshell, relational schema describes how real world entities are modelled in the database. The relational schema diagram for AFBSMO is illustrated in Figure 26.

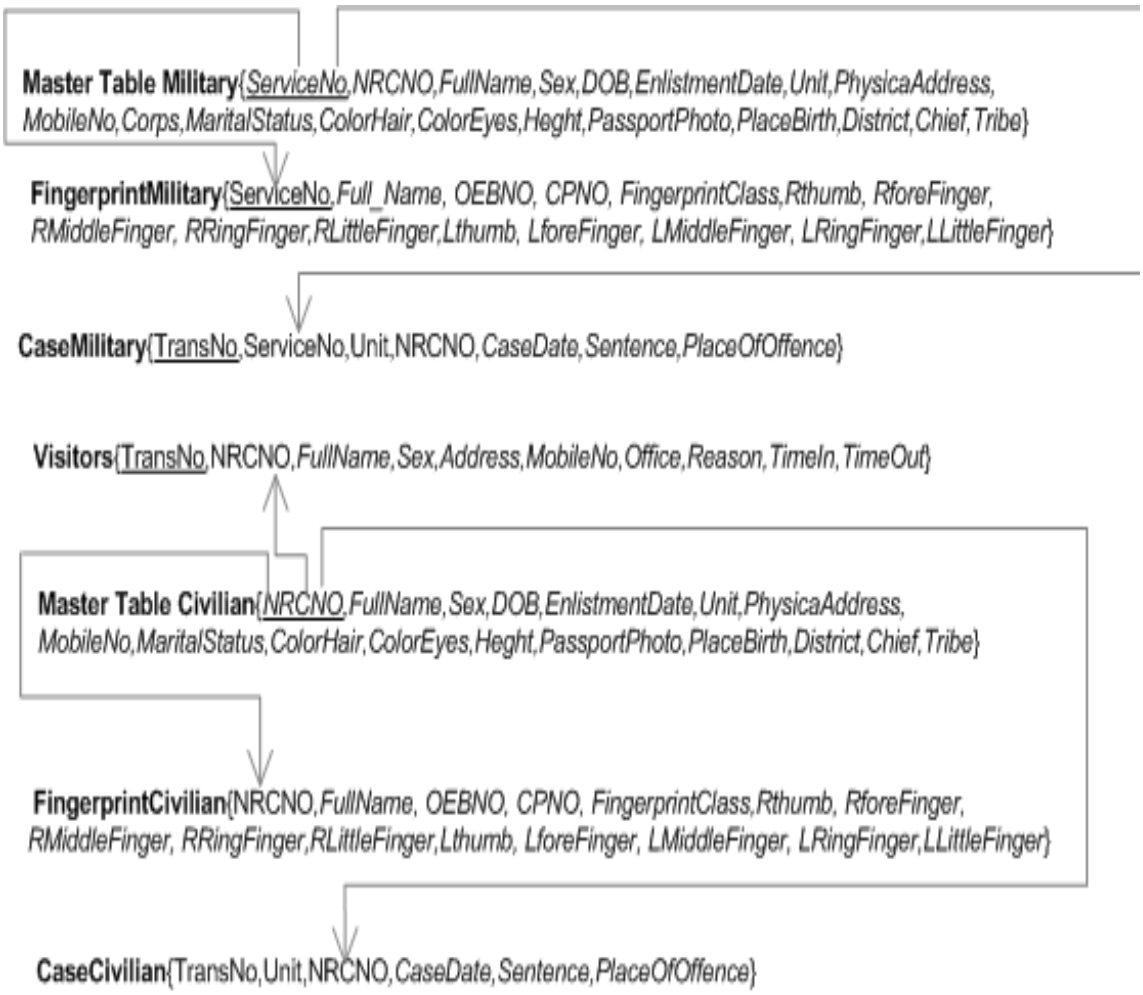


Figure 25:Relational Schema

5.6.2 **Entity Relationship Diagram (ERD)**

An Entity–Relationship (ER) Model was used to define the data or information characteristics of a business domain or its process requirements. The ERD in abstract lend to ultimate implementation in the relational database. The main components of ERD models are entities (things) and the relationships that exist among entities. Figure 27 shows ERD of AFBSMO.

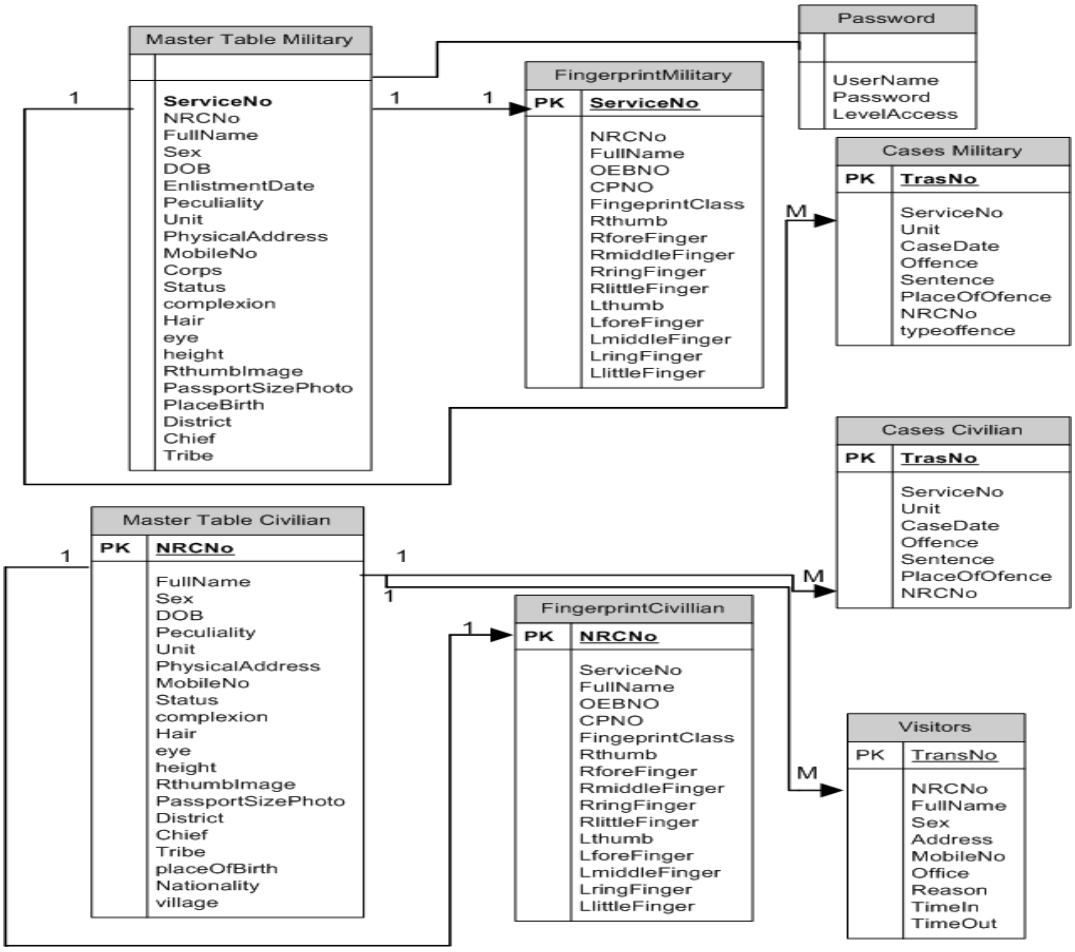


Figure 26: Entity Relationship Diagram (ERD)

5.7 **Structure Chart**

The structure chart design shows the breakdown of the system as per user navigational levels. The first screens on the chart welcome users to the system. After pressing next button users are prompted to enter user name, password and select the security levels of access. In this project users of the system are scalable and are categorized in three application security access levels. The three levels of access are system administrator, sentry and chief investigating officer (CIO). The detailed system navigation design chart is shown in Figure 28.

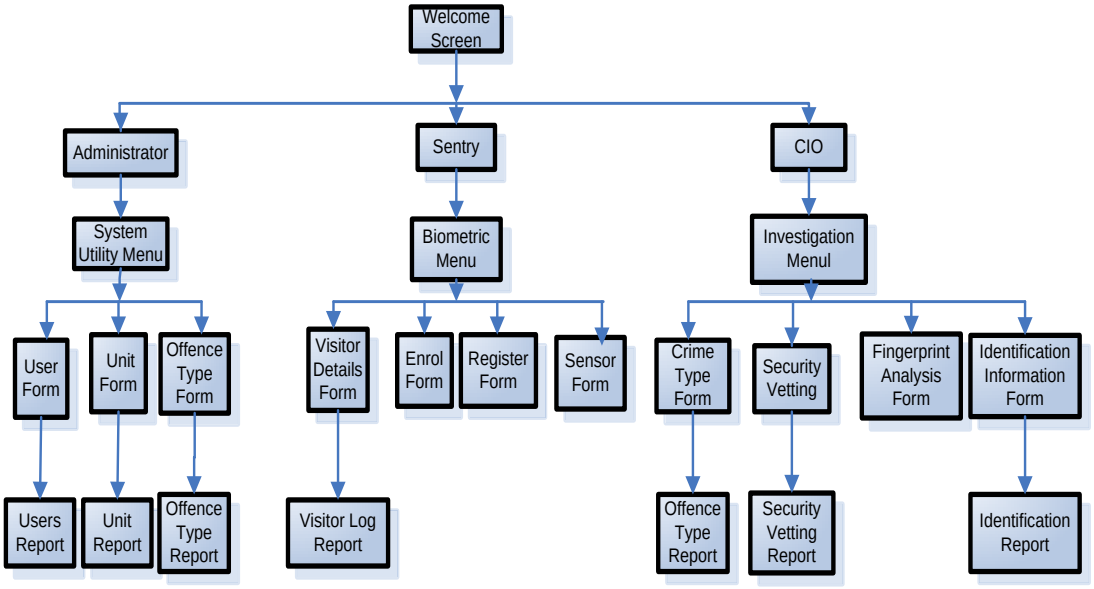


Figure 27:Structure Chart

5.8 Chapter Summary

The design chapter defined the architecture, components, modules and interfaces of developed system. This chapter involved transformation of specified requirements into product architectural design which described the structures and system behaviour. Requirement transformation was modelled logically using component data flow diagram and Entity Relationship Diagram (ERD). The chapter also covered physical design which described user process design aspect which has resulted into system development process. The next chapter described the product development stage.

CHAPTER 6. **DEVELOPMENT**

6.1 **Introduction**

Conceptual design guided the study into AFBSMO development process. Software development involved transformation of design concepts into the actual product. The actual software product was developed in two folds. The first stage was the creation of the database and tables as prescribed in an ERD chapter five paragraph 5.6. After the database was created, forms were developed and respective algorithms were coded. This chapter briefly describes application development stages and constraints resulted from that process.

6.2 **Background Preparations**

Preparation of the environment is the first stage in any application development. It involves setting up the global working environment for both software and database. Without proper environment preparations, development of tasks take-off time is often delayed and may end up frustrating the development process more especially in the three-tier or four-tier architecture. The good programming practice is to ensure that tools and global variables are well set before commencement of actual development.

6.2.1 **MySQL**

MySQL database is an open source application and runs on WAMP sever. The WAMP server was downloaded from MySQL website. The downloaded version was 5.5.24 64 bit. The 64 bit WAMP was installed on windows 7 (64 bit). The global variable port number 80 was changed to port 8080 to avoid conflict with IIS which was running on port 80. By default WAMP Server runs on port 80.

6.2.2 **Visual Studio 2010**

Business logic application development environment was prepared by installing visual studio 2010 and set the language property variable to C#. Visual studio 2010 and C# developer suit were chosen for the research because the fingerprint system development kit (SDK) was developed in C#. MySQL database connector was also downloaded and installed as MySQL to visual studio data connection adaptor.

6.3 **Physical Development**

This section describes the physical design aspects of the research.

6.3.1 **WAMP Configuration**

Wamp server is software heap for the Microsoft Windows operating system. It contains the Apache web server, Open SSL for SSL support, MySQL database and PHP programming language. Before the database was created WAMP server was conFigured to make sure that it was working properly. If the environmental variables were not configured properly WAMP server would fail to start. It normally shows umber colour an indication that the services have not yet started. Otherwise if everything is perfects WAMP Server icon turns green. Some important information about WAMP server and how the database was created in MySQL Database Management System (DBMS) are shown in Figure 29.

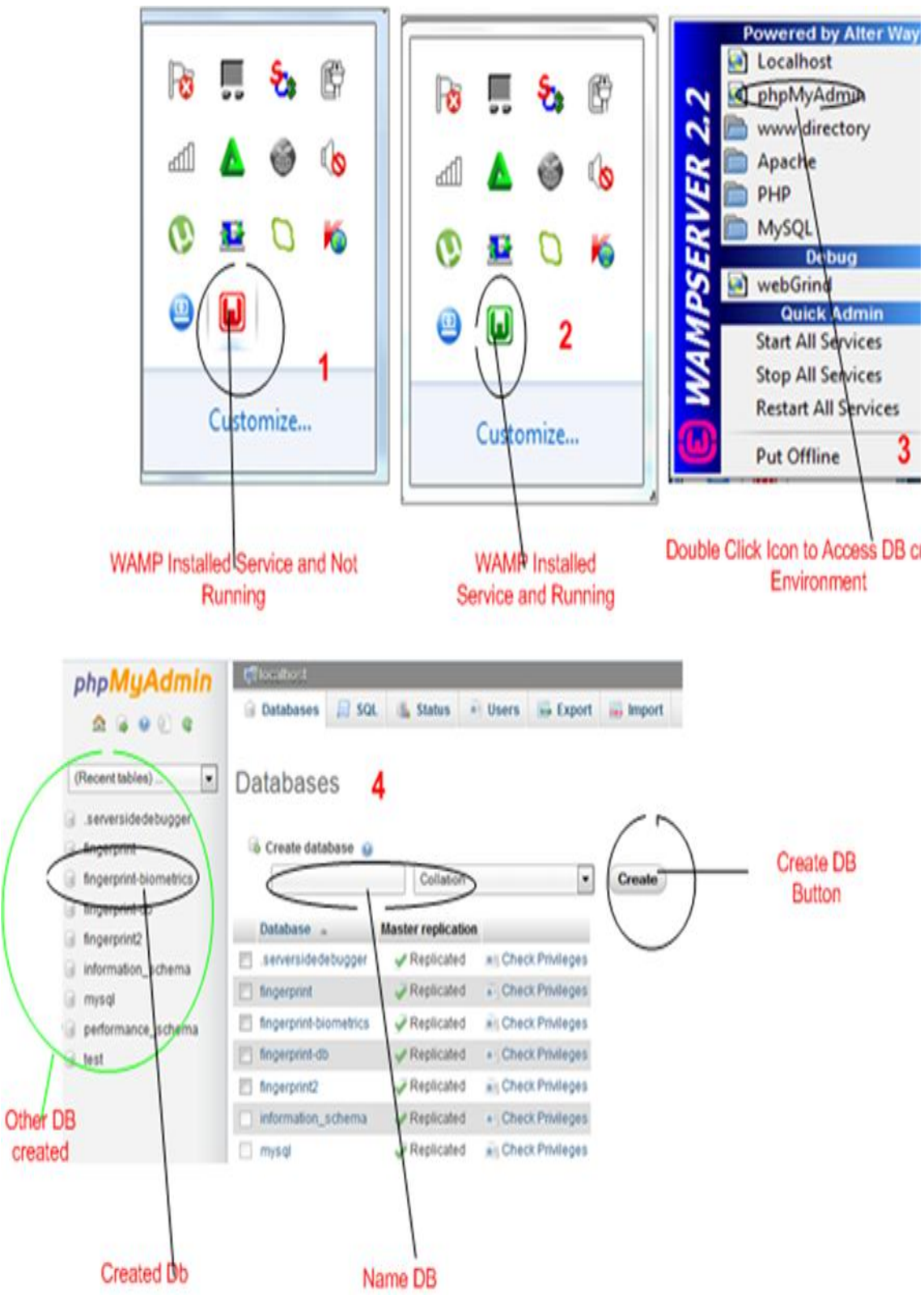


Figure 28:WAMP Configuration

6.3.2 **Database Schema Transformation**

The database schema was transformed into physical objects called database and tables. SQL statements were used to create database, table structures, attributes and associated relationship as expressed in subsequent SQL statements below.

6.3.2.1 **Master Table Military**

This is a main table which is used to store service pesonnel identity record and was created using SQL statement CREATE table as:

CREATE TABLE ` MasterTableMilitary ` (`serviceno` varchar(13) NOT NULL, `nrcno` varchar(11) NOT NULL, `fullname` varchar(30) NOT NULL, `sex` varchar(7) NOT NULL, `DOB` date NOT NULL, `EnlistmentDate` date NOT NULL, `peculiarity` varchar(34) NOT NULL, `unit` varchar(35) NOT NULL, `address` varchar(45) NOT NULL, `mobileno` varchar(13) NOT NULL, `corps` varchar(20) NOT NULL, `sstatus` varchar(17) NOT NULL, `complexion` varchar(20) NOT NULL, `hair` varchar(13) NOT NULL, `eyes` varchar(20) NOT NULL, `height` varchar(10) NOT NULL, `rthumb` blob NOT NULL, `photo` blob NOT NULL, `placebirth` varchar(34) NOT NULL, `district` varchar(30) NOT NULL, `chief` varchar(30) NOT NULL, `tribe` varchar(30) NOT NULL, `village` varchar(34) NOT NULL, `rank` varchar(45) NOT NULL, `mcategory` varchar(45) NOT NULL, PRIMARY KEY (`serviceno`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

6.3.2.2 **Master Table Civilian**

This table is main table for storing civilian identity details. It includes relational attributes and stipulated data type. The table was created using CREATE TABLE mastertablecivilian as:

CREATE TABLE `mastertablecivilian` (`nrcno` varchar(15) NOT NULL, `fullname` varchar(35) NOT NULL, `sex` varchar(15) NOT NULL, `address` varchar(50) NOT NULL, `DOB` date NOT NULL, `peculiarity` varchar(40) NOT NULL, `rthumb` blob NOT NULL, `photo` blob NOT NULL, `unit` varchar(45) NOT NULL, `mobileno` int(11) NOT NULL, `height` varchar(50) NOT NULL, `hair` varchar(34) NOT NULL, `eyes` varchar(34) NOT NULL, `complexion` varchar(35) NOT NULL, `placebirth` varchar(34) NOT NULL, `district` varchar(30) NOT NULL, `village`

varchar(44) NOT NULL, `chief` varchar(25) NOT NULL, `tribe` varchar(20) NOT NULL, `status` varchar(35) NOT NULL, `nationality` varchar(35) NOT NULL, `EnlistmentDate` varchar(45) NOT NULL, `pathrthumb` varchar(45) NOT NULL, `pathphoto` varchar(24) NOT NULL, PRIMARY KEY (`nrcno`), UNIQUE KEY `nrcno` (`nrcno`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

6.3.2.3 **Fingerprint Military**

Fingerprint military table store service personnel fingerprints. The total of ten fingerprints is stored per each individual service personnel. The table was created as follows:

CREATE TABLE `fingerprintmilitary` (`serviceno` varchar(13) NOT NULL, `nrcno` varchar(23) NOT NULL, `fullname` varchar(30) NOT NULL, `oebno` varchar(34) NOT NULL, `cpno` varchar(34) NOT NULL, `fingerprintclass` varchar(23) NOT NULL, `rthumb` blob NOT NULL, `rforefinger` blob NOT NULL, `rmiddlefinger` blob NOT NULL, `rringfinger` blob NOT NULL, `rlittlefinger` blob NOT NULL, `lthumb` blob NOT NULL, `lforefinger` blob NOT NULL, `lmiddlefinger` blob NOT NULL, `lringfinger` blob NOT NULL, `llittle` blob NOT NULL, PRIMARY KEY (`serviceno`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

6.3.2.4 **Fingerprint Civilian**

Fingerprint civilian table store service personnel fingerprints. The total of ten fingerprints is stored per each individual personnel. The table was created as follows:

CREATE TABLE `FingerPrintCevilian` (`serviceno` varchar(13) NOT NULL, `nrcno` varchar(23) NOT NULL, `fullname` varchar(30) NOT NULL, `OEBNO` varchar(34) NOT NULL, `cpno` varchar(34) NOT NULL, `fingerprintclass` varchar(23) NOT NULL, `rthumb` blob NOT NULL, `rforefinger` blob NOT NULL, `rmiddlefinger` blob NOT NULL, `rringfinger` blob NOT NULL, `rlittlefinger` blob NOT NULL, `lthumb` blob NOT NULL, `lforefinger` blob NOT NULL, `lmiddlefinger` blob NOT NULL, `lringfinger` blob NOT NULL, `llittle` blob NOT NULL, PRIMARY KEY (`nrcno`), UNIQUE KEY `serviceno` (`serviceno`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

6.3.2.5 **Case Military**

Case Military table stores detail of cases (offences) that the service personnel has commited during his service. The table was created as follows:

CREATE TABLE `casemilitary` (`transno` int(11) NOT NULL AUTO_INCREMENT, `serviceno` varchar(35) NOT NULL, `nrcno` varchar(11) NOT NULL, `unit` varchar(25) NOT NULL, `casedate` date NOT NULL, `offence` varchar(50) NOT NULL, `typeoffence` varchar(25) NOT NULL, `sentence` varchar(45) NOT NULL, `Description` varchar(250) NOT NULL, `fullname` varchar(45) NOT NULL, `caseno` varchar(23) NOT NULL, PRIMARY KEY (`transno`), UNIQUE KEY `serviceno` (`serviceno`), UNIQUE KEY `caseno` (`caseno`), KEY `serviceno_2` (`serviceno`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;ALTER TABLE `casemilitary` ADD CONSTRAINT `casemilitary_ibfk_1` FOREIGN KEY (`serviceno`) REFERENCES `mastertablemilitary` (`serviceno`);

6.3.2.6 **Case Civilian**

Case Civilian table stores detail of cases (offences) that the civilian personnel has commited during the service in military. The table was created as follows:

CREATE TABLE `casecivilian` (`transno` int(11) NOT NULL AUTO_INCREMENT, `nrcno` varchar(35) NOT NULL, `unit` varchar(35) NOT NULL, `casedate` date NOT NULL, `offence` varchar(40) NOT NULL, `typeoffence` varchar(40) NOT NULL, `Description` varchar(245) NOT NULL, `sentence` varchar(35) NOT NULL, `caseno` varchar(23) NOT NULL, `fullname` varchar(45) NOT NULL, PRIMARY KEY (`transno`), KEY `nrcno` (`nrcno`), KEY `nrcno_2` (`nrcno`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;ALTER TABLE `casecivilian` ADD CONSTRAINT `casecivilian_ibfk_5` FOREIGN KEY (`nrcno`) REFERENCES `mastertablecivilian` (`nrcno`);

6.3.2.7 **Password**

Password table stores users' authentication details. When login the system users supply authentication parameter which are stored in this table. The table was created as follows:

CREATE TABLE `password` (‘username` varchar (45) NOT NULL, `password` varchar (30) NOT NULL,`accesslevel` varchar(34) NOT NULL, PRIMARY KEY (`username`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

6.4 Application Development Environment

Application Development Environment refers to visual studio 2010 development settings.

6.4.1 **Visual Studio 2010**

During application creation Microsoft visual studio 2010 development environment was used with C# programming environment. Visual studio 2010 is the Microsoft appropriated development environment that was used to create objects such as forms, reports, classes, application settings and references etc. Visual studio 2010 is common language developer suit which is used for client side (form and report) and server side programming. Visual studio is language independent platform. Language independent means any programming language can be employed for server side processing. The language such as C, C++, Java, Perl and Visual basic .Net can be used in visual studio as server side programming or business logic. Visual studio is flexible because it supports many programming languages and developers are able reuse any of the programs components written in other languages through referencing (pointers) or directive (dll file import).

6.4.2 **Configurations**

Configuration in this case is referred to global and local setting of the developed application. Local settings are configuration of an application within the application. It describes the environment under which the developed objects interact with each other within an application. An example of this is how objects fingerprint analysis (form) implement fingerprint image normalisation algorithm. Global settings refer to how application objects utilise external objects. An example to this is how the developed application communicates with fingerprint sensor and database. The configuration parameters of each visual studio application are stored in the file called app.config.

6.4.3 **App.Config File**

App.config file is an acronym for Application configuration file. An app.config file is an extensive mark-up language (XML) file. As alluded to in earlier section app.config is used to store configuration of any application type developed in visual studio. The developed application parameters (settings) were for desktop application as shown in Figure app.config.file. This file is normally large in web application because it stores information such as connection strings, compilation assemblies, application authentication mode, controls and compiler specifications which is sometime referred to as codedom. This study was conducted on dot NET Framework version 4.0 as shown in app.config file given below.

---------- ***App.config file*** *shows configuration parameters of the system*----------

app.config file

<?xml version="1.0"?>

<configuration>

<configSections>

</configSections>

<connectionStrings>

<add

name="CSharpAFBSMO.Properties.Settings.fingerprint_biometricsConnectionString

"

connectionString="server=localhost;user id=root;database=fingerprint-biometrics"

providerName="MySql.Data.MySqlClient" />

</connectionStrings>

<startup><supportedRuntime version="v4.0"

sku=".NETFramework,Version=v4.0"/></startup></configuration>

6.5 **Process Communication**

Process communication refers to how various developed components of the system converse with each other. The developed components communicate or interact with other services or processes using referencing methods (pointers) or through dynamic link library (ddl) file importation as expressed in code (i) and (ii) Process Communication.

---------- *Code below shows Microsoft foundational library access by the system---*

Process Communication Code

using System.Drawing.Imaging;

using System.IO;

using System.Data.Odbc;

using MySql.Data.MySqlClient;

(ii) Process interaction ddl methods

Dynamic Data link (ddl) Method

class ZKFPCap

{

public const string ZKFPRI_NAME = @"ZKFPCap.dll";

source: (Sdk, Guide, & Reserved, 2012)

6.6 Methods and Algorithm Implementation

These sections describe how various algorithms were implemented in an application.

6.6.1 Database Connection

Database connection in this study was implemented using MySQL data adaptor provided by MySql.Data.MySqlClient library. Library invocation is as shown in paragraph 6.1. The Actual database name specified in RDBMS is **fingerprint-biometrics** accessed on port 3306 of the local server. The owner (username) of created database schema is root and the schema password is1q45ghh1 as given in the code below:

---------- *Code below shows Mysql database connection*----------

Database Connection Code

```
using System.Data.Odbc;
using MySql.Data.MySqlClient;
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password= 1q45ghh1;";
```

6.6.2 **Fingerprint Scanner Algorithm**

The fingerprint scanner algorithm is responsible for fingerprint connection and device management. The algorithm functionality was defined in ZKFPclass. ZKFPclass invoked ZKFP.dll library which contain fingerprint sensor Application Program Interface (API). The methods invoked in class include SensorInit (), SensorFree (), SensorOpen, SensorClose (), SensorGetCount (), SensorGetVersion (), SensorGetParameter () SensorSetParameterEx () and SensorCapture (). The class shown in the code below define the fingerprint scanner connection methods (Sdk et al., 2012).

--- *Code below shows ZKFPCap class that initialises functionalities of fingerprint sensor*-----

Fingerprint Scanner Operation Algorithm(ZKFPCap Class)

```
class ZKFPCap
{
publicconststring ZKFPRI_NAME = @"ZKFPCap.dll";
---Code Removed here
[DllImport(ZKFPRI_NAME, EntryPoint = "sensorGetCount")]
publicstaticexternint sensorGetCount();
[DllImport(ZKFPRI_NAME, EntryPoint = "sensorGetVersion")]
publicstaticexternint sensorGetVersion(byte[] version, int len);
---Code Removed here
[DllImport(ZKFPRI_NAME, EntryPoint = "sensorCapture")]
publicstaticexternint sensorCapture(IntPtr handle, byte[] imageBuffer, int
imageBufferSize);

}
```

Source :(Sdk et al., 2012)

6.6.3 **Fingerprint Processing Algorithm**

The fingerprint processing algorithm is implemented in the class ZKFinger10. ZKFinger10 class interfaces ZKFinger10 data link library. The library contains methods include BIOKEY_INIT(),BIOKEY_CLOSE(),BIOKEY_SET_PARAM(),BIOKEY_MATCHING_PARAM(),BIOKEY_MACHING_PARAM(),BIOKEY_GETLASTERROR(), BIOKEY_EXTRACT(),BIOKEY_DB_ADD(),BIOKEY_DB_DEL(), BIOKEY_DB_COUNT(), BIOKEY_DB_CLEAR(),BIOKEY_IDENTITY_TEMP() and BIOKEY_VERIFY().

Fingerprint processing algorithm is given in class ZKFinger10 shown in code below.

--- *Code below shows ZKFinger10 which imports dll libraries for fingerprint processing*---

Fingerprint Processing Code (ZKFinger10)

```
---Code Removed here
class ZKFinger10
{
publicconststring ZKFPRI_NAME = @"ZKFinger10.dll";
publicconstint THRESHOLD_MIDDLE = 55; // 1:N recommendation threshold
publicconstint THRESHOLD_LOW = 35;    // 1:1 recommendation threshold
[DllImport(ZKFPRI_NAME, EntryPoint = "BIOKEY_INIT")]
publicstaticexternIntPtr BIOKEY_INIT(int License, short[] isize, byte[] Params,
byte[] Buffer, int ImageFlag);
[DllImport(ZKFPRI_NAME, EntryPoint = "BIOKEY_MATCHINGPARAM")]
publicstaticexternint BIOKEY_MATCHINGPARAM(IntPtr Handle, int speed, int
threshold);
[DllImport(ZKFPRI_NAME, EntryPoint = "BIOKEY_GETLASTQUALITY")]
```

```
publicstaticexternint BIOKEY_GETLASTQUALITY();
---Code Removed here
[DllImport(ZKFPRI_NAME, EntryPoint = "BIOKEY_IDENTIFYTEMP")]
publicstaticexternint BIOKEY_IDENTIFYTEMP(IntPtr Handle, byte[] Template,
refint TID, refint Score);
[DllImport(ZKFPRI_NAME, EntryPoint = "BIOKEY_VERIFY")]
publicstaticexternint BIOKEY_VERIFY(IntPtr handle, byte[] Template1, byte[]
Template2);

}
}
```

Source: (Sdk et al., 2012)

6.6.4 **Bitmap Formatting**

The fingerprint image format was developed in bitmap format class. BitmapFormat class was defined by three (03) structures namely BITMAPFILEHEADER, MASK and BITMAPINFOHEADER as shown in bitmap formatting code. The data type in each structure was defined public in order to anywhere within an application. Bitmap class was dissected further into other methods and functions.

Bitmap Image formatting class

```
publicclass BitmapFormat
{
publicstruct BITMAPFILEHEADER
{
publicushort bfType;
publicint bfSize;
publicushort bfReserved1;
publicushort bfReserved2;
publicint bfOffBits;
}
public struct MASK
{
```

```
publicbyte redmask;
publicbyte greenmask;
publicbyte bluemask;
publicbyte rgbReserved;
}

publicstruct BITMAPINFOHEADER
{
publicint biSize;
publicint biWidth;
publicint biHeight;
publicushort biPlanes;
publicushort biBitCount;
publicint biCompression;
publicint biSizeImage;
publicint biXPelsPerMeter;
publicint biYPelsPerMeter;
publicint biClrUsed;
publicint biClrImportant;
}
```

Source: (Sdk et al., 2012)

6.6.4.1 **Rotation Algorithm**

Rotational algorithm provides image orientation by computing specific image size and formatting. Rotational algorithm once developed computed the image structure orientation on the sensor and screen. Rotation code is shown as illustrated in the code below:

--- *Code below shows RotatePic class position the fingerprint pictures on the form* ---

```
public static void RotatePic(byte[] BmpBuf, int width, int height, ref byte[] ResBuf)
{
int RowLoop = 0;
int ColLoop = 0;
```

```
int BmpBuflen = width * height;
try
{
for (RowLoop = 0; RowLoop < BmpBuflen; )
{
for (ColLoop = 0; ColLoop < width; ColLoop++)
{
ResBuf[RowLoop + ColLoop] = BmpBuf[BmpBuflen – RowLoop – width +
ColLoop];
}
RowLoop = RowLoop + width;
}}}
```

Source: (Sdk et al., 2012)

6.6.4.2 **Structure Conversion Algorithm**

The computed image structure is converted to byte data structure by method Structure To Bytes given below. The memory is managed by .NET marshal class which is provided for by System Runtime.Interop Services.Marshalin .NET 4.0 frameworks (Microsoft, 2015). The structure coversion is illustrated in below:

--- *Code below shows* public static byte [] StructToBytes *which performs marshalling process. .**NET**, **marshalling** is used to convert .**NET's** data into data suitable for storage in the memory---*

Structure to Bit Conversion Method

```
public static byte[] StructToBytes(object StructObj, int Size)
{
int StructSize = Marshal.SizeOf(StructObj);

----Some code Removed here

for (Loop = 0; Loop < StructSize; Loop++)
{
if (Loop != 2 && Loop != 3)
```

{

NewBytes[Count] = GetBytes[Loop];

Count++;

}}

return NewBytes;

}

else

{

return GetBytes;

}}

catch (Exception ex){ return GetBytes; } }

Source: (Sdk et al., 2012)

6.6.4.3 **Bitmap Image Regeneration**

The fingerprint image is fitted in to compute byte structure using the method Get Bitmap. Get Bitmap image regeneration function is defined by three structure headers namely BITMAPFILEHEADER, BITMAPINFOHEADER and MASK. The developed get Get Bitmap function is illustrated in the code below:

-----*Use the GetBitmap function to obtain a particular image in the image list as a bitmap object.* ------

GetBitmap Fuction

publicstaticvoid GetBitmap(byte[] buffer, int nWidth, int nHeight, refMemoryStream ms)

{{

BITMAPFILEHEADER BmpHeader = newBITMAPFILEHEADER();

BITMAPINFOHEADER BmpInfoHeader = newBITMAPINFOHEADER();

MASK[] ColorMask = newMASK[m_nColorTableEntries];

---- coded removed here

BmpHeader.bfOffBits = 10 + Marshal.SizeOf(BmpInfoHeader) +

BmpInfoHeader.biClrUsed * 4;

BmpHeader.bfSize = BmpHeader.bfOffBits + ((((BmpInfoHeader.biWidth *

BmpInfoHeader.biBitCount + 31) / 32) * 4) * BmpInfoHeader.biHeight);

---- coded removed here

ms.Write(StructToBytes(BmpInfoHeader, Marshal.SizeOf(BmpInfoHeader)), 0,

Marshal.SizeOf(BmpInfoHeader));

for (ColorIndex = 0; ColorIndex < m_nColorTableEntries; ColorIndex++)

{

ColorMask[ColorIndex].redmask = (byte)ColorIndex;

ColorMask[ColorIndex].greenmask = (byte)ColorIndex;

ColorMask[ColorIndex].bluemask = (byte)ColorIndex;

ColorMask[ColorIndex].rgbReserved = 0;

ms.Write(StructToBytes(ColorMask[ColorIndex],

Marshal.SizeOf(ColorMask[ColorIndex])), 0,

Marshal.SizeOf(ColorMask[ColorIndex]));

}

----coded removed here

6.6.4.4 **Bitmap Image Write Algorithm**

Bitmap image write algorithm is the final bevioural characteristic of the bitmap class. The output image of this method is black and white reconstructed fingerprint image. Write Bitmap was implemented as shown in code below:

----- *Below code write or convert image to bitmap file or format*-----

Bitmap write Algorithm

publicstaticvoid WriteBitmap(byte[] buffer, int nWidth, int nHeight)

{

---declaration code removed

try

{

--- Header Instance declaration

```
BmpHeader.bfType = 0x4D42;
BmpHeader.bfOffBits = 14 + Marshal.SizeOf(BmpInfoHeader) +
BmpInfoHeader.biClrUsed * 4;
BmpHeader.bfSize = BmpHeader.bfOffBits + ((((BmpInfoHeader.biWidth *
BmpInfoHeader.biBitCount + 31) / 32) * 4) * BmpInfoHeader.biHeight);
BmpHeader.bfReserved1 = 0;
BmpHeader.bfReserved2 = 0;
Stream FileStream = File.Open("finger.bmp", FileMode.Create, FileAccess.Write);
BinaryWriter TmpBinaryWriter = newBinaryWriter(FileStream);
TmpBinaryWriter.Write(StructToBytes(BmpHeader, 14));
TmpBinaryWriter.Write(StructToBytes(BmpInfoHeader,
Marshal.SizeOf(BmpInfoHeader)));
for (ColorIndex = 0; ColorIndex < m_nColorTableEntries; ColorIndex++)
{
------ ColorMask[ColorIndex] conversion code
TmpBinaryWriter.Write(StructToBytes(ColorMask[ColorIndex],
Marshal.SizeOf(ColorMask[ColorIndex])));
}
RotatePic(buffer, nWidth, nHeight, ref ResBuf);
TmpBinaryWriter.Write(ResBuf);
FileStream.Close();
TmpBinaryWriter.Close();
}
catch (Exception ex)
{}}
```

Source: (Sdk et al., 2012)

6.6.5 **Insert Data Method**

The insert data method was used to populate relation in the database. The methods include database connection definitions, data command definitions and data reader definitions as shown in insert data code. The method also includes function to open connection and function to Execute Reader. The routine for insert data method is show in the code below:

----- *Below code insert data into the database table*-----

Insert Data Method

Connection String

INSERT INTO TABLE_NAME (column1, column2, column3,...columnN)]
VALUES (value1, value2, value3,...valueN);

MySqlConnection();

MySqlCommand();

MySqlDataReader;

{

conn.Open();

myreader = cmd.ExecuteReader();

}

6.6.6 **Update Method**

Update method is used to change the value of data stored in the table of database. Update Data Method include database connection definitions, data command and data reader definitions as shown in update data method. The method also includes function to open connection and function to Execute Reader as illustrated in update method below:

----- *Below code update data into the database table*-----

Update Method

MySqlDataReader ;

Connection String;

UPDATE *table_name* SET *column1=value1,column2=value2,...* WHERE

some_column=some_value;

MySqlConnection

MySqlCommand

try

{

con.Open();

rddata = cmd1.ExecuteReader();

6.6.7 **Delete Method**

The delete method is used to erase record from the Database. It includes definitions such as data connection string, MySQL command, open connection function and ExecuteNonQuery function. The syntax for delete method is given in the code below:

----- *Below code delete data into the database table*-----

Delete method

MySqlConnection

DELETE FROM *table_name* WHERE *some_column=some_value*;

{

con.Open();

cmd1.ExecuteNonQuery();

}

finally

{

con.Close();

6.6.8 **Method to Adjust Brightness of Image**

The image brightness method was implemented to increase amount of light for reflexion. The brighter the item the more light it can reflect, the dimmer an item is the less light it can reflect. Brightness method implemented deals with RGB color space. It takes in an image and an integer between -255 and 255 in matrix to modify the brightness of pixel. Brightness was implemented to aid human visual sensation (Fords & Roberts, 1998b). The code for changing image brightness is shown below:

----- *Below code adjust brightnessness of image on the form* -----

Image AdjustBrightness_Methods

publicstaticBitmap AdjustBrightness(Bitmap Image, int Value)

```
{
---declaration  and other code removes here
---Matrix  to  increase brightness of an image
float[][] FloatColorMatrix ={
newfloat[] {1, 0f, 0f, 0f, 0f},
newfloat[] {0f, 1, 0f, 0f, 0f},
newfloat[] {0f, 0f, 1, 0f, 0f},
newfloat[] {0f, 0f, 0f, 1f, 0f},
newfloat[] {FinalValue, FinalValue, FinalValue ,0f,1f}
};
---some code removes here
```

6.6.9 **Method to Adjust Contrast of an Image**

Contrast in spectrum involves changing image values distribution to cover a wide range in RGB colour spectrum. The method uses function g(f) to generate new image B from a given image A via B (I,J). The function g(f) operate on each image pixel independently. All pixels values with original gray level image are changed to level g(f) by applying code illustrated below:

----- *Below code adjust Contrast of image on the form* -----

Contrast Adjustment Method

```
publicstaticbool Contrast(Bitmap b, sbyte nContrast)
{
if (nContrast < -100) returnfalse;
if (nContrast > 100) returnfalse;
double pixel = 0, contrast = (100.0 + nContrast) / 100.0;

contrast *= contrast;
int red, green, blue;
// GDI+ still lies to us - the return format is BGR, NOT RGB.
BitmapData bmData = b.LockBits(newRectangle(0, 0, b.Width, b.Height),
ImageLockMode.ReadWrite, PixelFormat.Format24bppRgb);
```

```
int stride = bmData.Stride;
System.IntPtr Scan0 = bmData.Scan0;
unsafe
{
byte* p = (byte*)(void*)Scan0;
int nOffset = stride - b.Width * 3;
for (int y = 0; y < b.Height; ++y)
{
for (int x = 0; x < b.Width; ++x)
{
blue = p[0];
green = p[1];
red = p[2];
---Code remove
p += 3;
}
p += nOffset;
}
}
UnlockBits(bmData);
returntrue;
}
```

6.6.10 **Method to Count Number of Pixel in Image**

The method of counting the number of black and white pixel was employed in this research. Pixel Count Function input image and output number of black and white pixels and then calculates the total number of Black and white pixels. The function Count Pixels has two parameters Bitmap and colour as given in the code below.

----- *Below code count no of pixels in an image on the form* -----

Pixel Count Function

```
privateint CountPixels(Bitmap bm, Color target_color)
```

```
{
int matches = 0;
for (int y = 0; y < bm.Height; y++)
{
for (int x = 0; x < bm.Width; x++)
{
if (bm.GetPixel(x, y) == target_color) matches++;
}}
return matches;
}
```

6.7 Chapter Summary

The Chapter described the construction process of the new system which involved technical setting, database development and application coding. The next task of the research was to test and record the result of developed application.

CHAPTER 7. **RESULTS AND DISCUSSION**

7.1 **Introduction**

This chapter unveils the results of study conducted at Army Headquaters military police Unit Lusaka Zambia. The results of this study are associated with modalities to improve security in the barracks through development and deployment of security application that integrate fingerprint biometrics in military police database. Prio to system development, data was collected through unstructured interview, observation, record inspection (Army forms e.g. FZA 12A) and literature review. The requirements were derived by using various data analysis models, product (software) designed and implemented. The results in this research are expressed in terms functionalities embodied in the developed system as prescribed by analysis and design chapter of this study.

7.2 **System Functionality**

According to Website techtarget.com Functionality is the sum or any aspect of what a product, such as a software application or computing device, can do for a user. The developed system functionalities are organised according defined system users (actor) type. System users may be defined as sentries, Criminal Investigation Officer (CIO) and system administrator. What determine user grouping are roles, duties and responsilities a particular user performs in an organisation. For example military police officer's (men) are incharge of verifying identity of visitors at the military establishment gates and are assign the permission to access sentry module of the developed system. The developed application has three modules or views namely sentry, CIO and System administrator modules. The major role of military police in this application is to verify identity of visitors as mentioned above through the usage visitor NRC and biometric identifier (fingerprint scanner). CIO is responsible for military police database mainly new employee security vetting and forensic investigation. Users defined as System administrator as overall permmission to access any module in the system. System Administrators have been given super user permission to enable them mornitoring the function of the system.

7.3 **Functional Requirements Implementation**

Functional requirements for the developed system were defined chapter 4 paragraphs 4.4.3.10, designed in chapter 5 paragraph 5.6 and coded in c# computer language as illustrate in chapter 6 (development chapter). These functionalities were embodied in developed system include login (application security screen), connect fingerprint sensor, fingerprint enrolment, human identification, human identity verification, capture fingerprint, criminal vetting, capture military / civilian personnel record, capture military /civilian personnel case record, and fingerprint analysis. The screen shot below illustrates the results of the study:

7.3.1 **Login**

Login form is the first screen the user access when intending to access the system resources. The developed system has capability to define three types of users that is users defined to operate at the main gate (Sentry), users defined to operate at military police Headquarters for criminal investigating office (CIO) and System Administrator (Administrator). Login screen is used to verify the authorize user of the system. The snap shot of the Login form is given in Figure 29.

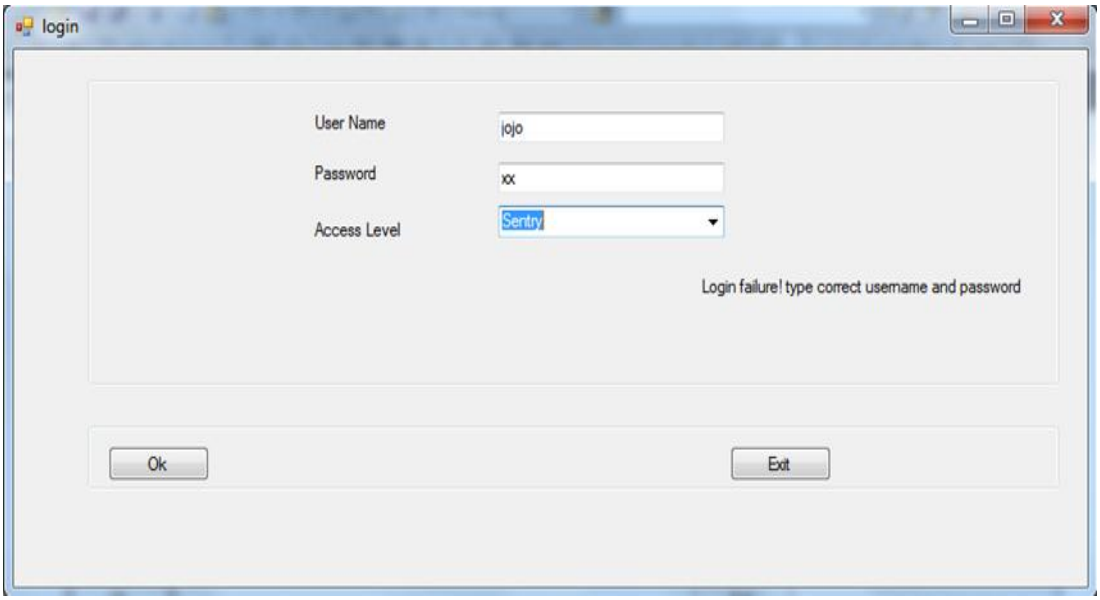


Figure 29: User Login Form

7.3.2 **Fingerprint Sensor Connection**

Fingerprint sensor connection functionality is used to connect a fingerprint sensor to the developed system. This application gives freedom for the operator to choose the type of scanner connected to the computer system for the application to detect the exact type of device drivers in use. If an operator chooses the wrong type of scanner which is not connected to the computer system, an application does not detect the sensor and instead generate error message. Figure 30 shows choose the scanner functionality implementation.

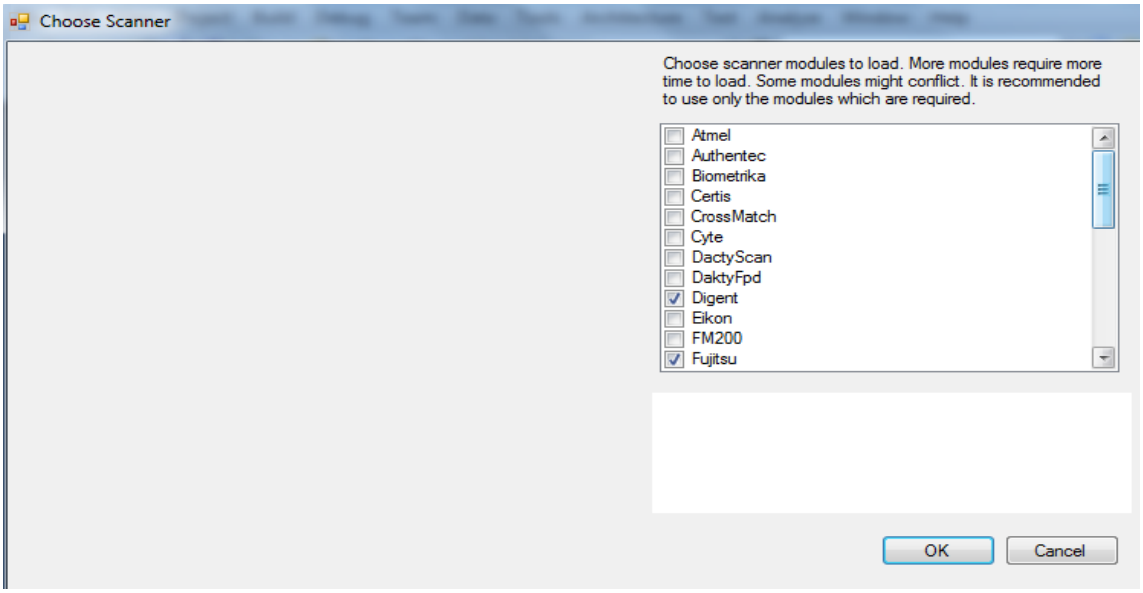


Figure 30: Fingerprint Scanner Connection Form

7.4 **Sentry Module**

Sentry module was developed for setries deployed at the main gates of the barracks. As discussed in introduction chapter of this study, the current mechanism of verifying identity of individuals visiting the barracks in Zambia Army is manual and same chapter has clearly atticulated the problems associated to this approach. In literature review, we learnt that security forces of the developing countries adopted biometric systems to solve similar problems. The sentry module was developed as an addon feature to improve human identity verification in the army. The method of developing

fingerprint biometric technology was learnt from literature review. The sentry module implemented fingerprint enrollement and identification requirements.

7.4.1 **Fingerprint Enrolment**

The fingerprint enrolment was developed in order to capture and store human fingerprint image for utilisation in human identification .The template creation unit organizes one or more feature sets into an enrolment template that will be saved in some persistent storage. The enrolment template is sometimes also referred to as a reference. Enrolment is the first stage of fingerprint recognition process as learnt from literature review. This requirement was implemented and the results are shown in Figure 31.

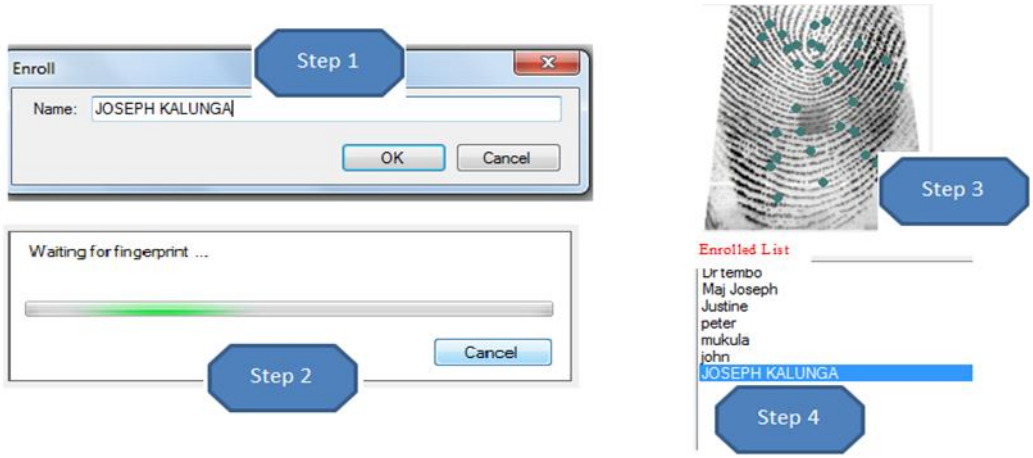


Figure 31: Fingerprint Enrollment Process Results

7.4.2 **Fingerprint Identification**

The fingerprint human identification system requirements were implemented by developing verification module. Operator verification results exist only in two states verified and not verified. The implementation of verification algorithm result is shown in Figure 32.

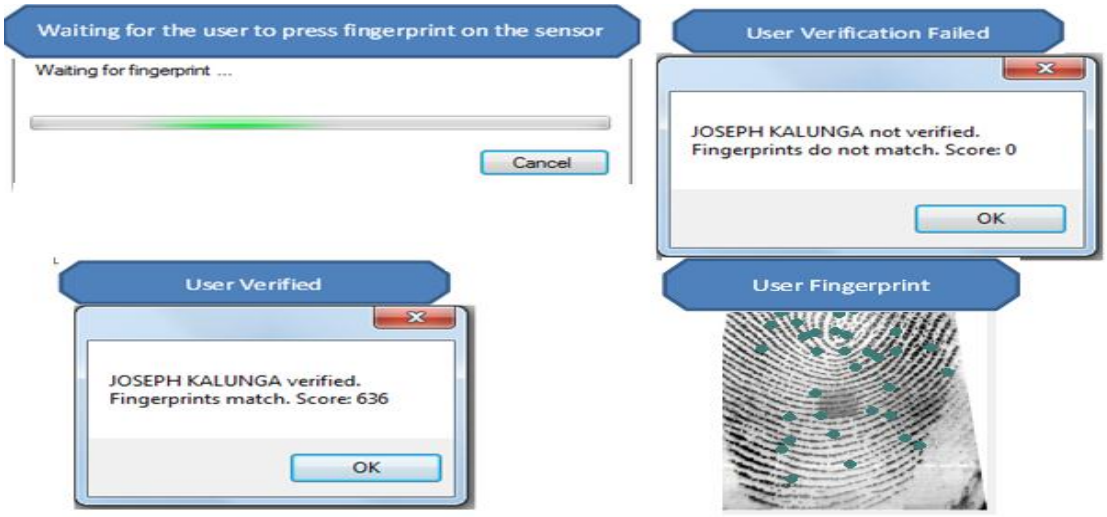


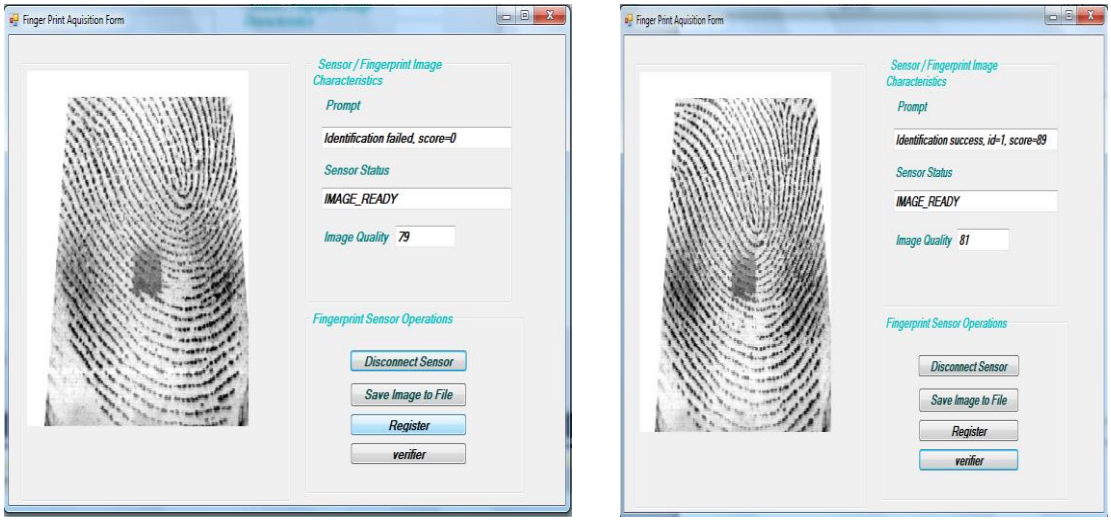
Figure 32: Fingerprint Verification Results

7.5 Criminal Investigation Module

Criminal investigation usually probes serious violation of military law within the army. It operates under the command of the Army provost marshal. Criminal investigation module encompasses criminal record and offences concerning military personnel. This module includes the functionality of conducting criminal vetting of potential military employees. It involves fingerprint capturing, storage and recongnition for criminal investigations purposes.

7.5.1 Fingerprint Capturing

Fingerprint acquisition form is used to capture fingerprint and store them in to the database. A total of ten (10) human fingerprint images are captured per donor using the form shown on Figure 34. Each captured fingerprint may be compared by the system against the same type of finger which is pressed on the sensor. For example a stored thumb image may be compared against live thumb captured during identification process as shown on the Figure 33 (a) and (b). This type of deliverable may be used during criminal investigation process to identify the fingerprints of a criminal. The same process is also conducted during security vetting of new employees.



(a) (b)

Figure 33: (a) Fingerprint Image Acquisition Form (b) Fingerprint Identification Results

7.5.2 **Criminal Vetting**

The form for security vetting has been automated as shown in Figure 34.

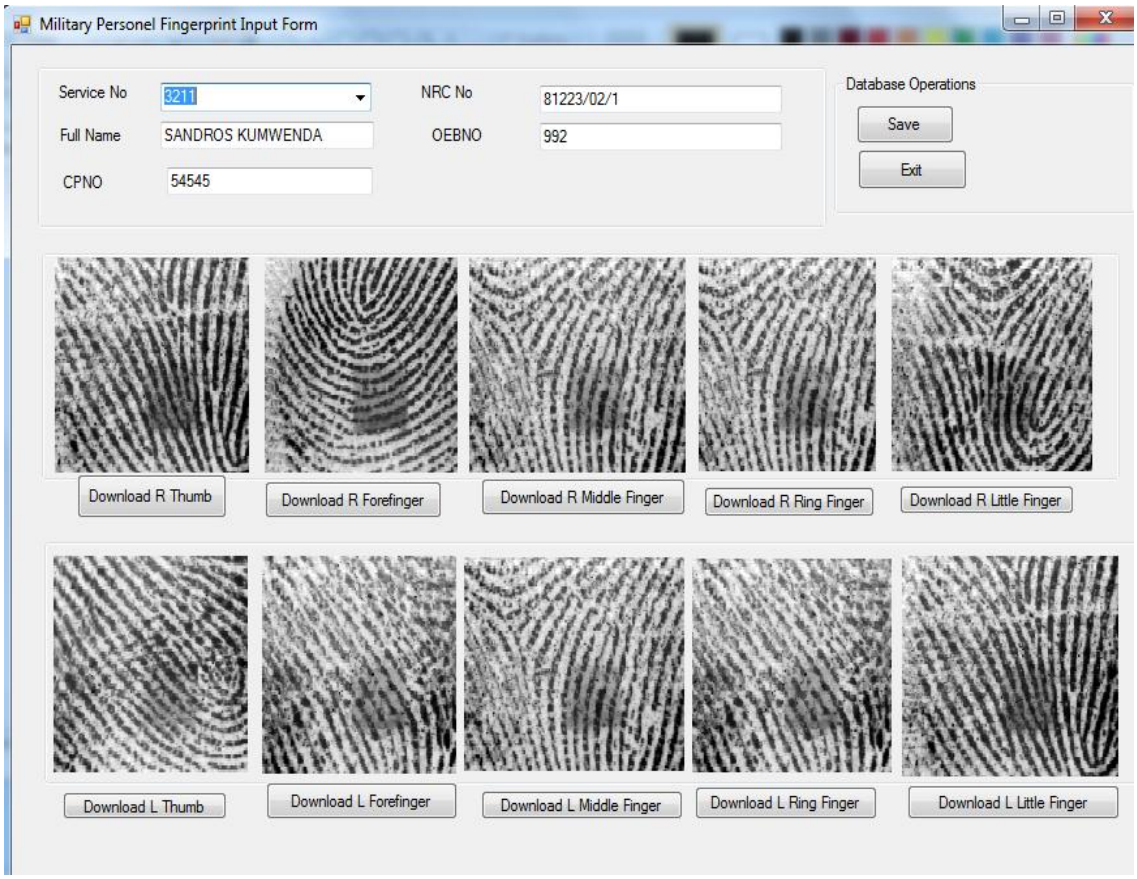


Figure 34: Criminal Vetting automated form

7.5.3 **Master Record MilitaryPersonnel**

Master record military form is used to capture service personnel identification details, birth record and personal description. Additionally, right thumb fingerprint image and passport size photo is also capture as shown in Figure 35.

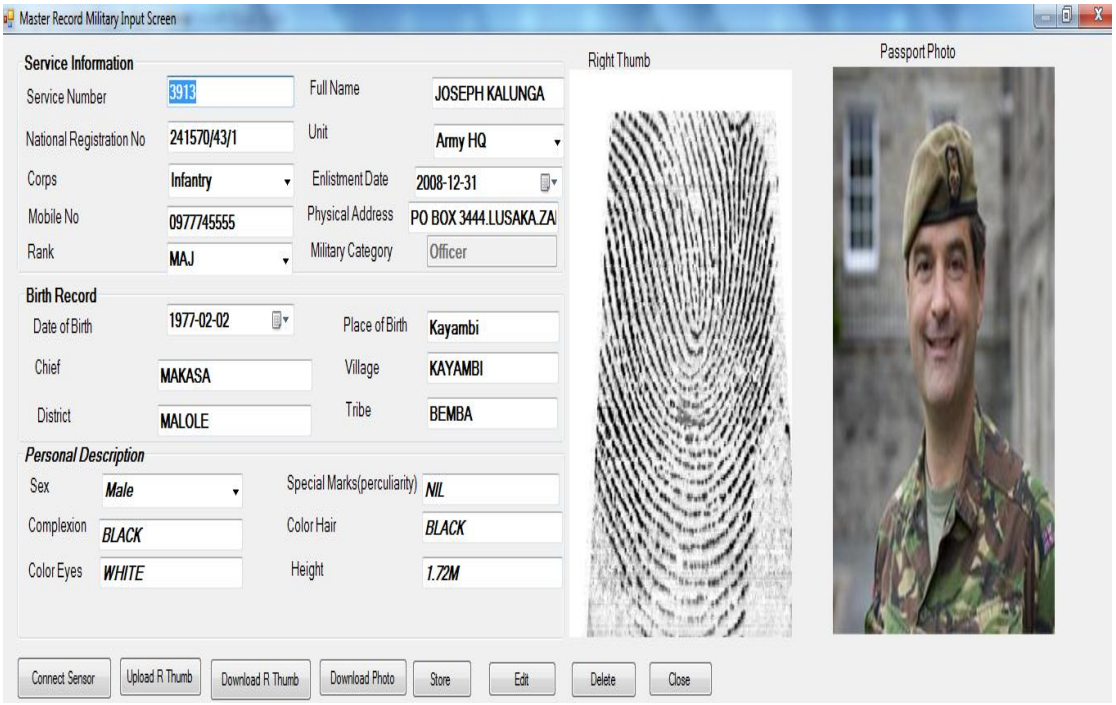


Figure 35: **Master Record Military Personnel Data Input Form**

7.5.4 **Master Record Civilian Personnel**

Master record civilian input form captures the details of civilian employees and stores them in the database. Civilian employees are citizens employed by the government to work in military as non military personnel. They provide supplementary labour to military work force. Civilian personal details are captured for security reasons. This requirement is important and must be implemented to optimise security in the military. Figure 36 shows civilian employees master record input form.

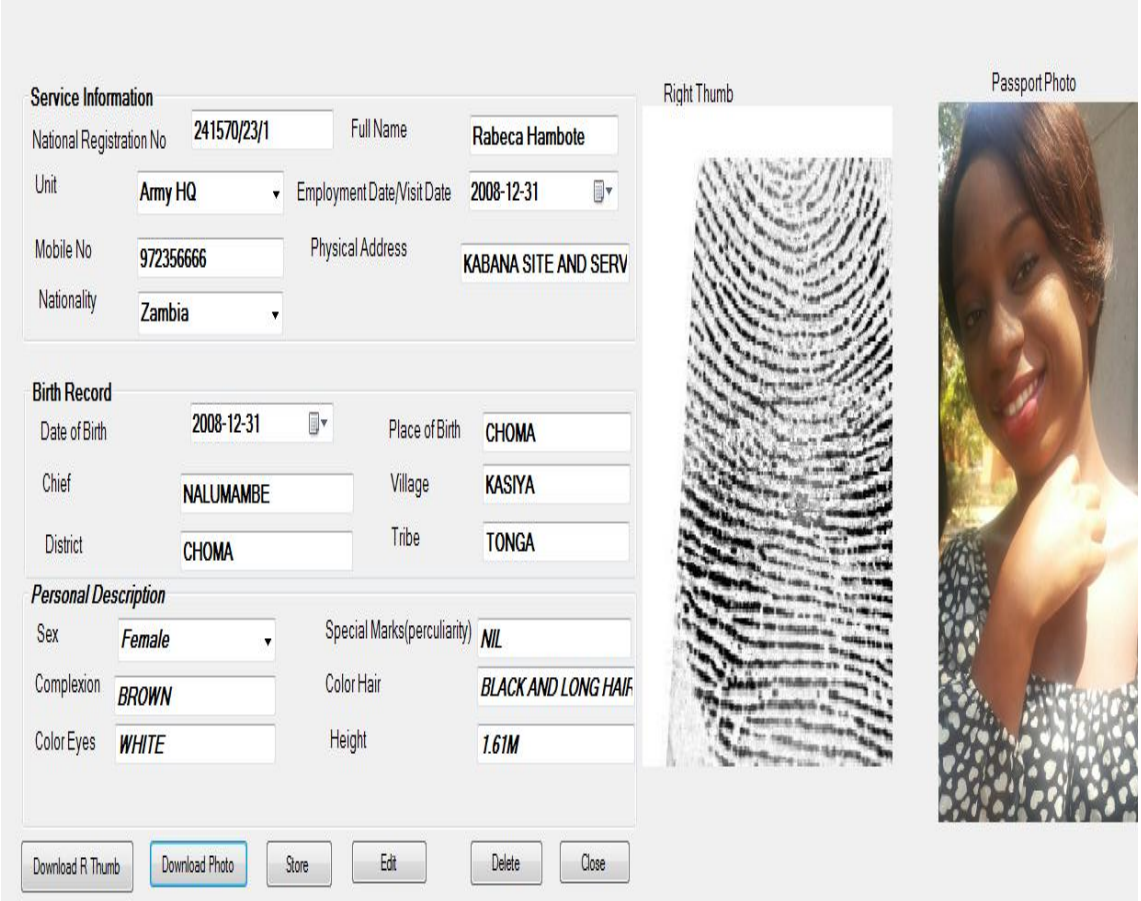


Figure 36: Master Record Civilian Personnel input form

7.5.5 **Military Personnel Case Record Input Form**

Military personnel case record input form is used to capture offences committed by service personnel during the service life time. This information is important for human resources management. Disciplinary record of an individual is very impotant more especially in military organisation where discipline is cardinal. The military cases record input form is shown in Figure 37.

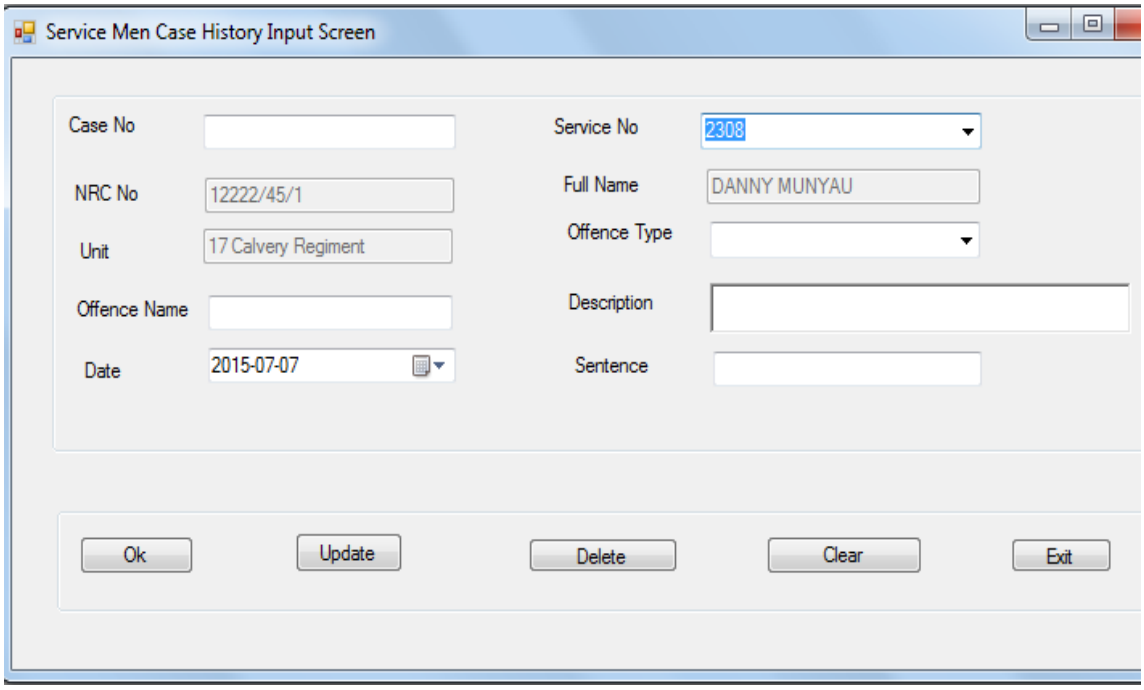


Figure 37: Service men Case History Input form

7.5.6 **Civilian Personnel Case Record Input Form**

Civilian personnel case input screen is used to capture cases committed by civilian employes while enganged to work in the military service and store them in the military police for future references. This form is different from the one used for service personnel because civilians employed in the military are not subjected to military law except when deployed in special cases of national interest. Civilian personnel case input form is shown in Figure 48.

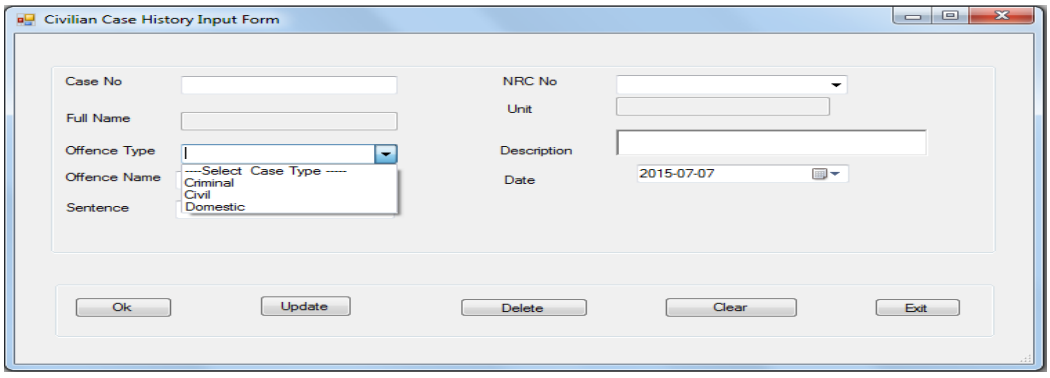


Figure 38: Civilian Personnel Case Record Input form

7.5.7 **Fingerprint Analysis**

Fingerprint analysis form is used for forensic investigation. Analysis form is embodied with intelligent to remove noise, reconstruct fingerprint image features and count the number of pixels in the fingerprint image. This form is also embodied with functionality to adjust contrast in fingerprint image as shown in Figure 39.

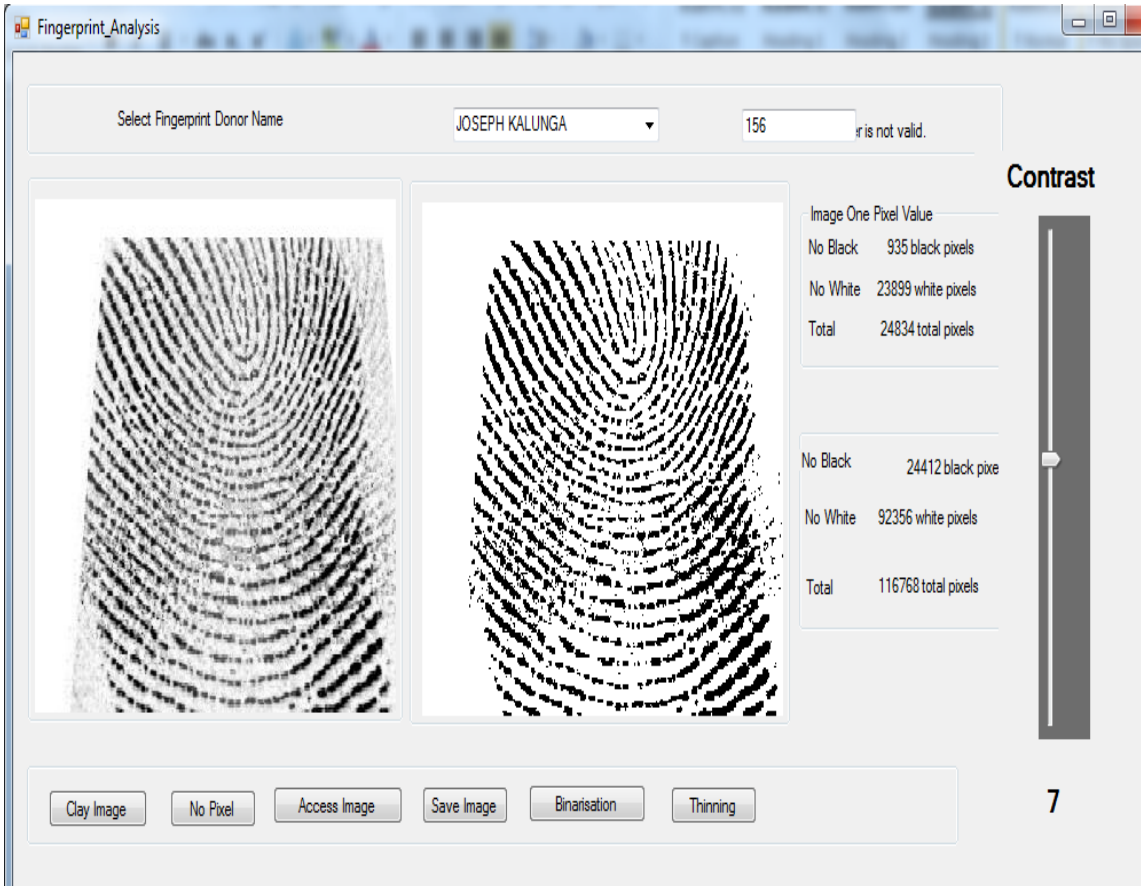


Figure 39: Fingerprint Analysis form

7.5.8 **Service Bio-key Identity Card**

The developed application is embodied with functionality to generate service identity card with improved security features as shown in Figure 40.

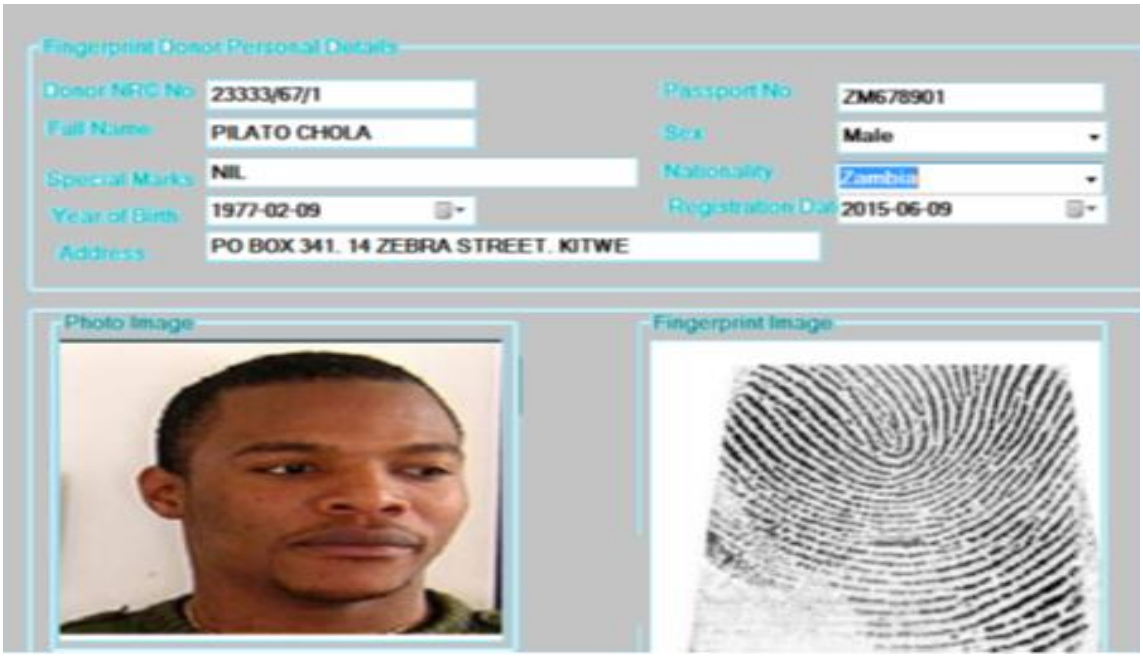


Figure 40: Service Bio Identity Card

7.6 Data Report

Data Reports is another special result which involves creation of dynamic data views. Reports are dynamic because they are dependant on specified parameters to fielter information of interest.

7.6.1 Military Personnel Masters Record Report

This report shows the comprehensive list of all service personnel employed in the army. Military personnel master record report is as shown in Figure 41.

Military Personel Master Record

Name	Sex	Rank	NRC No	Service No	Address	Corps
DANNY MUNYAU	MALE	COL	12222/45/1	2308	PO BOX 23. NDOLA	ARMOUR
SANDROS KUMWENDA	MALE	LT COL	81223/02/1	3211	21 ARAKAN BARRACKS	ARMOUR
CHINYAMA T	MALE	MAJ	233333/89/2	3912	LUSAKA	MEDICAL
JOSEPH KALUNGA	MALE	MAJ	241570/43/1	3913	PO BOX 3444 LUSAKA ZAMBIA	INFANTRY
EMMANUEL HAMUDUDU	FEMALE	CAPT	233334/12/1	4036	ARAKAN BARRACKS. FLAT NO 2	SIGNALS
KENNY MWENE	MALE	CAPT	711231/02/1	4356	ROAN RD 23.KABULONGA	MEDICAL
MARY BWALYA	MALE	SSGT	455555/98/1	4521	23 DABWA STREET	ARTILLERY
NGOZI PHIRI	MALE	SSGT	234512/23/1	4569	PO BOX LUSAKA ZAMBIA	ARMOUR
JOSEPH PHIRI	MALE	COL	344444/09/1	78901	PO BOX 5 KAPIRIMPOSHI	TRANSPORT AND LOGIST

Figure 41: Military Personnel Master Report

7.6.2 **Rank Report**

Rank report shows military ranks defined in the new system. The extract report view of ranks report is given in Figure 42.

A001	GENERAL	Officer
A002	LT GEN	Officer
A003	MAJ GEN	Officer
A004	BRIG GEN	Officer
A005	COL	Officer
A006	LT COL	Officer
A007	MAJ	Officer
A008	CAPT	Officer
A009	LT	Officer
A010	2LT	Officer
A011	O/CDT	Officer
A012	WO1	Soldier
A013	WO2	Soldier
A014	SSGT	Soldier
A015	SGT	Soldier
A016	CPL	Soldier

Figure 42: Rank Report

7.6.3 **Criminal Vetting General Report**

Criminal vetting general Report shows the list of people vetted for criminality involvements. The extracted view of this report contains false record for Mr Sandross Kumwenda is shown in Figure in Figure 43.

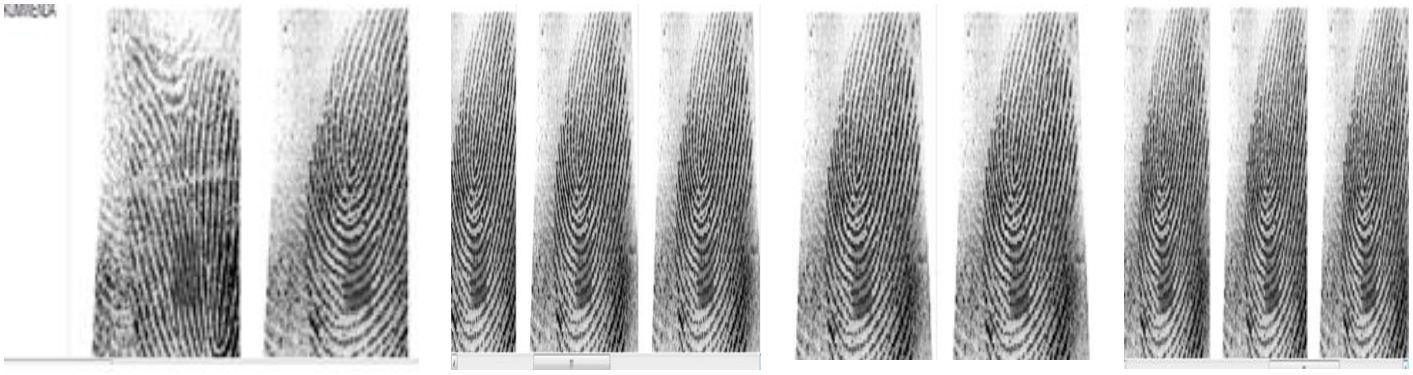


Figure 43: Criminal Vetting Fingerprint Report

7.6.4 **Service Personel Man Number Master**

The service Personel man number master report displays the comprehensive list of all service personal with their passport size photo. This report is important for planning

and may be used to trace those people masquerade personal identity. Figure 44 shows service Number Register. For other service register report formats see appendix E of this report.

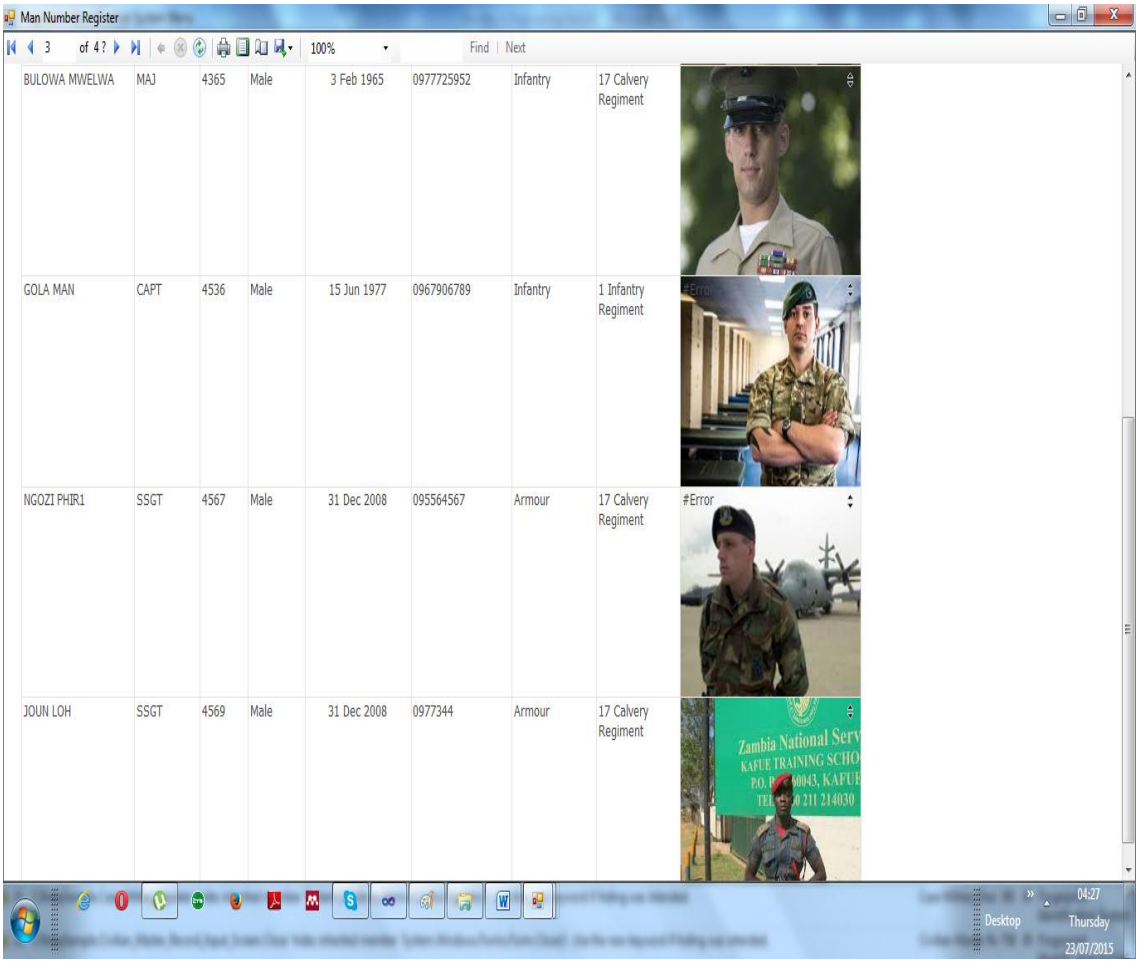


Figure 44: Service Personel Man Number Master Record

7.6.5 **Master Record Enquiry Report**

Master record enquiry report contains pictorial description of a service personnel, particulars, personal description, and birth record and employment details. This report displays only one record which is specified by the service number as illustrated in Figure 45.

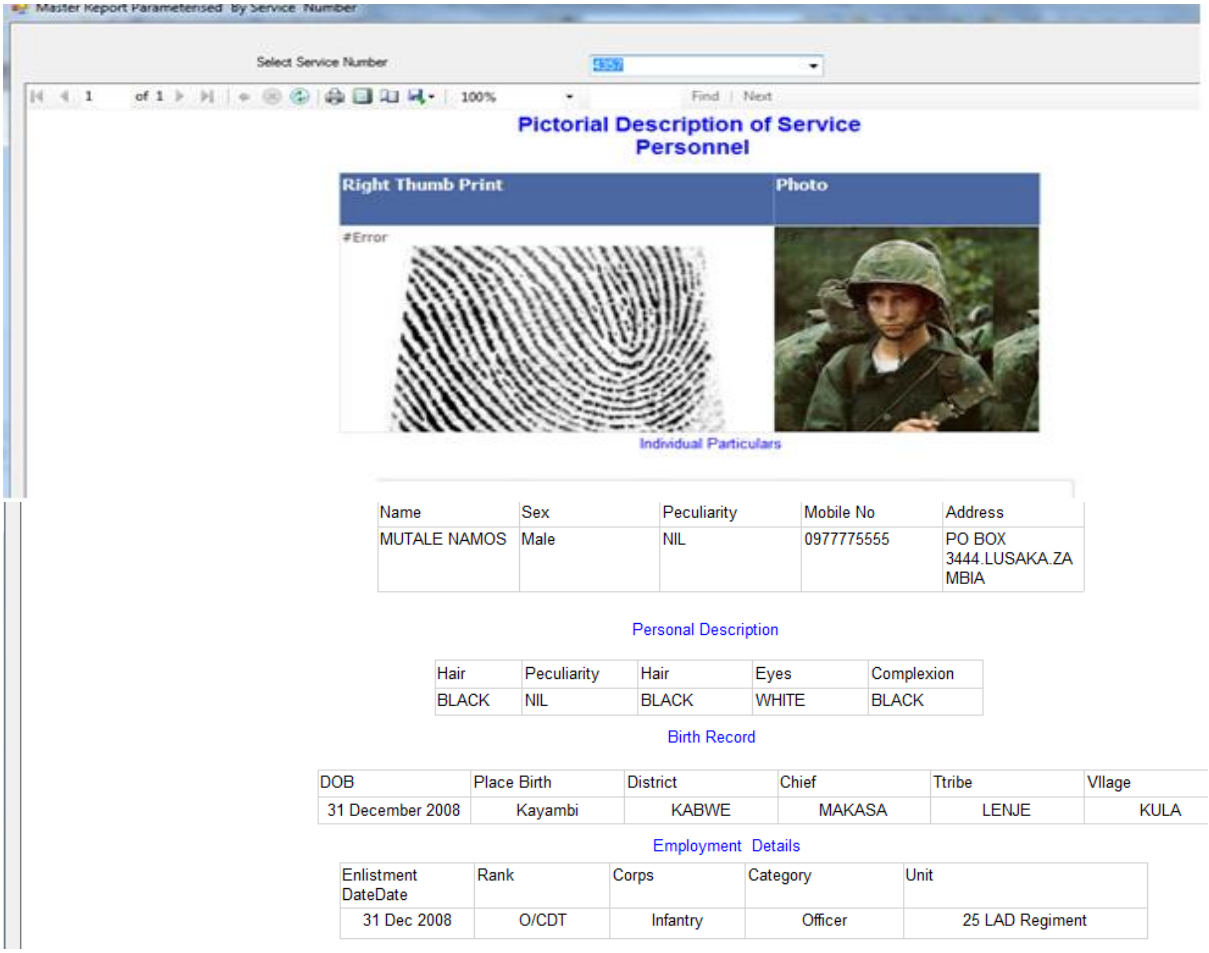


Figure 45: Service Personel Master Record Inquiry Report

7.6.6 **Disciplinary Record Report Sorted by Unit**

Disciplinary Record report displays various offences committed by serving personnel. This report is important because it keeps the record of offences committed while some one is serving in military service. Disciplinary record report may be useful for human resource management. Disciplinary service record report is sorted in unit is shown in Figure 46 and for other format of this report refers to Appendix E.

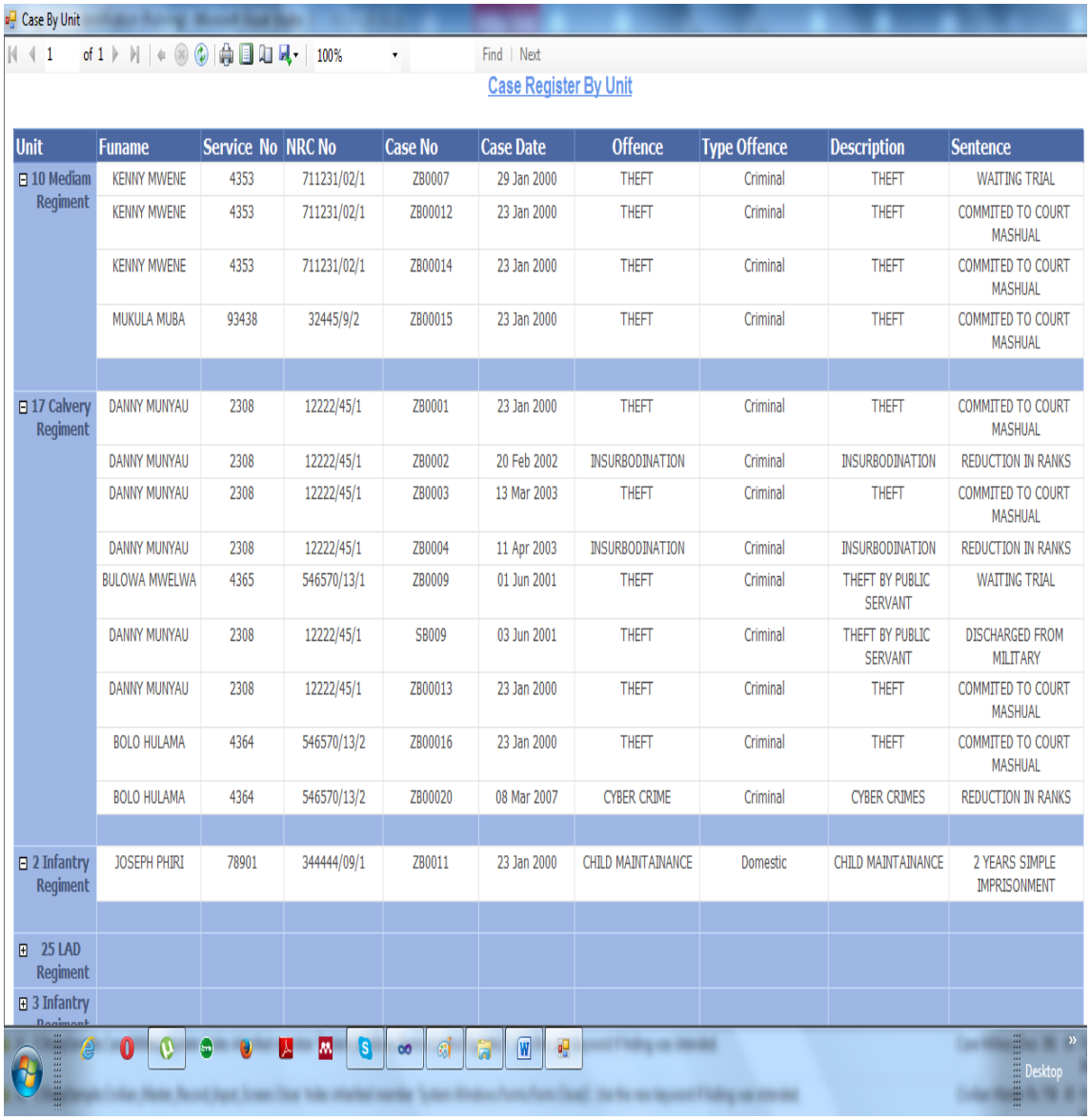


Figure 46: Disciplinary Record Report Sorted by Unit

7.7 Summary

This chapter displayed some of the results generated by the Developed AFBSMO. The result indicated that the developed application is capable identify enrolled personel using the fingerprint. Additionally, the system is embodied with intelligence to store record on the database and subsequently retrieve the same store records and disseminate information on demand (reporting). The fact that the developed application is also able store passport size photos resulted in the system generating fingerprints biometric ID. The generated biometrics ID has improved security features

as compared to the manual service Identity Card. In the next chapter we discuss and evaluate the performance of the developed application.

CHAPTER 8. **EVALUATION**

8.1 **INTRODUCTION**

Evaluation chapter assesses the necessity, quality and matters addressed by the developed product. This is aimed at evaluating the performance of the developed system and collecting information helpful in guiding success implementation. System attributes evaluated were product necessity, technological costing, quality and how this product contributed to fingerprint biometrics knowledge gaps.

8.2 **Product Neccessity**

The developed product is very necessary to the Zambia Army considering the following reasons:

i. The current traditional approach of human identity verification has lot of security flows as reported in the literature review and therefore, biometric identifier is required to optimise securityin military organisation.
ii. The developed application is embodied with functionalities to adequately address problems associated with current military police manual procedures.
iii. The approach to develop an application in house for military organisation is the best option considering security concern of malicious codes (back door attack) that may be associated with vendor software.
iv. As learn from literature review, technologies migrate from the military to civil community and not the other way round. The literature also has not shown us which Armies has donated or sold software of that nature to another Army. For example, Military Equipments such tanks and other wepons may be procured from one army to another but the technology behind remains with the vendor. Biometric traits that US Army used in Iraq were taken with when the operation was completed.

8.3 **Technology Costing**

In interms of development tool (technology) evaluation, AFBSMO was developed from the open source technologies. Open source software such as MySQL is a free software and community based. Microsoft visual studio 2010 express edition with CSharp programming language is free sotware too and the fingerprint sensor costed

less than 100 dollar. Considering the developed functionality or results and the cost of technologies employed, it is suffice to say that we made the precise choice of technology in this study.

8.4 **Quality**

The quality of the developed product was measured based on accuracy, reliability, maintainability, security, reusability, integration, reporting and scalability.

8.4.1 **Accuracy and Reliability**

Accuracy of the sofware product measures the extent to which a program fulfils its specification. The developed application has archieved all the fuctional requirements stipulated in analysis chapter of this study. For example to evaluate the performance of developed system FAR and FRR were calculated. In the first batch of 10 fingerprint enrolled, the total of 10 print were identified. In the next batch of 10 no fingerprint was rejected. In the third batch no rejection was record again. Therefore, biometric error allowance is provided for at the rate 0.001% FAR and 0.001% FRR. This provision is learnt from the literature reviews stated that any biometric system is associate with errors of false acceptance and false rejection due to various factors stipulated in same chapter. The accuracy level of biometric sub system is good. From preliminary results, the developed database for military police has also shown good results in terms of record management. System uptime is pited at 99.9% reliable because it does not depend on any internet connectivity. Reliability is an extent to which a program can be maintained so that it can fulfil its specific function.

8.4.2 **Maintainability, Security and Usability**

Product Maintenance involves tracing and correction of errors. The developed system maintainability is rated at 60% because of the three-tier architecture implemented. It is very difficult for unexperienced programmer to trace the actual source of error in three-tier architecture. Security is implemented in traditional way using user name and password and is rated at 50%. The security of this application is planned to be enforced by third hand security tool such as firewalls. Re-usability addresses the concept of writing code so that it can be used more than once. Code in this application is highly reusable and is rated at 70% because in most cases the structure and the syntax of the code remains the same but what changes frequently are variable names.

8.4.3 **Integration and Reporting**

Fingerprint Biometric System has been integrated effectively in military database as shown in the results. The integration rate may evaluate to an estimation of 99.99%. The developed application reporting capabilitility is awesome as shown in the results chapter of this study.

8.4.4 **Scalability**

Scalability is the capability of a system to handle a growing amount of work in a proficient manner and abilities to accommodate expanded functionality in future. Users defined in the developed AFBSMO are divided into three categories of users namely sentries, Criminal investigation officer and system administrator. Scalability implies that system administrator can define limitless numbers of users in this system and assigned them with any of the three categories. Each category has different user prevailages.

8.5 **Addressing Knowledge Gaps**

Knowledge gap is highlighted in literature of this study. This section appraise the developed product against biometric standard, privacy issues, legal framework and disability issues reported in literature review as lacking coverage in existing biometric application. The study has evaluated these concerns as follows:

8.5.1 **Standards**

The issue of biometric standard is still unresolved. The research intended to use FBI predefined stardards. For example fingerprint image Resolution prescribed by FBI should be (R_N) ≥500 dpi but instead we archieve (RN) ≥300 dpi. Other pararameters too are different. This differences is due to the low quality of fingerprint sensor (digitalpesona) used in this study. For details on biometric standards check appendix section D.

8.5.2 **Privacy Issues and Legal Framework**

Privacy and legal framework issues sorounding biometrics system and collection of sensitive data have been resolved in this study. The product is supported legally because in Zambia Nation security prevails over any other law or concern. The

developed application aim is to improve national security through integrating biometrics system in military police database.

8.5.3 **Provision for the Disabled**

In literature review chapter, we learnt that most biometric system has not provided modalities of identifying disabled people. In our study, these people are provided for by means of employing their NRCs which has personal details and the system allow capturing of passport size photos. The captured photo and other personal details are stored in the database for future references.

8.5.4 **Summary**

This chapter evaluated the developed system based on product necessity, technology costing and quality. From the evaluation process the developed system is rated above average.

CHAPTER 9. **CONCLUSION**

9.1 **Introduction**

The overall aim of this research was to improve security in the barracks through development and deployment of security application that integrate fingerprint

biometrics system in to military police database. The specific objectives of this research were to investigate the system development process of an Automatic Fingerprint Biometrics system and fuse it in the military police database.

9.2 **Study Process**

Prior to system development, preliminarly investigation was conducted to ascertain the feasibility of integration process. From the investigation conducted, it was learnt that Military police were facing operational problems include, crisis to verify the true identity of visitors, huge operational cost, record storage/retrieval problems, technological obselence, duplication of works and labour intensive due to large volume of documents being handled. The military police operational procedures and documentation were learnt using observation, record inspection (check AFG 12A form in appendix B) and unstructured interviews. The integration of fingerprint biometrics system in military database was the only solution to improve security and operation problems of military police in the barracks.

Literature taught us that, architectural design of fingerprint biometrics system has three components namely fingerprint sensor, business logic and relational database. Literature also taught us that fingerprints captured for biometric identification are processed further into normalisation, segmentation, binalisation, thinning and finally skeleton image (template). On the other hand fingerprint capture for employment criminal vetting and criminal investigation may be subjected to enhancement and noise removal during the process of analysis. Furthermore, fingerprint biometric system has four important algorithms namely feature extraction, enrollment, matching and identification (verification). These and the other requirements collected during preliminary investigation were modelled in to rich piture, database schema diagram, ERD and class diagram as shown in analysis stage. The ultimate result of objective one was an integrated system design.

9.3 **Developed Product**

After application design, the design docucument were translated into the actual computer application. The application developed was a desktop application and is named automated fingerprint biometric system for military organisation. It is developed using MySQL database on Dot Net framework 4.0 and C# object oriented

programming language. The developed computer application for military police has the following features:

i. Ability to store and retrieve information,
ii. Automated Fingerprint acquisition for criminal investigation,
iii. Ability to identify human being using the fingerprint,
iv. Has automated employee discplinary record,
v. Generate service identity card which include bio-key feature and
vi. Reporting and data inquiry capabilities.

9.4 Challenges

The following were difficulties experienced during this study:

i. Some system development kit (SDK) classes could not work on visual studio 2010 developer kit.
ii. Microsoft foundational classes found in visual studio 2010 lacked specialised image processing classes.
iii. Thinning process could not work on Dot Net framework.
iv. The biometric system is limited to enrol and verify ten fingerprints at time. This is due to usage of domestic SDK.
v. The quality of fingerprint image produced by digital personal scanner contains some black spots (noise). This could increase error rates in mutuae based fingerprint biometrics system.
vi. The research lack digital camera for capturing identity card passport sizes photos because of the cost of camera which was unaffordable. Hence, experimental pictures captured into the system were downloaded from the internet.

9.5 Recommendation and Future Work

In future Biometrics system is expected to play a major role not only in military but also other industries, such as medicine, science, robotics, engineering, manufacturing and all areas of vertical enterprise businesses. Biometrics offers excellent value to various industries, but also challenges to individual privacy. Besides that, the cost of biometric devices is also expected to go down and therefore the number of biometrics

devices would explode on the market. In amide of these projections, the following are recommended:

1) There is need to conduct another research aiming at developing biometric algorithm which is more scalable than the one used.
2) Military to collaborate with the Universities in researches such as this one which involves security of information as modern warfare is centred information and telecommunication System (ICT).
3) Military should be encouraged to develop their own systems because of the risk associated with off-shelf software. It is believed that off the shelf software may be planted with a malicious code or software that intends to steal information or distract computer systems at the later date.

However, fingerprint biometrics still suffer from problems such as huge implementation cost, high error rate (false acceptance and false rejection rate), slow to process and lack of predefined standards.

REFERENCES

Jia. A, Cai, L., Lu, P., & Liu, X. (2007). Fingerprint matching based on weighting method and the SVM, *70*, 849–858.

Clodfelter, R (2010). Journal of Retailing and Consumer Services Biometric technology in retailing : Will consumers accept fingerprint authentication ? *Journal of Retailing and Consumer Services*, *17*(3), 181–188.

Abhyankar, A., & Schuckers, S. (2009). Integrating a wavelet based perspiration liveness check with fingerprint recognition, *42*, 452–464.

Adams, J. (2000). Biometrics and smart cards. *Biometric Technology Today*, *8*(7), 8–11.

Apacs, H. (2006). The Chip and PIN success story. *Card Technology Today*, *18*(7–8), 10–11.

Awad, A. I., & Baba, K. (2012). Singular Point Detection for Efficient Fingerprint Classification Graduate School of Information Science and Electrical Engineering Kyushu University Library, *2*(1), 1–7.

Bazen, A. M. (2002). Fingerprint Identification - Feature Extraction , Matching , and Database Search.

Bhattacharya, S., & Mali, K. (2013). A Bar Code Design and Encoding for Fingerprints. *Procedia Technology*, *10*, 672–679.

Bolle, R. M., Senior, A. W., Ratha, N. K., & Pankanti, S. (2006). Fingerprint Minutiae : A Constructive Definition. *Lecture Notes in Computer Science*, *2359*, 58–66.

Boswell, T. (2009). Smart card security evaluation: Community solutions to intractable problems. *Information Security Technical Report*, *14*(2), 57–69.

Bracha, H. S. (2006). Human brain evolution and the "Neuroevolutionary Time-depth Principle:" Implications for the Reclassification of fear-circuitry-related traits in DSM-V and for studying resilience to warzone-related posttraumatic stress disorder. *Progress in Neuro-Psychopharmacology & Biological Psychiatry*, *30*(5), 827–53.

Breebaart, J., Buhan, I., De Groot, K., & Kelkboom, E. (2011). Evaluation of a template protection approach to integrate fingerprint biometrics in a PIN-based payment infrastructure. *Electronic Commerce Research and Applications*, *10*(6), 605–614.

Canadian police get real-time fingerprint and palm ID system. (2011). *Biometric Technology Today*, *2011*(6), 12.

Cao, K., Pang, L., Liang, J., & Tian, J. (2013). Fingerprint classification by a hierarchical classifier. *Pattern Recognition*, *46*(12), 3186–3197.

Cappelli, R., & Ferrara, M. (2012). A fingerprint retrieval system based on level-1 and level-2 features. *Expert Systems with Applications*, *39*(12), 10465–10478.

Cappelli, R., Ferrara, M., & Maltoni, D. (2015). Large-scale fingerprint identification on GPU. *Information Sciences*, *306*, 1–20.

Chaudhari, A. S., Patnaik, G. K., & Patil, S. S. (2014). Implementation of Minutiae Based Fingerprint Identification System Using Crossing Number Concept. *Informatica Economica*, *18*(1/2014), 17–26.

Danese, G., Giachero, M., Leporati, F., & Nazzicari, N. (2011). An embedded multi-core biometric identification system. *Microprocessors and Microsystems*, *35*(5), 510–521.

De Marsico, M., Nappi, M., Riccio, D., & Tortora, G. (2009). A multiexpert collaborative biometric system for people identification. *Journal of Visual Languages & Computing*, *20*(2), 91–100.

Egli, N. M., Champod, C., & Margot, P. (2007). Evidence evaluation in fingerprint comparison and automated fingerprint identification systems — Modelling within finger variability, *167*, 189–195.

Fons, M., Fons, F., & Cantó, E. (2012). Biometrics-based consumer applications driven by reconfigurable hardware architectures. *Future Generation Computer Systems*, *28*(1), 268–286.

Gold, S. (2010). Military biometrics on the frontline. *Biometric Technology Today*, *2010*(10), 7–9.

Gold, S. (2014). Biometrics at the border. *Biometric Technology Today*, *2014*(10), 5–9.

Holder, E. H., & Robinson, L. O. (2010). U.S. Department of Justice Office of Justice Programs. *Juvenile Justice*, 1–63.

Horan, P. (2000). Using Rich Pictures in Information Systems Teaching. *1st International Conference on Systems Thinking in Mangement, 2000*, *1*, 257–262.

Jain, A. K., Hong, L., & Kulkarni, Y. (1999). A Multimodal Biometric System Using Fingerprint, Face, and Speech. *International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA)*.

Jain, A. K., Prabhakar, S., & Pankanti, S. (2002). On the similarity of identical twin ÿngerprints, *35*, 2653–2663.

Jain, A., Nandakumar, K., & Ross, A. (2005). Score normalization in multimodal biometric systems. *Pattern Recognition*, *38*(12), 2270–2285.

Jimoh, R. G., Olaniyi, A. S., & Adewole, K. S. (2011). Adoption of Fingerprinting as an Automated Access Control Technique in University Hostels. *ARPN Journal of Systems and Software*, *1*(4), 149–153.

Khalil, M. S., Mohamad, D., Khan, M. K., & Al-Nuzaili, Q. (2010). Fingerprint verification using statistical descriptors. *Digital Signal Processing*, *20*(4), 1264–1273.

Kumar, A., & Wu, C. (2012). Automated human identification using ear imaging. *Pattern Recognition*, *45*(3), 956–968.

Lin, H., Yifei, W.,& Jain, A. (1998). Fingerprint Image Enhancement: Algorithm and Performance Evaluation. *IEEE Trans. Pattern Anal. Machine Intel*, *20*(8), 777–789.

Langer, A. M. (2008). *Analysis and Design of Information Systems Third Edition British Library Cataloguing in Publication Data*. 3rd ed. New York, Springer. pp 248-257

Law enforcement agencies tap into biometrics. (2012). *Biometric Technology Today*, *2012*(6), 2.

Lewis, I. (2014). Ever had your fingerprints taken? Meeting the challenges of 21st Century access control. *Biometric Technology Today*, *2014*(5), 9–11.

Lin, C. H., Chen, J. L., & Tseng, C. Y. (2011). Optical sensor measurement and biometric-based fractal pattern classifier for fingerprint recognition. *Expert Systems with Applications*, *38*(5), 5081–5089.

Liu, F., Zhao, Q., & Zhang, D. (2011). A novel hierarchical fingerprint matching approach. *Pattern Recognition*, *44*(8), 1604–1613.

Liu, Y. (2011). Scenario study of biometric systems at borders. *Computer Law & Security Review*, *27*(1), 36–44.

Lumini, A., Maio, D., & Maltoni, D. (1997). Continuous versus exclusive classification for fingerprint retrieval. *Pattern Recognition Letters*, *18*(10), 1027–1034.

Maltoni, D. (2005). A Tutorial on Fingerprint Recognition. *Advanced Studies in Biometrics*, *3161/2005*, 121–138.

Mansfield-Devine, S. (2012). Biometrics at war: The US military's need for identification and authentication. *Biometric Technology Today*, *2012*(5), 5–8.

Marsico, M. De, Nappi, M., Riccio, D., & Tortora, G. (2009). Journal of Visual Languages and Computing A multiexpert collaborative biometric system for people identification, *20*, 91–100.

Martinez-Diaz, M., Fierrez, J., Galbally, J., & Ortega-Garcia, J. (2011). An evaluation of indirect attacks and countermeasures in fingerprint verification systems. *Pattern Recognition Letters*, *32*(12), 1643–1651.

Mayes, K. E., Markantonakis, K., & Hancke, G. (2009). Transport ticketing security and fraud controls. *Information Security Technical Report*, *14*(2), 87–95.

Montoya Zegarra, J. a., Leite, N. J., & da Silva Torres, R. (2009). Wavelet-based fingerprint image retrieval. *Journal of Computational and Applied Mathematics*, *227*(2), 294–307.

Nagaty, K. A. (2005). An adaptive hybrid energy-based fingerprint matching technique. *Image and Vision Computing*, *23*(5), 491–500.

Nguyen, N. (2015). Chokepoint: Regulating US student mobility through biometrics. *Political Geography*, *46*, 1–10.

Nigam, I., Vatsa, M., & Singh, R. (2015). Ocular biometrics: A survey of modalities and fusion approaches. *Information Fusion*, *26*, 1–35.

Nunn, S. (2001). Police technology in cities: Changes and challenges. *Technology in Society*, *23*(1), 11–27.

O'Gorman, L. (1999). An overview of fingerprint verification technologies. *Information Security Technical Report*, *4*(1), 28–29.

Rajanna, U., Erol, A., & Bebis, G. (2010). A comparative study on feature extraction for fingerprint classification and performance improvements using rank-level fusion. *Pattern Analysis and Applications*, *13*(3), 263–272.

ZKTeco Inc.(2012). ZKFinger SDK development guild . Version 3.0.

Vatsa, M., Singh, R., Noore, A., & Morris, K. (2011). Simultaneous latent fingerprint recognition, *11*, 4260–4266.

William,M.,(2012). Using ESF Database Migration Toolkit to migrate MySQL, Postgres, Oracle, SQL Server to DB2. *IBM developer works*, [online]. Available at: *http://www.ibm.com/developerworks/data/library/techarticle/dm-0606khatri/[accessed 26.10.2016]*

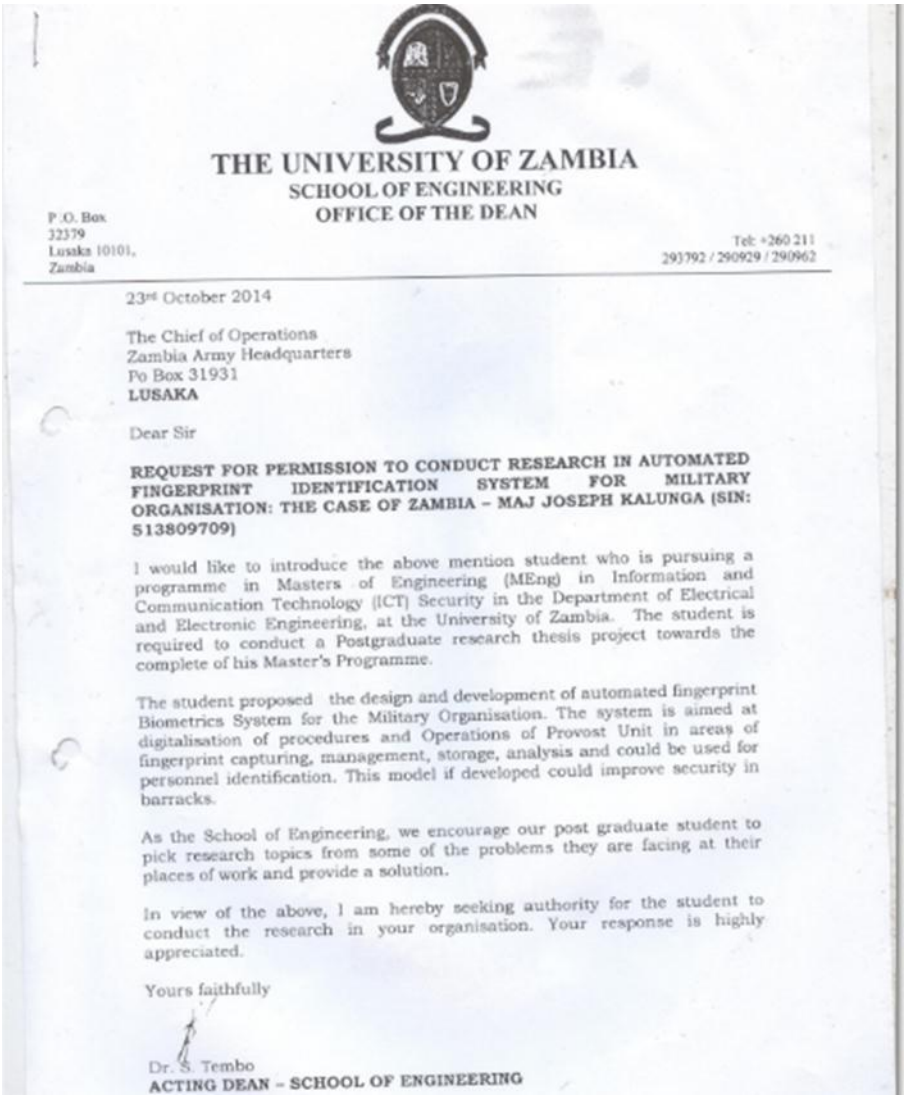
Zhu, L., & Zhang, S. (2010). Multimodal biometric identification system based on finger geometry, knuckle print and palm print. *Pattern Recognition Letters*, *31*(12), 1641–1649.

APPENDICES

APPENDIX A: SECURITY CLEARANCE

The appendix section A contains documents for purposes of security clearance so as to conduct the study in the Zambia Army. These documents include the Univiversity of Zambia Request letter and Zambia Army reply Loose Minute as shown below:

A.1 University Research Student Introductory Letter



THE UNIVERSITY OF ZAMBIA
SCHOOL OF ENGINEERING
OFFICE OF THE DEAN

P.O. Box
32379
Lusaka 10101,
Zambia

Tel: +260 211
293792 / 290929 / 290962

23rd October 2014

The Chief of Operations
Zambia Army Headquarters
Po Box 31931
LUSAKA

Dear Sir

REQUEST FOR PERMISSION TO CONDUCT RESEARCH IN AUTOMATED FINGERPRINT IDENTIFICATION SYSTEM FOR MILITARY ORGANISATION: THE CASE OF ZAMBIA – MAJ JOSEPH KALUNGA (SIN: 513809709)

I would like to introduce the above mention student who is pursuing a programme in Masters of Engineering (MEng) in Information and Communication Technology (ICT) Security in the Department of Electrical and Electronic Engineering, at the University of Zambia. The student is required to conduct a Postgraduate research thesis project towards the complete of his Master's Programme.

The student proposed the design and development of automated fingerprint Biometrics System for the Military Organisation. The system is aimed at digitalisation of procedures and Operations of Provost Unit in areas of fingerprint capturing, management, storage, analysis and could be used for personnel identification. This model if developed could improve security in barracks.

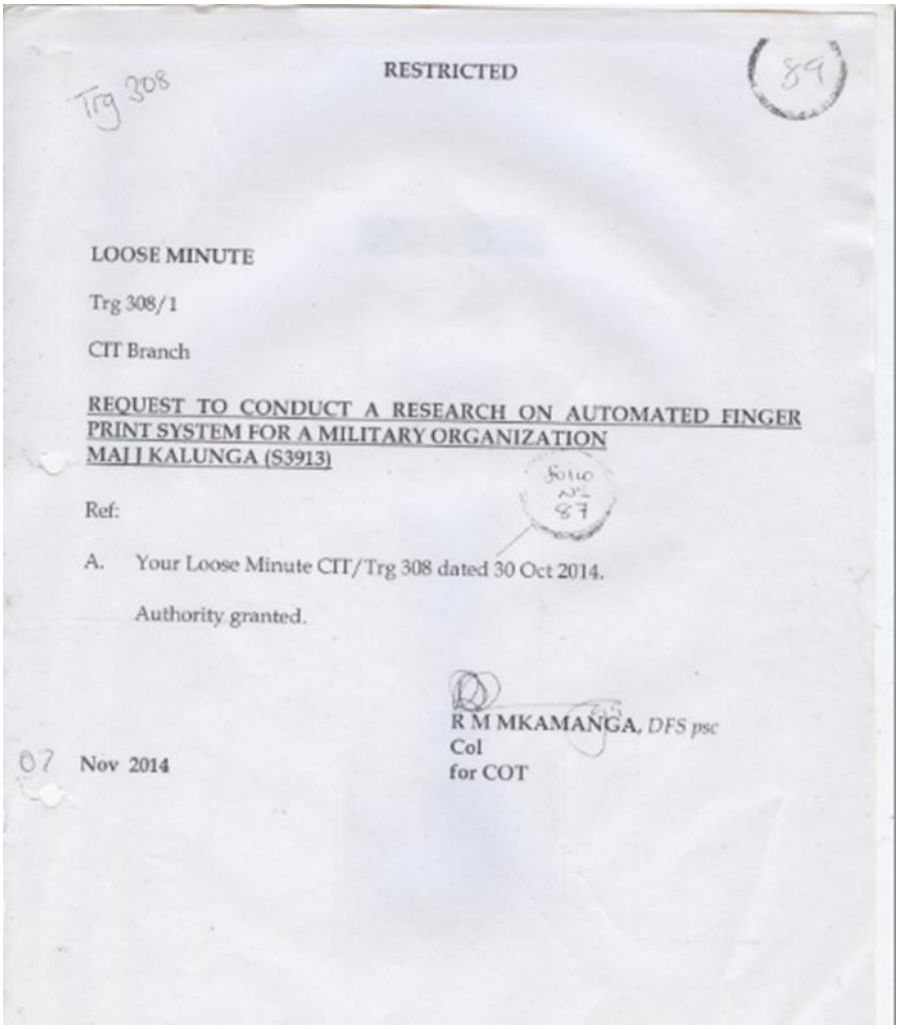
As the School of Engineering, we encourage our post graduate student to pick research topics from some of the problems they are facing at their places of work and provide a solution.

In view of the above, I am hereby seeking authority for the student to conduct the research in your organisation. Your response is highly appreciated.

Yours faithfully

Dr. S. Tembo
ACTING DEAN – SCHOOL OF ENGINEERING

A.2 Zambia Army Research Authorization Loose Minute



RESTRICTED

Trg 308

89

LOOSE MINUTE

Trg 308/1

CIT Branch

REQUEST TO CONDUCT A RESEARCH ON AUTOMATED FINGER PRINT SYSTEM FOR A MILITARY ORGANIZATION
MAJ I KALUNGA (S3913)

Ref:

A. Your Loose Minute CIT/Trg 308 dated 30 Oct 2014.

Authority granted.

R M MKAMANGA, *DFS psc*
Col
for COT

07 Nov 2014

APPENDIX B: PROJECT PLANNING AND COSTING

B.1 Planning

The project started on the 1st January and is expected to be completed after hundred and eleven (111) days less Saturdays and Sunday. The planning has not observed any territorial or international public holidays. Project activities includes: (i) Feasibility Study (ii) **Data collection**. Two data collection methods were employed record inspection and observation. An organisation visited was Military police (provost) Army Unit Special Investigation Branch (SIB) located at Army Headquarters (Arakan Barracks) Lusaka Zambia. The ultimate

result was to understand the management and utilisation of fingerprint for human identification in the Army. For sample military application employment in the public services or non-criminal enquiry form (FZA MP 12A check the appendix section of this report. (iii) **Literature review**. Examine various literatures on algorithms, techniques and methods used in developing a fingerprint biometrics system. (iv) **Analysis**. This activity involves examination of the finding discovered in both data collection and literature review. (vi) **Design**. Design phase include database and application design abstract structures.(vii) application Development and testing . This activity involves the actual program coding and modular testing of the operation of specific computer language code. (viii) D**ocumentation**. Documentation activity involves the actual writing of the thesis and recording the results of an application. For the project schedule and Gantt chart refer to table B.1.1 and Figure B.1.2 respectively.

Table B.1.1 Project Activity Timeline

Activity	Start Date	Number of Days	End Date
Feasibility Study	05-Jan-2015	5	09-Jan-2015
Requirements Elicitation	12-Jan-2015	5	16-Jan-2015
Literature Review	19-Jan-2014	14	03-Feb-2015
Requirement Analysis	04-Feb-2015	12	20-Feb-2015
System Design	23-Feb-2015	5	27-Feb-2015
Development and Coding	02-March-2015	34	22-April-2015
Testing	23-April-2015	5	27-April-2015
Documentation	28-April-2015	21	29-May-2015

B.1.2 Gantt chart

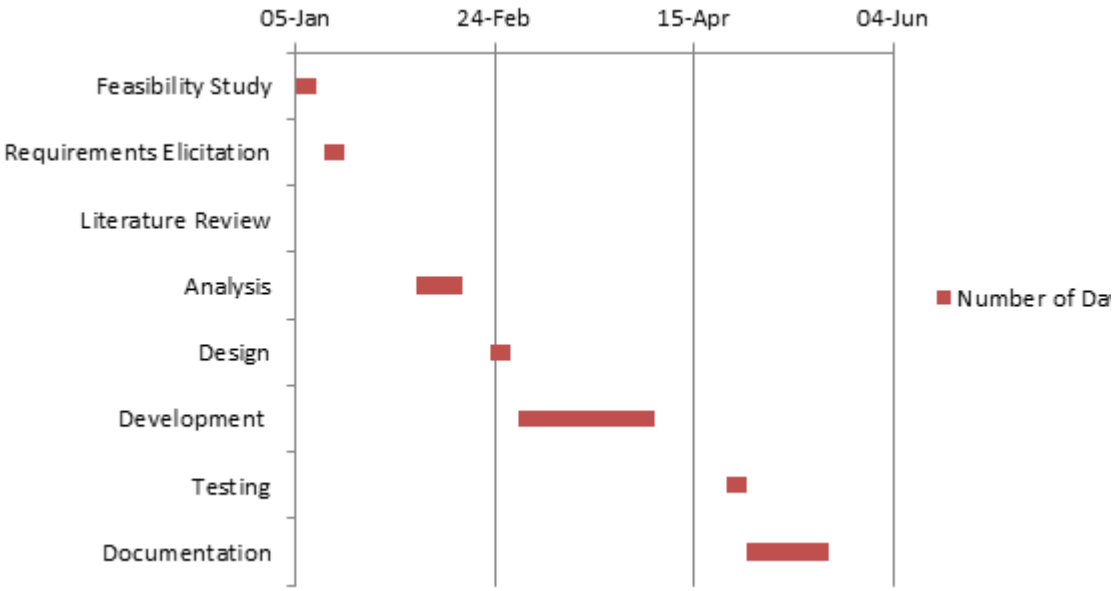


Figure B.1.2 Project Ghantt Chart

B.2 Project Costing

Table B.2 Project Requirements Costing

Nameof the Requirement	Specification	Functional/Relevance	Unit Cost($)	Total Cost($)
Hardware - Laptop	- harddisk 80GB Minimum - processor 3.2 MHz. Minimum. Preferable core i3 and above - graphics frequency 3.30 MHz	Used for development and testing	500.00	500.00
Fingerprint Scanner	- Image resolution 500 pixels per inch. - Image area 9.75mm X 0.41mm/ 192 X 8 pixel - ISO / IEC 7816 T=0 and T=1 - Up to 8Mhz smart cards, and a 412 kbit/s communication speed	Fingerprint capture	80.00	80.00

Digital Camera	- 18.0 Megapixels - 18-55mm lenses - Speed 3frs - Full-high definition	Capturing passport size photos	500.00	500.00
Software - Visual Studio	Version 2005 and up	Software development environment and compiler	259.99	600.00
- Microsoft Project	Version 2003 and up	For project planning	349.00	3490.00
- Microsoft Visual	Version 2003	For project diagrams	259.99	259.99
Labour	N/A	System Development in terms of man hours at standard rate of		
Total Cost				***5429.99***

B.3 Rich Picture

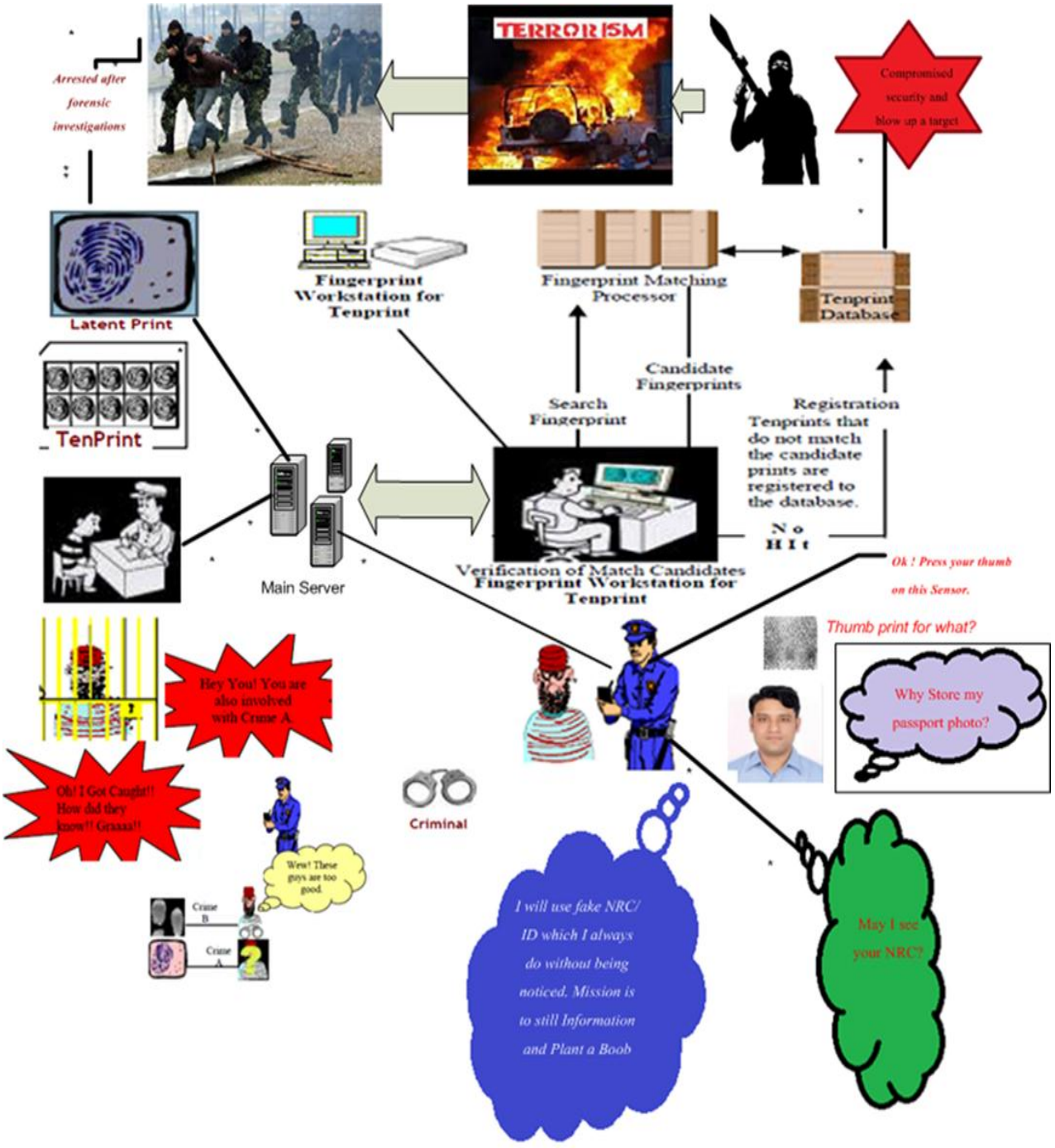
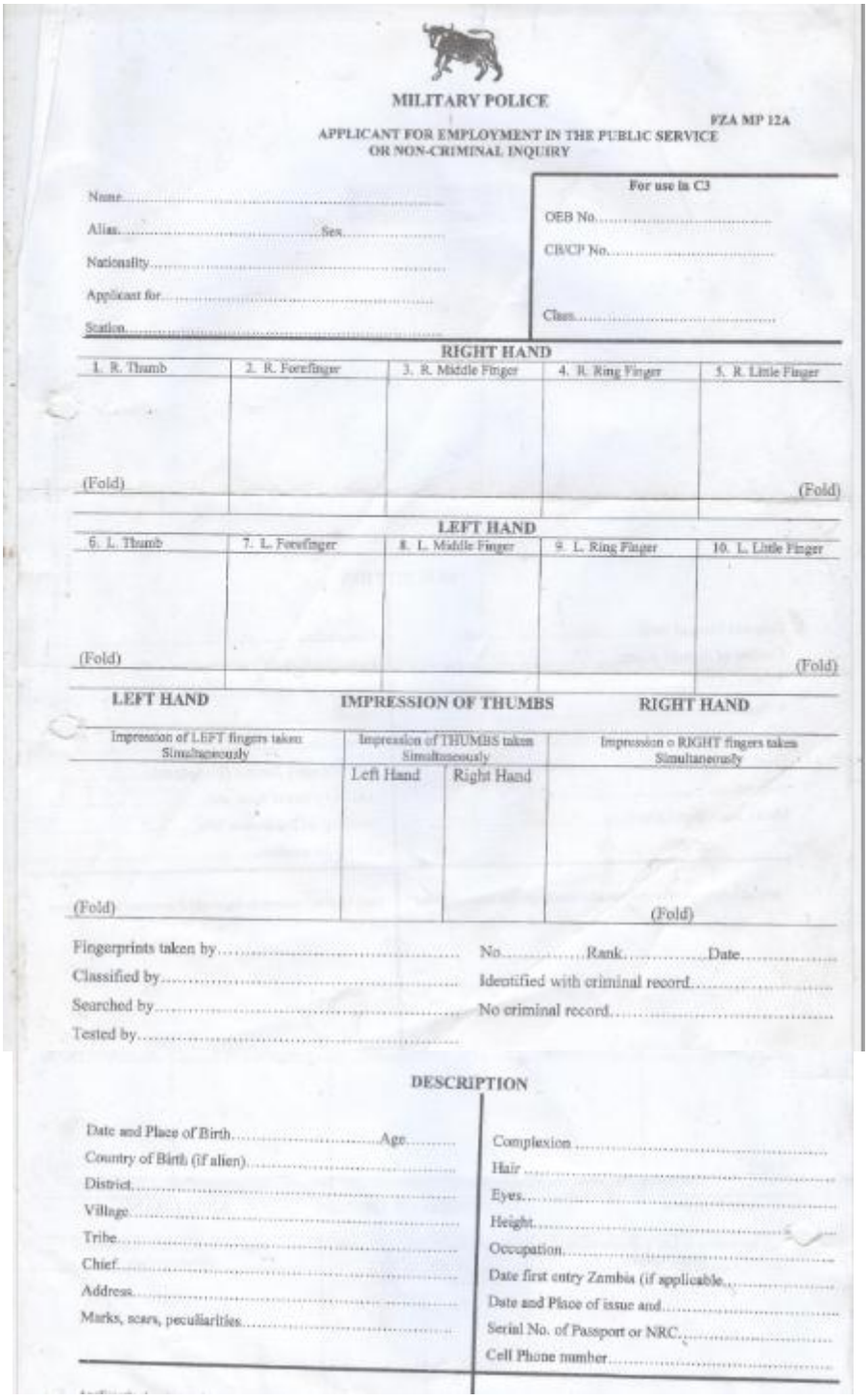


Figure B.3: Rich Picture of the AFBSMO

APPENDIX C: Government Prescribed forms

C.1 Applicantion for Employment in Public Service or non-Criminal Inquiry blank(FZA MP 12A)



MILITARY POLICE

FZA MP 12A

APPLICANT FOR EMPLOYMENT IN THE PUBLIC SERVICE
OR NON-CRIMINAL INQUIRY

Name.....

Alias.....Sex.....

Nationality.....

Applicant for.....

Station.....

For use in C3

OEB No.....

CB/CP No.....

Class.....

RIGHT HAND

1. R. Thumb	2. R. Forefinger	3. R. Middle Finger	4. R. Ring Finger	5. R. Little Finger
(Fold)				(Fold)

LEFT HAND

6. L. Thumb	7. L. Forefinger	8. L. Middle Finger	9. L. Ring Finger	10. L. Little Finger
(Fold)				(Fold)

LEFT HAND — IMPRESSION OF THUMBS — RIGHT HAND

Impression of LEFT fingers taken Simultaneously	Impression of THUMBS taken Simultaneously		Impression o RIGHT fingers taken Simultaneously
	Left Hand	Right Hand	
(Fold)			(Fold)

Fingerprints taken by..... No.....Rank.....Date.....

Classified by..... Identified with criminal record.....

Searched by..... No criminal record.....

Tested by.....

DESCRIPTION

Date and Place of Birth.....Age.....

Country of Birth (if alien).....

District.....

Village.....

Tribe.....

Chief.....

Address.....

Marks, scars, peculiarities.....

Complexion.....

Hair.....

Eyes.....

Height.....

Occupation.....

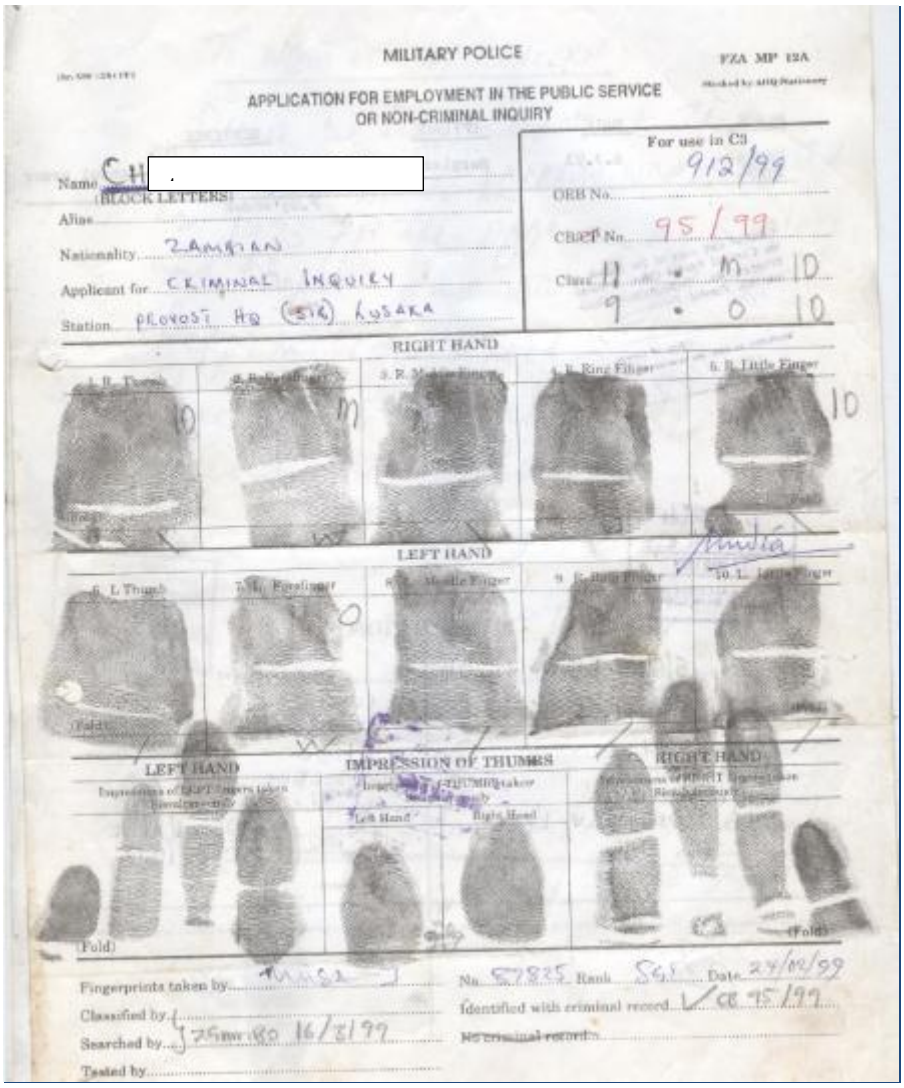
Date first entry Zambia (if applicable).....

Date and Place of issue and.....

Serial No. of Passport or NRC.....

Cell Phone number.....

C.2 Applicantion for Employment in Public Service or non-Criminal Inquiry filled in(FZA MP 12A)



MILITARY POLICE

FZA MP 12A

APPLICATION FOR EMPLOYMENT IN THE PUBLIC SERVICE
OR NON-CRIMINAL INQUIRY

For use in CB

Name CH [redacted]
(BLOCK LETTERS)

Alias

Nationality ZAMBIAN

Applicant for CRIMINAL INQUIRY

Station PROVOST HQ (SK) LUSAKA

ORB No. 912/99

CB/EP No. 95/99

Class H • M 10
9 • O 10

RIGHT HAND

L. R. Thumb | L. R. Forefinger | L. R. Middle Finger | L. R. Ring Finger | L. R. Little Finger

LEFT HAND

L. L. Thumb | L. L. Forefinger | L. L. Middle Finger | L. L. Ring Finger | L. L. Little Finger

LEFT HAND | IMPRESSION OF THUMBS | RIGHT HAND

Left Hand | Right Hand

(Fold) (Fold)

Fingerprints taken by MWS2

No. 57825 Rank SGT Date 24/09/99

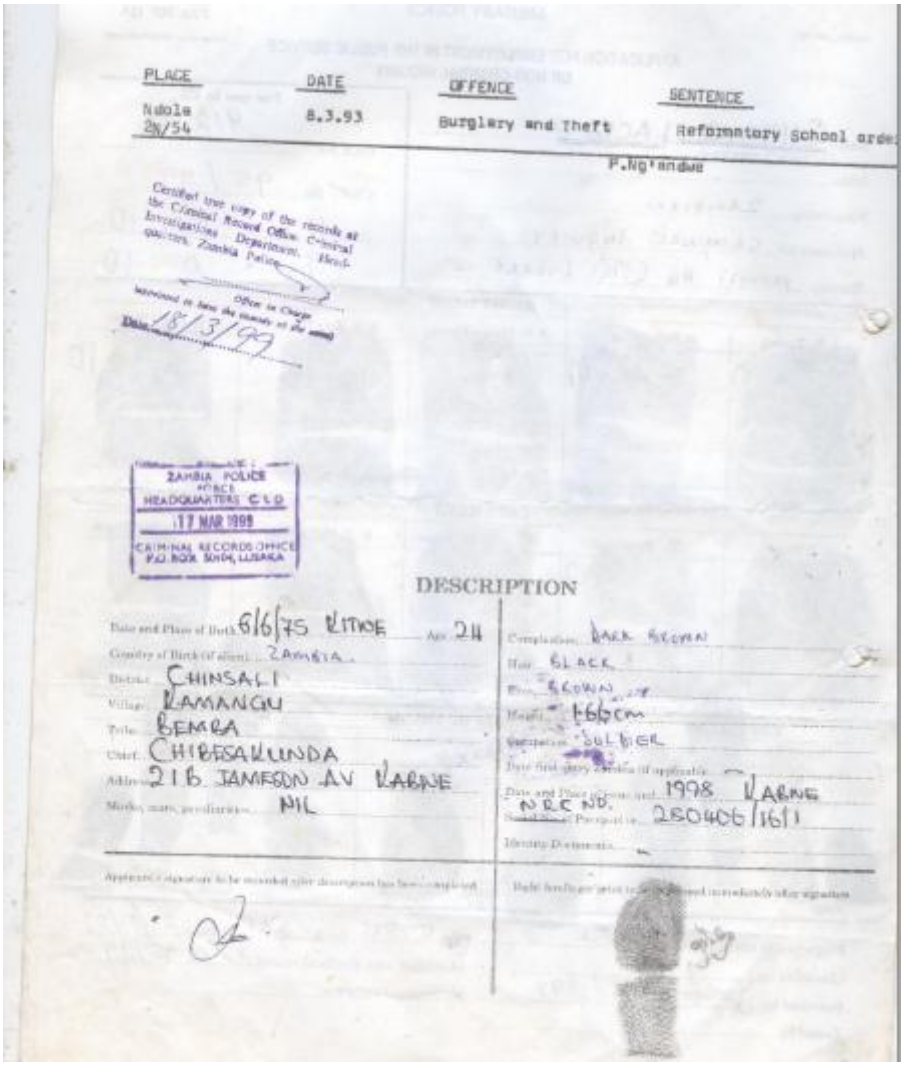
Classified by

Identified with criminal record CB 95/99

Searched by 75mw 180 16/8/99

~~No criminal record~~

Tested by



PLACE	DATE	OFFENCE	SENTENCE
Ndola 2A/54	8.3.93	Burglary and Theft	Reformatory School ord

P.Ng'andwe

Certified true copy of the records at the Criminal Record Office, Criminal Investigations Department, Police Headquarters, Zambia Police, Ndola.

Officer in Charge

Date 18/3/99

ZAMBIA POLICE FORCE HEADQUARTERS C.I.D. 17 MAR 1999 CRIMINAL RECORDS OFFICE P.O. BOX 50409, LUSAKA

DESCRIPTION

Date and Place of Birth 6/6/75 KITWE Age 24

Country of Birth of alien ZAMBIA

District CHINSALI

Village KAMANGU

Tribe BEMBA

Chief CHIBESAKUNDA

Address 216 JAMESON AV KABWE

Marks, scars, peculiarities NIL

Complexion DARK BROWN

Hair BLACK

Eyes BROWN

Height 166cm

Occupation SOLDIER

Date and Place of Arrest 1998 KABWE

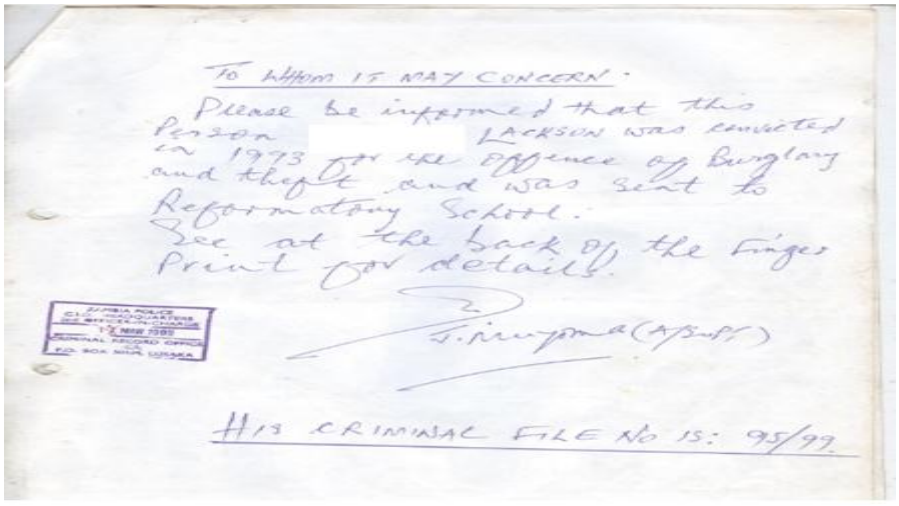
NRC NO. 280406/1611

Identity Documents —

Applicant's signature to be recorded after description has been completed

Right forefinger print to be obtained immediately after signature

C.3 Criminal Security Vetting Police reply Letter (FZA MP 12A)



TO WHOM IT MAY CONCERN.

Please be informed that this person LACKSON was convicted in 1993 for the offence of Burglary and theft and was sent to Reformatory School.

See at the back of the Finger Print for details.

ZAMBIA POLICE CRIMINAL INVESTIGATIONS OFFICER-IN-CHARGE 17 MAR 1999 CRIMINAL RECORD OFFICE P.O. BOX 50409, LUSAKA

S. Mwamba (A/Supt)

HIS CRIMINAL FILE NO IS: 95/99.

APPENDIX D: Fingerprint Biometrics System Parameter Specifications

D.1 Digital Personal fingerprint Scanner Parameter Specification

Table D.1 SDK Digital Persona fingerprint Scanner Specification

Template size	310 or 1152 Byte
Rotation	0 – 360 degree
FAR	<= 0.001%
FRR	<= 2.0%
Registration time	0.5 second
Average verification speed	2500 pieces/second
Image quality	>=300DPI

D.2 FBI Prescribed Design parameters specifications for AFIS

Table D.2 FBI Prescribed Design specifications for AFIS

Parameter	Requirement	
	IAFIS IQS (4-finger scanners at 500 dpi)	**PIV IQS** (single-finger scanners)
Area *height* (h) × *width* (w)	$h \geq 45.7$ mm (1.8") and $w \geq 73.2$ mm (2.88")	$h \geq 16.5$ mm (0.650") and $w \geq 12.8$ mm (0.504")
Native resolution R_N	$R_N \geq 500$ dpi	$R_N \geq 500$ dpi
Output resolution R_O	$R_O = 500$ dpi ± 1%	$R_O = 500$ dpi ± 2%
Gray-level quantization	256 gray-levels (8 bit per pixel)	256 gray-levels (8 bit per pixe
Gray range *DR*	for at least 80% of the image: $DR \geq 200$ for at least 99% of the image: $DR \geq 128$	for at least 80% of the image: $DR \geq$
Geometric accuracy D_{AC} (ACross-bar) D_{AL} (ALong-bar)	for at least 99% of the test: $D_{AC} \leq 1\%$ $D_{AL} \leq 0.016$"	for at least 99% of the test: $D_{AC} \leq 1.8\%$ $D_{AL} \leq 0.027$"
Gray level uniformity[2]	for at least 99% of the cases: $D_{RC}^{dark} \leq 1$; $D_{RC}^{light} \leq 2$ for least for 99.9% of the pixels: $D_{PP}^{dark} \leq 8$; $D_{PP}^{light} \leq 22$ for every two small areas: $D_{SA}^{dark} \leq 3$; $D_{SA}^{light} \leq 12$	for at least 99% of the cases: $D_{RC}^{dark} \leq 1.5$; $D_{RC}^{light} \leq 3$ for at least 99% of the pixels: $D_{PP}^{dark} \leq 8$; $D_{PP}^{light} \leq 22$ for every two small areas: $D_{SA}^{dark} \leq 3$; $D_{SA}^{light} \leq 12$
I/O linearity[3] D_{Lin}	$D_{Lin} \leq 7.65$	No requirements
Spatial frequency response	$MTF_{min}(f) \leq MTF(f) \leq 1.05$ see Nill (2005) for IAFIS $MTF_{min}(f)$	$MTF_{min}(f) \leq MTF(f) \leq 1.12$ see Nill (2006) for PIV $MTF_{min}(f)$
Signal-to-noise ratio[4] *SNR*	$SNR \geq 125$	$SNR \geq 70.6$

APPENDIX E: Reports

1 of 1 100% Find | Next

Man Number Register Arranged By Barracks

unit	Name	Rank	Service No	Sex	DOB	Phone	Corps	Hair	Eyes	Height
1 Infantry Regiment	PUNDA MULE	CAPT	4337	Male	15 Jun 1977	0955787698	Infantry	BLACK	WHITE	1.78M
	GREZY MWALYA	2LT	4362	Male	03 Feb 1965	0977725951	Artillery	BLACK	WHITE	1.79M
	GOLA MAN	CAPT	4536	Male	15 Jun 1977	0967906789	Infantry	BROWN	BLACK	1.90M
10 Mediam Regiment	KENNY MWENE	CAPT	4353	Male	31 Dec 2008	0977415234	Medical	BLACK	RES	1.78M
	KENNY MWENE	CAPT	4356	Male	05 Mar 1978	0977415234	Medical	BLACK	RES	1.78M
	MARY BWALYA	SSGT	4521	Male	05 Mar 1979	0955137809	Artillery	BLACK	NRMAL	1.45M
	JUMA NGOHI	CPL	7909	Male	31 Dec 2008	097890000	Artillery	BROWN	WHITE	1.89M
	MUKULA MUBA	CPL	93438	Male	31 Dec 2008	0955578907	Artillery	BLACK	RED	1.89M
17 Calvery Regiment	DANNY MUNYAU	COL	2308	Male	21 Jul 1956	0977734444	Armour	BLACK	NORMAL	1.73
	BOLO HULAMA	BRIG GEN	4364	Male	03 Feb 1965	0977725951	Infantry	BLACK	WHITE	1.70M
	BULOWA MWELWA	MAJ	4365	Male	03 Feb 1965	0977725952	Infantry	BLACK	WHITE	1.70M
	NGOZI PHIR1	SSGT	4569	Male	31 Dec 2008	095564567	Armour	BLACK	RED	1.34M
2 Infantry Regiment	JOSEPH PHIRI	COL	78901	Male	08 Feb 1995	097455434	Transport and Logist	BLACK	RED	1.72M

Figure E.1 Service Register Arranged by Unit

E.2 Military Personnel Comprehensive Register

Military Personnel Comprehensive Register

Name	Rank	Man No	Sex	DOB	Enlistment Date	Peculiarity	Complexion	Hair	Height	Corp
DANNY MUNYAU	COL	2308	Male	21 Jul 1956	23 Jun 2010	NIL	BLACK	BLACK	1.73	Armou
SANDROS KUMWENDA	LT COL	3211	Male	05 Jun 1960	28 Oct 2014	NILL	DARK	BLACK	1.73M	Armou
Chinyama T	MAJ	3912	Male	22 Jul 2015	17 Mar 2014	nil	light	black	1.8m	Medic
JOSEPH KALUNGA	MAJ	3913	Male	05 Mar 1978	31 Dec 2008	NIL	BLACK	BLACK	1.72M	Infant
EMMANUEL HAMUDUDU	CAPT	4036	Female	22 Jun 1970	13 Feb 2006	NIL	LIGHT	BLACK	1.53	Signal
PUNDA MULE	CAPT	4337	Male	15 Jun 1977	21 Jul 1989	NIL	BROWN	BLACK	1.78M	Infant
KENNY MWENE	CAPT	4353	Male	31 Dec 2008	31 Dec 2008	NIL	BLACK	BLACK	1.78M	Medic
DERRICH C	LT GEN	4355	Male	31 Dec 2008	31 Dec 2008	NIL	BLACK	BLACK	1.72M	Infant
KENNY MWENE	CAPT	4356	Male	05 Mar 1978	16 Jan 2003	NIL	BLACK	BLACK	1.78M	Medic
MUTALE NAMOS	O/CDT	4357	Male	31 Dec 2008	31 Dec 2008	NIL	BLACK	BLACK	1.70M	Infant
DANNY HAMBENZO	PTE	4358	Male	31 Dec 2008	31 Dec 2008	NIL	BLACK	BLACK	1.70M	Infant
BENADO HAMBENZO	PTE	4359	Male	16 Feb 1966	08 Jun 2004	NIL	BLACK	BLACK	1.70M	Infant
DEGY HAMBENO	MAJ	4360	Male	16 Feb 1966	08 Jun 2004	NIL	BLACK	BLACK	1.70M	Artille
GREBY BWALYA	2LT	4361	Male	03 Feb 1965	08 Jun 2004	NIL	BROWN	BLACK	1.79M	Artille
GREZY MWALYA	2LT	4362	Male	03 Feb 1965	08 Jun 2004	NIL	BROWN	BLACK	1.79M	Artille
FOHO LOS	CAPT	4363	Male	03 Feb 1965	08 Jun 2004	NIL	BROWN	BLACK	1.79M	Infant
BOLO HULAMA	BRIG GEN	4364	Male	03 Feb 1965	08 Jun 2004	NIL	BROWN	BLACK	1.70M	Infant
BULOWA MWELWA	MAJ	4365	Male	03 Feb 1965	08 Jan 2000	NIL	BROWN	BLACK	1.70M	Infant

Figure E.2 Military Personnel Comprehensive Register

E.3 Military Personnel Comprehensive Case Register

Comprehensive Case Register

Name	Service No	Case Date	Case No	NRC No	Unit	Type Offence	Description	Sentence
DANNY MUNYAU	2308	23 Jan 2000	ZB0001	12222/45/1	17 Calvery Regiment	Criminal	THEFT	COMMITED TO C(MASHUAL
DANNY MUNYAU	2308	20 Feb 2002	ZB0002	12222/45/1	17 Calvery Regiment	Criminal	INSURBODINATION	REDUCTION IN RA
DANNY MUNYAU	2308	13 Mar 2003	ZB0003	12222/45/1	17 Calvery Regiment	Criminal	THEFT	COMMITED TO C(MASHUAL
DANNY MUNYAU	2308	11 Apr 2003	ZB0004	12222/45/1	17 Calvery Regiment	Criminal	INSURBODINATION	REDUCTION IN RA
SANDROS KUMWENDA	3211	23 Jan 2000	ZB0005	81223/02/1	64 Armored Regiment	Criminal	THEFT	COMMITED TO C(MASHUAL
Chinyama T	3912	23 Jan 2000	ZB0006	233333/89/2	Army HQ	Civil	CHILD MAINTAINANCE	DEDUCT K 200 F(MANTAINANCE
KENNY MWENE	4353	29 Jan 2000	ZB0007	711231/02/1	10 Mediam Regiment	Criminal	THEFT	WAITING TRIAL
EMMANUEL HAMUDUDU	4036	29 Jan 2000	ZB0008	233334/12/1	Army HQ	Criminal	THEFT	2 YEAR WITH HAI
BULOWA MWELWA	4365	01 Jun 2001	ZB0009	546570/13/1	17 Calvery Regiment	Criminal	THEFT BY PUBLIC SERVANT	WAITING TRIAL
DANNY MUNYAU	2308	03 Jun 2001	SB009	12222/45/1	17 Calvery Regiment	Criminal	THEFT BY PUBLIC SERVANT	DISCHARGED FRC MILITARY
Chinyama T	3912	23 Jan 2000	ZB00011	233333/89/2	Army HQ	Criminal	THEFT	COMMITED TO C(MASHUAL
JOSEPH PHIRI	78901	23 Jan 2000	ZB0011	344444/09/1	2 Infantry Regiment	Domestic	CHILD MAINTAINANCE	2 YEARS SIMPLE IMPRISONMENT

Figure E.3 Military Personnel Comprehensive Case Register

E.4Service Personel Case Report Filtered by Name

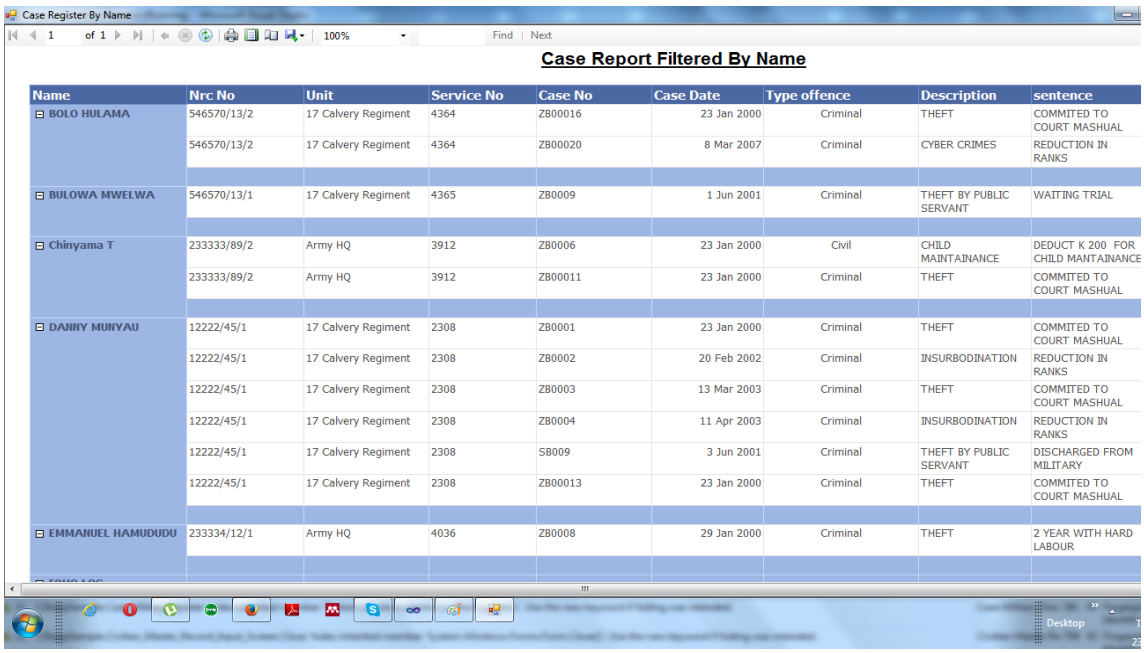


Figure E.4 Service Personel Case Report Filtered by Name

APPENDIX F: APPLICATION CODES

Appendix section F, contains some applications codes for the selected objects of the developed system. The researcher considered the bulkiness of computer codes to just pick few forms to illustrate the concept behind the forms operations. Note that if all codes were to be inserted in this report the document could be so huge. The following are some of the forms were code had been extracted:

F.1 Logic Form Code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using System.Runtime.InteropServices;
using System.IO;
using System.Data.Odbc;
using MySql.Data.MySqlClient;

namespace CSharpSample
{
public partial class login : Form
{
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";

MySqlConnection con;
MySqlCommand cmd;
MySqlDataReader rddata;
// OdbcDataReader rddata;
//OdbcDataReader rddata;
public login()
{
InitializeComponent();
}

privatevoid ok1_Click(object sender, EventArgs e)
{
string Connector1;
Connector1 = levelaccess.Text;
switch (Connector1)
{
case "Provost HQ":
```

```
con = newMySqlConnection(mySQLconn);

con.Open();
cmd = newMySqlCommand("SELECT * FROM password WHERE username ='" +
username.Text + "'AND Password = '" + password.Text + "'", con);

lblstatus.Text = "User Exists";

rddata = cmd.ExecuteReader();
while (rddata.Read())
{

}
if (rddata.HasRows == true)
{
Menu1 f1 = newMenu1();
f1.ShowDialog(); // Shows Form2
login.ActiveForm.Close();
}
else
{
//Response.Redirect("login.aspx");
lblstatus.Text = "Login failure! type correct username and password";

//cmd.ExecuteNonQuery();
rddata.Close();
con.Close();
// lblstatus.Text = "Record picked successfully";

}

break;
case "Sentry":
con = newMySqlConnection(mySQLconn);
con.Open();
md = newMySqlCommand("SELECT * FROM passwords WHERE username ='" +
username.Text + "'AND Password = '" + password.Text + "'", con);

lblstatus.Text = "User Exists";

rddata = cmd.ExecuteReader();
while (rddata.Read())
{
}
if (rddata.HasRows == true)
{
ChooseScannerForm f3 = new ChooseScannerForm();
f3.ShowDialog(); // Shows Form2
login.ActiveForm.Close();
}
```

```
else
{
//Response.Redirect("login.aspx");
lblstatus.Text = "Login failure! type correct username and password";

//cmd.ExecuteNonQuery();
rddata.Close();
con.Close();
// lblstatus.Text = "Record picked successfully";
}

break;
case "Administrator":
con = newMySqlConnection(mySQLconn);
con.Open();
cmd = newMySqlCommand("SELECT * FROM passwordg WHERE username ='" +
username.Text + "'AND Password = '" + password.Text + "'", con);
lblstatus.Text = "User Exists";

rddata = cmd.ExecuteReader();
while (rddata.Read())
{
}
if (rddata.HasRows == true)
{
administrator f3 = new administrator();
f3.ShowDialog(); // Shows Form2


}
else
{
//Response.Redirect("login.aspx");
lblstatus.Text = "Login failure! type correct username and password";

//cmd.ExecuteNonQuery();
rddata.Close();
con.Close();
// lblstatus.Text = "Record picked successfully";

}

break;


}
}

privatevoid ok_Click(object sender, EventArgs e)
{
```

```
this.Close();

}}}
```

F.2 BitmapFormat Class Code

```
using System.IO;
namespace CSharpSample
{
public class BitmapFormat
{
public struct BITMAPFILEHEADER
{
public ushort bfType;
public int bfSize;
public ushort bfReserved1;
public ushort bfReserved2;
public int bfOffBits;
}

public struct MASK
{
public byte redmask;
public byte greenmask;
public byte bluemask;
public byte rgbReserved;
}

public struct BITMAPINFOHEADER
{
public int biSize;
public int biWidth;
public int biHeight;
public ushort biPlanes;
public ushort biBitCount;
public int biCompression;
public int biSizeImage;
public int biXPelsPerMeter;
public int biYPelsPerMeter;
public int biClrUsed;
public int biClrImportant;
}

/*****************************************
```

```
Picture Rotate

**********************************************/
public static void RotatePic(byte[] BmpBuf, int width, int height, ref byte[] ResBuf)
{
int RowLoop = 0;
int ColLoop = 0;
int BmpBuflen = width * height;

try
{
for (RowLoop = 0; RowLoop < BmpBuflen; )
{
for (ColLoop = 0; ColLoop < width; ColLoop++)
{
ResBuf[RowLoop + ColLoop] = BmpBuf[BmpBuflen - RowLoop - width + ColLoop];
}

RowLoop = RowLoop + width;
}
}
catch (Exception ex)
{
//ZKCE.SysException.ZKCELogger logger = new ZKCE.SysException.ZKCELogger(ex);
//logger.Append();
}
}

/**********************************************
StructToBytes
**********************************************/
public static byte[] StructToBytes(object StructObj, int Size)
{
int StructSize = Marshal.SizeOf(StructObj);
byte[] GetBytes = new byte[StructSize];
try
{
IntPtr StructPtr = Marshal.AllocHGlobal(StructSize);
Marshal.StructureToPtr(StructObj, StructPtr, false);
Marshal.Copy(StructPtr, GetBytes, 0, StructSize);
Marshal.FreeHGlobal(StructPtr);
if (Size == 14)
{
byte[] NewBytes = new byte[Size];
```

```
int Count = 0;
int Loop = 0;

for (Loop = 0; Loop < StructSize; Loop++)
{
if (Loop != 2 && Loop != 3)
{
NewBytes[Count] = GetBytes[Loop];
Count++;
}
}
return NewBytes;
}
else
{
return GetBytes;
}
}
catch (Exception ex)
{
//ZKCE.SysException.ZKCELogger logger = new ZKCE.SysException.ZKCELogger(ex);
//logger.Append();

return GetBytes;
}
}

/*********************************************
GetBitmap  Function
*********************************************/
public static void GetBitmap(byte[] buffer, int nWidth, int nHeight, ref MemoryStream ms)
{
int ColorIndex = 0;
ushort m_nBitCount = 8;
int m_nColorTableEntries = 256;
byte[] ResBuf = new byte[nWidth * nHeight];
try
{
BITMAPFILEHEADER BmpHeader = new BITMAPFILEHEADER();
BITMAPINFOHEADER BmpInfoHeader = new BITMAPINFOHEADER();
MASK[] ColorMask = new MASK[m_nColorTableEntries];
BmpInfoHeader.biSize = Marshal.SizeOf(BmpInfoHeader);
BmpInfoHeader.biWidth = nWidth;
BmpInfoHeader.biHeight = nHeight;
```

```
BmpInfoHeader.biPlanes = 1;
BmpInfoHeader.biBitCount = m_nBitCount;
BmpInfoHeader.biCompression = 0;
BmpInfoHeader.biSizeImage = 0;
BmpInfoHeader.biXPelsPerMeter = 0;
BmpInfoHeader.biYPelsPerMeter = 0;
BmpInfoHeader.biClrUsed = m_nColorTableEntries;
BmpInfoHeader.biClrImportant = m_nColorTableEntries;
BmpHeader.bfType = 0x4D42;
BmpHeader.bfOffBits = 10 + Marshal.SizeOf(BmpInfoHeader) + BmpInfoHeader.biClrUsed * 4;
BmpHeader.bfSize = BmpHeader.bfOffBits + ((((BmpInfoHeader.biWidth *
BmpInfoHeader.biBitCount + 31) / 32) * 4) * BmpInfoHeader.biHeight);
BmpHeader.bfReserved1 = 0;
BmpHeader.bfReserved2 = 0;
ms.Write(StructToBytes(BmpHeader, 14), 0, 14);
ms.Write(StructToBytes(BmpInfoHeader, Marshal.SizeOf(BmpInfoHeader)), 0,
Marshal.SizeOf(BmpInfoHeader));
for (ColorIndex = 0; ColorIndex < m_nColorTableEntries; ColorIndex++)
{
ColorMask[ColorIndex].redmask = (byte)ColorIndex;
ColorMask[ColorIndex].greenmask = (byte)ColorIndex;
ColorMask[ColorIndex].bluemask = (byte)ColorIndex;
ColorMask[ColorIndex].rgbReserved = 0;

ms.Write(StructToBytes(ColorMask[ColorIndex], Marshal.SizeOf(ColorMask[ColorIndex])), 0,
Marshal.SizeOf(ColorMask[ColorIndex]));
}

RotatePic(buffer, nWidth, nHeight, ref ResBuf);
ms.Write(ResBuf, 0, nWidth * nHeight);
}
catch (Exception ex)
{
// ZKCE.SysException.ZKCELogger logger = new ZKCE.SysException.ZKCELogger(ex);
// logger.Append();
}
}

/*********************************************
WriteBitmap
*********************************************/
public static void WriteBitmap(byte[] buffer, int nWidth, int nHeight)
{
int ColorIndex = 0;
```

```
ushort m_nBitCount = 8;
int m_nColorTableEntries = 256;
byte[] ResBuf = new byte[nWidth * nHeight];
try
{
BITMAPFILEHEADER BmpHeader = new BITMAPFILEHEADER();
BITMAPINFOHEADER BmpInfoHeader = new BITMAPINFOHEADER();
MASK[] ColorMask = new MASK[m_nColorTableEntries];
BmpInfoHeader.biSize = Marshal.SizeOf(BmpInfoHeader);
BmpInfoHeader.biWidth = nWidth;
BmpInfoHeader.biHeight = nHeight;
BmpInfoHeader.biPlanes = 1;
BmpInfoHeader.biBitCount = m_nBitCount;
BmpInfoHeader.biCompression = 0;
BmpInfoHeader.biSizeImage = 0;
BmpInfoHeader.biXPelsPerMeter = 0;
BmpInfoHeader.biYPelsPerMeter = 0;
BmpInfoHeader.biClrUsed = m_nColorTableEntries;
BmpInfoHeader.biClrImportant = m_nColorTableEntries;
BmpHeader.bfType = 0x4D42;
BmpHeader.bfOffBits = 14 + Marshal.SizeOf(BmpInfoHeader) + BmpInfoHeader.biClrUsed * 4;
BmpHeader.bfSize = BmpHeader.bfOffBits + ((((BmpInfoHeader.biWidth * BmpInfoHeader.biBitCount + 31) /
32) * 4) * BmpInfoHeader.biHeight);
BmpHeader.bfReserved1 = 0;
BmpHeader.bfReserved2 = 0;
Stream FileStream = File.Open("finger.bmp", FileMode.Create, FileAccess.Write);
BinaryWriter TmpBinaryWriter = new BinaryWriter(FileStream);
TmpBinaryWriter.Write(StructToBytes(BmpHeader, 14));
TmpBinaryWriter.Write(StructToBytes(BmpInfoHeader, Marshal.SizeOf(BmpInfoHeader)));
for (ColorIndex = 0; ColorIndex < m_nColorTableEntries; ColorIndex++)
{
ColorMask[ColorIndex].redmask = (byte)ColorIndex;
ColorMask[ColorIndex].greenmask = (byte)ColorIndex;
ColorMask[ColorIndex].bluemask = (byte)ColorIndex;
ColorMask[ColorIndex].rgbReserved = 0;
TmpBinaryWriter.Write(StructToBytes(ColorMask[ColorIndex], Marshal.SizeOf(ColorMask[ColorIndex])));
}
RotatePic(buffer, nWidth, nHeight, ref ResBuf);
TmpBinaryWriter.Write(ResBuf);
FileStream.Close();
TmpBinaryWriter.Close();
}
catch (Exception ex)
{
```

```
//ZKCE.SysException.ZKCELogger logger = new ZKCE.SysException.ZKCELogger(ex);
//logger.Append();
}}}}
```

Appendix F.3 Service Personel Master Record Input Form Codes

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using System.Runtime.InteropServices;
using System.IO;
using System.Data.Odbc;
using MySql.Data.MySqlClient;

namespace CSharpSample
{
publicpartialclass Application_Form : Form
{
publicImage filter { get; set; }string imagename;
bool gConnected = false;
byte[] g_FPBuffer;
int g_FPBufferSize = 0;
bool g_bIsTimeToDie = false;
IntPtr g_Handle = IntPtr.Zero;
IntPtr g_biokeyHandle = IntPtr.Zero;
IntPtr g_FormHandle = IntPtr.Zero;
int g_nWidth = 0;
int g_nHeight = 0;
bool g_IsRegister = false;
int g_RegisterTimeCount = 0;
int g_RegisterCount = 0;
constint REGISTER_FINGER_COUNT = 3;

byte[][] g_RegTmps = newbyte[3][];
byte[] g_RegTmp = newbyte[2048];
byte[] g_VerTmp = newbyte[2048];
```

```
constint MESSAGE_FP_RECEIVED = 0x0400 + 6;

[DllImport("user32.dll", EntryPoint = "SendMessageA")]
publicstaticexternint SendMessage(IntPtr hwnd, int wMsg, IntPtr wParam, IntPtr lParam);
public Application_Form()
{
InitializeComponent();
}
privatevoid Ok_Click(object sender, EventArgs e)
{

}

privatevoid Application_Form_Load(object sender, EventArgs e)
{
g_FormHandle = this.Handle;
fillbox1();
fillbox();
}
privatevoid fillbox()
{
MySqlDataReader myreader;
//ma.Items.Clear();
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
string selectSQL = " SELECT * from unit ";
MySqlConnection con = newMySqlConnection(mySQLconn);
MySqlCommand cmd = newMySqlCommand(selectSQL, con);
try
{
con.Open();
myreader = cmd.ExecuteReader();
while (myreader.Read())
{
string scountry = myreader.GetString("unit");
unit.Items.Add(scountry);
}
myreader.Close();
}
catch (Exception err)
{
lblstatus.Text = " Error reading list";
lblstatus.Text += err.Message;
```

```
}
}
privatevoid fillbox1()
{
MySqlDataReader myreader;
//ma.Items.Clear();
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
string selectSQL = " SELECT * from RANK ";
MySqlConnection con = newMySqlConnection(mySQLconn);
MySqlCommand cmd = newMySqlCommand(selectSQL, con);
try
{
con.Open();
myreader = cmd.ExecuteReader();
while (myreader.Read())
{
string scountry = myreader.GetString("rankname");
rank.Items.Add(scountry);
}
myreader.Close();
}
catch (Exception err)
{
lblstatus.Text = " Error reading list";
lblstatus.Text += err.Message;
}
}
privatevoid serviceno_TextChanged(object sender, EventArgs e)
{
MySqlDataReader rddata;
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
string selectSQL;
selectSQL = " SELECT * from mastertablemilitary where serviceno= '" + serviceno.Text + "'";
MySqlConnection con = newMySqlConnection(mySQLconn);
MySqlCommand cmd1 = newMySqlCommand(selectSQL, con);
try
{
con.Open();
rddata = cmd1.ExecuteReader();
// int added = 0;
while (rddata.Read())
{
```

```
//usergroup.Enabled = true;
serviceno.Text = rddata["serviceno"].ToString();
fullname.Text = rddata["fullname"].ToString();
unit.Text = rddata["unit"].ToString();
nrcno.Text = rddata["nrcno"].ToString();
//dob.Text = rddata["dob"].ToString();
// enlisteddate.Text = rddata["EnlistementDate"].ToString();
chief.Text = rddata["chief"].ToString();
village.Text = rddata["village"].ToString();
district.Text = rddata["district"].ToString();
tribe.Text = rddata["tribe"].ToString();
peculiarity.Text = rddata["peculiarity"].ToString();
complexion.Text = rddata["complexion"].ToString();
hair.Text = rddata["hair"].ToString();
eyes.Text = rddata["eyes"].ToString();
height.Text = rddata["height"].ToString();
mobileno.Text = rddata["mobileno"].ToString();
corps.Text = rddata["corps"].ToString();
rank.Text = rddata["rank"].ToString();
sex.Text = rddata["sex"].ToString();
address.Text = rddata["address"].ToString();
placebirth.Text = rddata["placebirth"].ToString();
mcategory.Text = rddata["mcategory"].ToString();
byte[] imgg = (byte[])(rddata["photo"]);
byte[] imgg1 = (byte[])(rddata["rthumb"]);
if (imgg == null)
photo.Image = null;
else
{
MemoryStream mstream = newMemoryStream(imgg);
MemoryStream mstream1 = newMemoryStream(imgg1);
photo.Image = System.Drawing.Image.FromStream(mstream);
rthumb.Image = System.Drawing.Image.FromStream(mstream1);
}}
rddata.Close();
lblstatus.Text = "";
}
catch (Exception err)
{
lblstatus.Text = " Error Geting Author. ";
lblstatus.Text = err.Message;
}
finally
{
```

```
con.Close();
// photoimage.Image = null;
//fimage.Image = null;
}
privatevoid btnConnect_Click(object sender, EventArgs e)
{
if (!gConnected)
{
int ret = 0;
byte[] paramValue = newbyte[64];

// Enable log
Array.Clear(paramValue, 0, paramValue.Length);
paramValue[0] = 1;
ZKFPCap.sensorSetParameterEx(g_Handle, 1100, paramValue, 4);
ret = ZKFPCap.sensorInit();
if (ret != 0)
{
MessageBox.Show("Init Failed, ret=" + ret.ToString());
return;
}
g_Handle = ZKFPCap.sensorOpen(0);

Array.Clear(paramValue, 0, paramValue.Length);
ZKFPCap.sensorGetVersion(paramValue, paramValue.Length);
ret = paramValue.Length;
Array.Clear(paramValue, 0, paramValue.Length);
ZKFPCap.sensorGetParameterEx(g_Handle, 1, paramValue, ref ret);
g_nWidth = BitConverter.ToInt32(paramValue, 0);
this.rthumb.Width = g_nWidth;
ret = paramValue.Length;
Array.Clear(paramValue, 0, paramValue.Length);
ZKFPCap.sensorGetParameterEx(g_Handle, 2, paramValue, ref ret);
g_nHeight = BitConverter.ToInt32(paramValue, 0);
this.rthumb.Height = g_nHeight;
ret = paramValue.Length;
Array.Clear(paramValue, 0, paramValue.Length);
ZKFPCap.sensorGetParameterEx(g_Handle, 106, paramValue, ref ret);
g_FPBufferSize = BitConverter.ToInt32(paramValue, 0);
g_FPBuffer = newbyte[g_FPBufferSize];
Array.Clear(g_FPBuffer, 0, g_FPBuffer.Length);

// get vid&pid
ret = paramValue.Length;
```

```
Array.Clear(paramValue, 0, paramValue.Length);
ZKFPCap.sensorGetParameterEx(g_Handle, 1015, paramValue, ref ret);
int nVid = BitConverter.ToInt16(paramValue, 0);
int nPid = BitConverter.ToInt16(paramValue, 2);
// Manufacturer
ret = paramValue.Length;
Array.Clear(paramValue, 0, paramValue.Length);
ZKFPCap.sensorGetParameterEx(g_Handle, 1101, paramValue, ref ret);
string manufacturer =
System.Text.Encoding.ASCII.GetString(paramValue);
// Product
ret = paramValue.Length;
Array.Clear(paramValue, 0, paramValue.Length);
ZKFPCap.sensorGetParameterEx(g_Handle, 1102, paramValue, ref ret);
string product = System.Text.Encoding.ASCII.GetString(paramValue);
// SerialNumber
ret = paramValue.Length;
Array.Clear(paramValue, 0, paramValue.Length);
ZKFPCap.sensorGetParameterEx(g_Handle, 1103, paramValue, ref ret);
string serialNumber =
System.Text.Encoding.ASCII.GetString(paramValue);

// Fingerprint Alg
short[] iSize = newshort[24];
iSize[0] = (short)g_nWidth;
iSize[1] = (short)g_nHeight;
iSize[20] = (short)g_nWidth;
iSize[21] = (short)g_nHeight; ;
g_biokeyHandle = ZKFinger10.BIOKEY_INIT(0, iSize, null, null, 0);
if (g_biokeyHandle == IntPtr.Zero)
{
MessageBox.Show("BIOKEY_INIT failed");
return;
}

// Set allow 360 angle of Press Finger
ZKFinger10.BIOKEY_SET_PARAMETER(g_biokeyHandle, 4, 180);

// Set Matching threshold
ZKFinger10.BIOKEY_MATCHINGPARAM(g_biokeyHandle, 0, ZKFinger10.THRESHOLD_MIDDLE);

// Init RegTmps
for (int i = 0; i < 3; i++)
{
```

```
g_RegTmps[i] = newbyte[2048];
}

Thread captureThread = newThread(newThreadStart(DoCapture));
captureThread.IsBackground = true;
captureThread.Start();
g_bIsTimeToDie = false;

gConnected = true;
// btnRegister.Enabled = true;
// btnVerify.Enabled = true;
btnConnect.Text = "Disconnect Sensor";

txtPrompt.Text = "Please put your finger on the sensor";
}
else
{
FreeSensor();

ZKFinger10.BIOKEY_DB_CLEAR(g_biokeyHandle);
ZKFinger10.BIOKEY_CLOSE(g_biokeyHandle);

gConnected = false;
//btnRegister.Enabled = false;
// btnVerify.Enabled = false;
btnConnect.Text = "Connect Sensor";
}
}

privatevoid FreeSensor()
{
g_bIsTimeToDie = true;
Thread.Sleep(100);
ZKFPCap.sensorClose(g_Handle);

// Disable log
byte[] paramValue = newbyte[4];
paramValue[0] = 0;
ZKFPCap.sensorSetParameterEx(g_Handle, 1100, paramValue, 4);

ZKFPCap.sensorFree();
}

privatevoid DoCapture()
```

```
{
while (!g_bIsTimeToDie)
{
int ret = ZKFPCap.sensorCapture(g_Handle, g_FPBuffer, g_FPBufferSize);
if (ret > 0)
{
SendMessage(g_FormHandle, MESSAGE_FP_RECEIVED, IntPtr.Zero, IntPtr.Zero);
}
Thread.Sleep(30);
}
}

protectedoverridevoid DefWndProc(refMessage m)
{
switch (m.Msg)
{
case MESSAGE_FP_RECEIVED:
{
try
{
MemoryStream ms = newMemoryStream();
BitmapFormat.GetBitmap(g_FPBuffer, g_nWidth, g_nHeight, ref ms);
Bitmap bmp = newBitmap(ms);
this.rthumb.Image = bmp;

txtStatus.Text = "IMAGE_READY";

int ret = 0;
int id = 0;
//int tempret = 0;
int score = 0;
int quality = 0;
int getb = 0;

if (g_IsRegister)
{
Array.Clear(g_RegTmp, 0, g_RegTmp.Length);
ret = ZKFinger10.BIOKEY_EXTRACT(g_biokeyHandle, g_FPBuffer, g_RegTmp, 0);
// temp1.Text = ret.ToString();
if (ret > 0)
{
Array.Copy(g_RegTmp, g_RegTmps[g_RegisterTimeCount++], ret);

// Get fingerprint quality
```

```
quality = ZKFinger10.BIOKEY_GETLASTQUALITY();
txtQuality.Text = quality.ToString();

txtPrompt.Text = string.Format("Still press finger {0} time", REGISTER_FINGER_COUNT -
g_RegisterTimeCount);

if (g_RegisterTimeCount == REGISTER_FINGER_COUNT)
{
Array.Clear(g_RegTmp, 0, g_RegTmp.Length);

int size = 0;

unsafe
{
fixed (byte* Template1 = g_RegTmps[0])
{
fixed (byte* Template2 = g_RegTmps[1])
{
fixed (byte* Template3 = g_RegTmps[2])
{
byte*[] pTemplate = newbyte*[3] { Template1, Template2, Template3 };

size = ZKFinger10.BIOKEY_GENTEMPLATE(g_biokeyHandle, pTemplate, 3, g_RegTmp);
}}}
if (size > 0)
getb = ZKFinger10.BIOKEY_DB_ADD(g_biokeyHandle, ++g_RegisterCount, size, g_RegTmp);
// temp3.Text = g_RegTmp.LongCount().ToString();
ZKFinger10.BIOKEY_DB_ADD(g_biokeyHandle, ++g_RegisterCount, size, g_RegTmp);
txtPrompt.Text = string.Format("Register succeeded, fid={0}, totalCount={1}", g_RegisterCount,
ZKFinger10.BIOKEY_DB_COUNT(g_biokeyHandle));

g_IsRegister = false;
}
else
{
txtPrompt.Text = "Register failed";
}
g_RegisterTimeCount = 0;
}
}
else
{
txtPrompt.Text = "Extract template failed";
}
```

```
}
else
{
Array.Clear(g_VerTmp, 0, g_VerTmp.Length);
if ((ret = ZKFinger10.BIOKEY_EXTRACT(g_biokeyHandle, g_FPBuffer, g_VerTmp, 0)) > 0)
{
// Get fingerprint quality
quality = ZKFinger10.BIOKEY_GETLASTQUALITY();
txtQuality.Text = quality.ToString();
ret = ZKFinger10.BIOKEY_IDENTIFYTEMP(g_biokeyHandle, g_VerTmp, ref id, ref score);
int tempret = ret;


if (ret > 0)
{
txtPrompt.Text = string.Format("Identification success, id={0}, score={1}", id, score);
}
else
{
txtPrompt.Text = string.Format("Identification failed, score={0}", score);

}
}
else
{
txtPrompt.Text = "Extract template failed";
}
}
}
catch (Exception ex)
{
MessageBox.Show(ex.Message.ToString());
}
}
break;

default:
base.DefWndProc(ref m);
break;
}
}

privatevoid uploadphoto_Click(object sender, EventArgs e)
{
```

```
try
{
FileDialog fldlg = newOpenFileDialog();

//specify your own initial directory
fldlg.InitialDirectory = @":C\Users\bwalya sichula\Pictures";
//this will allow only those file extensions to be added
fldlg.Filter = "Image File (*.jpg;*.bmp;*.gif)|*.jpg;*.bmp;*.gif";
if (fldlg.ShowDialog() == DialogResult.OK)
{
imagename = fldlg.FileName;
Bitmap newimg = newBitmap(imagename);
photo.SizeMode = PictureBoxSizeMode.StretchImage;
pathphoto.Text = imagename;
photo.Image = (Image)newimg;
}
fldlg = null;
}
catch (System.ArgumentException ae)
{
imagename = " ";
MessageBox.Show(ae.Message.ToString());
}
catch (Exception ex)
{
MessageBox.Show(ex.Message.ToString());
}}
privatevoid insserty_Click(object sender, EventArgs e)
{
Byte[] imageBt,imageBt1 = null;
FileStream fstream = newFileStream(this.pathphoto.Text, FileMode.Open, FileAccess.Read);
FileStream fstream1 = newFileStream(this.pathrthumb.Text, FileMode.Open, FileAccess.Read);
BinaryReader br = newBinaryReader(fstream);
BinaryReader br1 = newBinaryReader(fstream1);
imageBt = br.ReadBytes((int)fstream.Length);
imageBt1 = br1.ReadBytes((int)fstream1.Length);
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
string Query = " Insert into
mastertablemilitary(serviceno,fullname,nrcno,unit,DOB,EnlistmentDate,corps,chief,village,district,tribe,sex,
peculiarity, complexion, hair, eyes, height, mobileno, address, photo, rthumb,placebirth,rank,mcategory) values ('"
+ serviceno.Text + "','" + fullname.Text + "','" + nrcno.Text + "','" + unit.Text + "','" + dob.Text + "','" +
enlisteddate.Text + "','" + corps.SelectedItem.ToString() + "','" + chief.Text + "','" + village.Text + "','" +
district.Text + "','" + tribe.Text + "','" + sex.SelectedItem.ToString() + "','" + peculiarity.Text + "','" +
```

```
complexion.Text + "'," + hair.Text + "'," + eyes.Text + "'," + height.Text + "'," + mobileno.Text + "'," +
address.Text + "', @IMG,@IMG1 ,'" + placebirth.Text + "'," + rank.SelectedItem.ToString() + "'," +
mcategory.Text + "' ); ";
MySqlConnection conn = newMySqlConnection(mySQLconn);
MySqlCommand cmd = newMySqlCommand(Query, conn);
MySqlDataReader myreader;
try
{
conn.Open();
cmd.Parameters.Add(newMySqlParameter("@IMG", imageBt));
cmd.Parameters.Add(newMySqlParameter("@IMG1", imageBt1));
myreader = cmd.ExecuteReader();
MessageBox.Show(" One Record Saved");
serviceno.Text = "";
fullname.Text = "";
nrcno.Text = "";
dob.Text = "";
chief.Text = "";
village.Text = "";
district.Text = "";
tribe.Text = "";
peculiarity.Text = "";
complexion.Text = "";
hair.Text = "";
eyes.Text = "";
height.Text = "";
mobileno.Text = "";
address.Text = "";
photo.Image = null;
rthumb.Image = null;

while (myreader.Read())
{}}
catch (Exception ex)
{
MessageBox.Show(ex.Message);
}}
privatevoid rank_SelectedIndexChanged(object sender, EventArgs e)
{
MySqlDataReader rddata;
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
string selectSQL;
selectSQL = " SELECT * from rank where rankname= '" + rank.SelectedItem.ToString() + "'";
```

```
MySqlConnection con = newMySqlConnection(mySQLconn);
MySqlCommand cmd1 = newMySqlCommand(selectSQL, con);
try
{
con.Open();
rddata = cmd1.ExecuteReader();
// int added = 0;
while (rddata.Read())
{
//usergroup.Enabled = true;
mcategory.Text = rddata["mcategory"].ToString();
//unitname.Text = rddata["unit"].ToString();
//formation1.Items.ToString() = rddata["formation1"].ToString();
}
rddata.Close();
lblstatus.Text = "";
}
catch (Exception err)
{
lblstatus.Text = " Error Geting Author. ";
lblstatus.Text = err.Message;
}
finally
{
con.Close();
// photoimage.Image = null;
//fimage.Image = null;
}}
privatevoid button1_Click(object sender, EventArgs e)
{

if (this.saveFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
this.rthumb.Image.Save(this.saveFileDialog1.FileName, System.Drawing.Imaging.ImageFormat.Png);
rthumb.Image = null;
}

privatevoid upload_Click(object sender, EventArgs e)
{


try
{
```

```
FileDialog fldlg = newOpenFileDialog();

//specify your own initial directory

fldlg.InitialDirectory = @":C\Users\bwalya sichula\Pictures";

//this will allow only those file extensions to be added

fldlg.Filter = "Image File
(*.jpg;*.bmp;*.png;*.gif)|*.jpg;*.bmp;*.png;*.gif";

if (fldlg.ShowDialog() == DialogResult.OK)
{

imagename = fldlg.FileName;

Bitmap newimg = newBitmap(imagename);

rthumb.SizeMode = PictureBoxSizeMode.StretchImage;
pathrthumb.Text = imagename;

rthumb.Image = (Image)newimg;

}

fldlg = null;
```

```
}

catch (System.ArgumentException ae)
{
imagename = " ";
MessageBox.Show(ae.Message.ToString());
}
catch (Exception ex)
{
MessageBox.Show(ex.Message.ToString());
}}
privatevoid edit_Click(object sender, EventArgs e)
{
MySqlDataReader rddata;
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
string selectSQL;
selectSQL = " UPDATE mastertablemilitary SET serviceno = '" + serviceno.Text + "', fullname ='" +
fullname.Text + "',nrcno ='" + nrcno.Text + "',unit ='" + unit.SelectedItem.ToString() + "', DOB ='" + dob.Text +
"',EnlistmentDate ='" + enlisteddate.Text + "', corps ='" + corps.SelectedItem.ToString() + "', chief ='" + chief.Text
+ "',village = '" + village.Text + "', district='" + district.Text + "', tribe = '" + tribe.Text + "',sex='" +
sex.SelectedItem.ToString() + "', peculiarity='" + peculiarity.Text + "',complexion = '" + complexion.Text + "',hair
= '" + hair.Text + "', eyes ='" + eyes.Text + "',height ='" + height.Text + "',mobileno ='" + mobileno.Text +
"',address ='" + address.Text + "',placebirth ='" + placebirth.Text + "', rank ='" + rank.SelectedItem.ToString() +
"',mcategory ='" + mcategory.Text + "' WHERE serviceno= '" + serviceno.Text + "'";
MySqlConnection con = newMySqlConnection(mySQLconn);
MySqlCommand cmd1 = newMySqlCommand(selectSQL, con);
try
{
con.Open();
rddata = cmd1.ExecuteReader();
rddata.Close();
lblstatus.Text = "";
}
catch (Exception err)
{
lblstatus.Text = " Error Geting Author. ";
lblstatus.Text = err.Message;
}
finally
{
con.Close();
```

```
MessageBox.Show(" One Record ammended");
serviceno.Text = "";
fullname.Text = "";
nrcno.Text = "";
dob.Text = "";
chief.Text = "";
village.Text = "";
district.Text = "";
tribe.Text = "";
peculiarity.Text = "";
complexion.Text = "";
hair.Text = "";
eyes.Text = "";
height.Text = "";
mobileno.Text = "";
address.Text = "";
photo.Image = null;
rthumb.Image = null;
}}
privatevoid Delete_Click(object sender, EventArgs e)
{
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
MySqlConnection con = newMySqlConnection(mySQLconn);
// MySqlCommand cmd1 = new MySqlCommand(selectSQL, con);
MySqlCommand cmd1 = newMySqlCommand("DELETE FROM mastertablemilitary  WHERE serviceno = '" +
serviceno.Text + "'", con);
//MySqlCommand cmd1 = new MySqlCommand(selectSQL, con);
try
{
con.Open();
cmd1.ExecuteNonQuery();
}
finally
{
con.Close();
MessageBox.Show(" One Record Deleted");
serviceno.Text = "";
fullname.Text = "";
nrcno.Text = "";
dob.Text = "";
chief.Text = "";
village.Text = "";
district.Text = "";
```

```
tribe.Text = "";
peculiarity.Text = "";
complexion.Text = "";
hair.Text = "";
eyes.Text = "";
height.Text = "";
mobileno.Text = "";
address.Text = "";
photo.Image = null;
rthumb.Image = null;
}}
privatevoid Close_Click(object sender, EventArgs e)
{
this.Close();
}}}
```

Appendix F.3 Case Military Input Form codes

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using System.Runtime.InteropServices;
using System.IO;
using System.Data.Odbc;
using MySql.Data.MySqlClient;
namespace CSharpSample
{
public partial class Case_Military : Form
{
public Case_Military()
{
InitializeComponent();
fillbox();
}
privatevoid fillbox()
{

MySqlDataReader myreader;

//ma.Items.Clear();
```

```
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
string selectSQL = " SELECT * from mastertablemilitary ";
MySqlConnection con = newMySqlConnection(mySQLconn);
MySqlCommand cmd = newMySqlCommand(selectSQL, con);

try
{
con.Open();
myreader = cmd.ExecuteReader();
while (myreader.Read())
{
string scountry = myreader.GetString("serviceno");
serviceno.Items.Add(scountry);
}
myreader.Close();
}
catch (Exception err)
{
lblstatus.Text = " Error reading list";
lblstatus.Text += err.Message;
}}
privatevoid groupBox1_Enter(object sender, EventArgs e)
{}
privatevoid Insert_Click(object sender, EventArgs e)
{
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
string Query = " Insert into casemilitary(serviceno, nrcno, unit, casedate, offence,typeoffence,sentence,
description, fullname,caseno) values ('" + serviceno.Text + "','" + nrcno.Text + "','"+ unit.Text + "','" +
casedate.Text + "','" + offence.Text + "','" + offencetype.SelectedItem.ToString()+ "','" + sentence.Text + "','" +
description.Text + "','" + fullname.Text + "','"+ caseno.Text + "'); ";
MySqlConnection conn = newMySqlConnection(mySQLconn);
MySqlCommand cmd = newMySqlCommand(Query, conn);
MySqlDataReader myreader;
try
{
conn.Open();

myreader = cmd.ExecuteReader();
MessageBox.Show(" One Record Saved");
serviceno.Text = "";
nrcno.Text = "";
```

```
offence.Text = "";
sentence.Text = "";
description.Text = "";
fullname.Text = "";
caseno.Text = "";
unit.Text = "";
While (myreader.Read())
{}}
catch (Exception ex)
{
MessageBox.Show(ex.Message);
}}
privatevoid serviceno_SelectedIndexChanged(object sender, EventArgs e)
{
MySqlDataReader rddata;
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
string selectSQL;
selectSQL = " SELECT * from mastertablemilitary where serviceno= '" + serviceno.Text + "'";
MySqlConnection con = newMySqlConnection(mySQLconn);
MySqlCommand cmd1 = newMySqlCommand(selectSQL, con);
try
{
con.Open();
rddata = cmd1.ExecuteReader();
// int added = 0;
while (rddata.Read())
{
//usergroup.Enabled = true;
serviceno.Text = rddata["serviceno"].ToString();
fullname.Text = rddata["fullname"].ToString();
unit.Text = rddata["unit"].ToString();
nrcno.Text = rddata["nrcno"].ToString();
}
rddata.Close();
lblstatus.Text = "";
}
catch (Exception err)
{
lblstatus.Text = " Error Geting Author. ";
lblstatus.Text = err.Message;
}
finally
{
```

```
con.Close();
// photoimage.Image = null;
//fimage.Image = null;
}
}

privatevoid caseno_TextChanged(object sender, EventArgs e)
{
MySqlDataReader rddata;
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
string selectSQL;
selectSQL = " SELECT * from casemilitary where caseno= '" + caseno.Text + "'";
MySqlConnection con = newMySqlConnection(mySQLconn);
MySqlCommand cmd1 = newMySqlCommand(selectSQL, con);
try
{
con.Open();
rddata = cmd1.ExecuteReader();
// int added = 0;
while (rddata.Read())
{
//usergroup.Enabled = true;
serviceno.Text = rddata["serviceno"].ToString();
fullname.Text = rddata["fullname"].ToString();
unit.Text = rddata["unit"].ToString();
nrcno.Text = rddata["nrcno"].ToString();
offence.Text = rddata["offence"].ToString();
offencetype.Text = rddata["typeoffence"].ToString();
casedate.Text = rddata["casedate"].ToString();
description.Text = rddata["description"].ToString();
caseno.Text = rddata["caseno"].ToString();
sentence.Text = rddata["sentence"].ToString();
}
rddata.Close();
lblstatus.Text = "";
}
catch (Exception err)
{
lblstatus.Text = " Error Geting Author. ";
lblstatus.Text = err.Message;
}
finally
{
```

```
con.Close();
// photoimage.Image = null;
//fimage.Image = null;}}
privatevoid Update_Click(object sender, EventArgs e)
{
MySqlDataReader rddata;
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
string selectSQL;
selectSQL = " UPDATE casemilitary SET serviceno = '" + serviceno.Text + "', nrcno ='" + nrcno.Text + "',unit ='"
+ unit.Text + "', casedate ='" + casedate.Text + "',typeoffence ='" + offencetype.SelectedItem.ToString() + "',
description ='" + description.Text + "', fullname ='" + fullname.Text + "',caseno ='" + caseno.Text + "' WHERE
caseno= '" + caseno.Text + "'";
MySqlConnection con = newMySqlConnection(mySQLconn);
MySqlCommand cmd1 = newMySqlCommand(selectSQL, con);
try
{
con.Open();
rddata = cmd1.ExecuteReader();
rddata.Close();
lblstatus.Text = "";
}
catch (Exception err)
{
lblstatus.Text = " Error Geting Author. ";
lblstatus.Text = err.Message;
}
finally
{
con.Close();
MessageBox.Show(" Information has been changed");

privatevoid delete_Click(object sender, EventArgs e)
{
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
MySqlConnection con = newMySqlConnection(mySQLconn);
// MySqlCommand cmd1 = new MySqlCommand(selectSQL, con);

MySqlCommand cmd1 = newMySqlCommand("DELETE FROM casemilitary WHERE caseno = '" + caseno.Text
+ "'", con);
//MySqlCommand cmd1 = new MySqlCommand(selectSQL, con);
try
{
```

```
con.Open();
cmd1.ExecuteNonQuery();
}
finally
{
con.Close();
MessageBox.Show("one Record Deleted ");
serviceno.Text = "";
nrcno.Text = "";
offence.Text = "";
sentence.Text = "";
description.Text = "";
fullname.Text = "";
caseno.Text = "";
unit.Text = "";
}
}

privatevoid clear_Click(object sender, EventArgs e)
{
MessageBox.Show(" Form Cleared for new transanction");
serviceno.Text = "";
nrcno.Text = "";
offence.Text = "";
sentence.Text = "";
description.Text = "";
fullname.Text = "";
caseno.Text = "";
unit.Text = "";
}

privatevoid Exit_Click(object sender, EventArgs e)
{
this.Close();
}}}
```

Appendix F.4 Fingerprint Analysis Input Form Code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
```

```
using AForge.Imaging.Filters;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using System.Runtime.InteropServices;
using System.IO;
using System.Data.Odbc;
using MySql.Data.MySqlClient;
using AForge.Imaging;
using AForge.Imaging.Filters;
using System.Globalization;
namespace CSharpSample
{
publicpartialclass Fingerprint_Analysis : Form
{
privateBitmap resultBitmap = null;
privateBitmap originalBitmap = null;
privateBitmap previewBitmap = null;
privateOtsu ot = newOtsu();
privateBitmap Otsuimage;
publicfloat contrast= 0;
publicBitmap NewBitmap;
public Fingerprint_Analysis()
{
InitializeComponent();
fillbox();
}
privatevoid fillbox()
{

MySqlDataReader myreader;

//ma.Items.Clear();

string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";

string selectSQL = " SELECT * from mastertable ";
MySqlConnection con = newMySqlConnection(mySQLconn);
MySqlCommand cmd = newMySqlCommand(selectSQL, con);

try
{
con.Open();
```

```
myreader = cmd.ExecuteReader();
while (myreader.Read())
{
string scountry = myreader.GetString("fullname");
dname.Items.Add(scountry);
}
myreader.Close();
}
catch (Exception err)
{
lblstatus.Text = " Error reading list";
lblstatus.Text += err.Message;
}

}
class ImageBrightnessNormalizer
{
privateconstfloat MIN_BRIGHTNESS = 0;
privateconstfloat MAX_BRIGHTNESS = 1;

publicstaticBitmap NormalizeImageBrightness(Bitmap image)
{
float minBrightness = MAX_BRIGHTNESS;
float maxBrightness = MIN_BRIGHTNESS;

/* Get the brightness range of the image. */
for (int x = 0; x < image.Width; x++)
{
for (int y = 0; y < image.Height; y++)
{
float pixelBrightness = image.GetPixel(x, y).GetBrightness();
minBrightness = Math.Min(minBrightness, pixelBrightness);
maxBrightness = Math.Max(maxBrightness, pixelBrightness);
}
}

/* Normalize the image brightness. */
for (int x = 0; x < image.Width; x++)
{
for (int y = 0; y < image.Height; y++)
{
Color pixelColor = image.GetPixel(x, y);
float normalizedPixelBrightness = (pixelColor.GetBrightness() - minBrightness) / (maxBrightness -
minBrightness);
```

```
Color normalizedPixelColor =
ColorConverter.ColorFromAhsb(pixelColor.A, pixelColor.GetHue(),
pixelColor.GetSaturation(), normalizedPixelBrightness);
image.SetPixel(x, y, normalizedPixelColor);
}}
return image;
}}
Privatevoid normalisation_Click(object sender, EventArgs e)
{
image2.Image = ImageBrightnessNormalizer.NormalizeImageBrightness(new
Bitmap(image1.Image));
Otsuimage = (Bitmap)image1.Image.Clone();
image2.SizeMode = PictureBoxSizeMode.CenterImage;
//ApplyFilter(true);
}
privatevoid dname_SelectedIndexChanged(object sender, EventArgs e)
{
MySqlDataReader rddata;
string mySQLconn = "datasource=localhost;Database =fingerprint-biometrics;
port=3306;username=root;password=;";
string selectSQL;
selectSQL = " SELECT * from mastertable where fullname= '" + dname.Text + "'";
MySqlConnection con = newMySqlConnection(mySQLconn);
MySqlCommand cmd1 = newMySqlCommand(selectSQL, con);
try
{
con.Open();
rddata = cmd1.ExecuteReader();
// int added = 0;
while (rddata.Read())
{
//usergroup.Enabled = true;

byte[] imgg = (byte[])(rddata["fimage"]);
if (imgg == null)
image1.Image = null;
else
{
MemoryStream mstream = newMemoryStream(imgg);
image1.Image = System.Drawing.Image.FromStream(mstream);
}}rddata.Close();
lblstatus.Text = "";
}
catch (Exception err)
```

```
{
lblstatus.Text = " Error Geting Author. ";
lblstatus.Text = err.Message;
}
finally
{
con.Close();
// photoimage.Image = null;
}}
privateint CountPixels(Bitmap bm, Color target_color)
{
int matches = 0;
for (int y = 0; y < bm.Height; y++)
{
for (int x = 0; x < bm.Width; x++)
{
if (bm.GetPixel(x, y) == target_color) matches++;
}}
return matches;
}
privatevoid nopixel_Click(object sender, EventArgs e)
{
Bitmap bm = newBitmap(image1.Image);
Bitmap bm1 = newBitmap(image2.Image);
int black_pixels = CountPixels(bm, Color.FromArgb(255, 0, 0, 0));
int white_pixels = CountPixels(bm, Color.FromArgb(255, 255, 255, 255));
int black_pixels1 = CountPixels(bm1, Color.FromArgb(255, 0, 0, 0));
int white_pixels1 = CountPixels(bm1, Color.FromArgb(255, 255, 255, 255));
lblBlack.Text = black_pixels + " black pixels";
lblWhite.Text = white_pixels + " white pixels";
lblTotal.Text = white_pixels + black_pixels + " total pixels";
lblBlack1.Text = black_pixels1 + " black pixels";
lblWhite1.Text = white_pixels1 + " white pixels";
lblTotal1.Text = white_pixels1 + black_pixels1 + " total pixels";
}
privatevoid btnSaveNewImage_Click(object sender, EventArgs e)
{
if (resultBitmap != null)
{
SaveFileDialog sfd = newSaveFileDialog();
sfd.Title = "Specify a file name and file path";
sfd.Filter = "Png Images(*.png)|*.png|Jpeg Images(*.jpg)|*.jpg";
sfd.Filter += "|Bitmap Images(*.bmp)|*.bmp";
```

```
if (sfd.ShowDialog() == System.Windows.Forms.DialogResult.OK)
{
string fileExtension = Path.GetExtension(sfd.FileName).ToUpper();
ImageFormat imgFormat = ImageFormat.Png;

if (fileExtension == "BMP")
{
imgFormat = ImageFormat.Bmp;
}
elseif (fileExtension == "JPG")
{
imgFormat = ImageFormat.Jpeg;
}StreamWriter streamWriter = newStreamWriter(sfd.FileName, false);
resultBitmap.Save(streamWriter.BaseStream, imgFormat);
streamWriter.Flush();
streamWriter.Close();
resultBitmap = null;}}
privatevoid znormz_Click(object sender, EventArgs e)
{
Bitmap temp = (Bitmap)Otsuimage.Clone();
ot.Convert2GrayScaleFast(temp);
int otsuThreshold = ot.getOtsuThreshold((Bitmap)temp);
ot.threshold(temp, otsuThreshold);
textBox1.Text = otsuThreshold.ToString();
image2.Image = temp;
}}privatevoid thinning_Click(object sender, EventArgs e)
}{
privatevoid Fingerprint_Analysis_Load(object sender, EventArgs e)
{
Otsuimage = (Bitmap)image2.Image.Clone();
}
privatevoid trcThreshold_Scroll(object sender, EventArgs e)
{
lblContrastValue.Text = trcThreshold.Value.ToString();
//image2.Image = BitmapFilter.Contrast(m_Bitmap, (Bitmap)image2.Image, trcThreshold.Value);
}
publicstaticBitmap AdjustBrightness(Bitmap Image, int Value)
{
Bitmap TempBitmap = Image;
Bitmap NewBitmap = newBitmap(TempBitmap.Width, TempBitmap.Height);
Graphics NewGraphics = Graphics.FromImage(NewBitmap);
float FinalValue = (float)0.001f;
```

```
float[][] FloatColorMatrix ={
newfloat[] {1, 0f, 0f, 0f, 0f},
newfloat[] {0f, 1, 0f, 0f, 0f},
newfloat[] {0f, 0f, 1, 0f, 0f},
newfloat[] {0f, 0f, 0f, 1f, 0f},
newfloat[] {FinalValue, FinalValue, FinalValue ,0f,1f}
};
ColorMatrix NewColorMatrix = newColorMatrix(FloatColorMatrix);
ImageAttributes Attributes = newImageAttributes();
Attributes.SetColorMatrix(NewColorMatrix);
NewGraphics.DrawImage(TempBitmap, newRectangle(0, 0, TempBitmap.Width, TempBitmap.Height), 0, 0,
TempBitmap.Width, TempBitmap.Height, GraphicsUnit.Pixel, Attributes);
Attributes.Dispose();
NewGraphics.Dispose();
return NewBitmap;
}
publicstaticbool Contrast(Bitmap b, sbyte nContrast)
{
if (nContrast < -100) returnfalse;
if (nContrast > 100) returnfalse;
double pixel = 0, contrast = (100.0 + nContrast) / 100.0;
contrast *= contrast;
int red, green, blue;
// GDI+ still lies to us - the return format is BGR, NOT RGB.
BitmapData bmData = b.LockBits(newRectangle(0, 0, b.Width, b.Height), ImageLockMode.ReadWrite,
PixelFormat.Format24bppRgb);
int stride = bmData.Stride;
System.IntPtr Scan0 = bmData.Scan0;
unsafe
{
byte* p = (byte*)(void*)Scan0;
int nOffset = stride - b.Width * 3;
for (int y = 0; y < b.Height; ++y)
{
for (int x = 0; x < b.Width; ++x)
{
blue = p[0];
green = p[1];
red = p[2];
pixel = red / 255.0;
pixel -= 0.5;
pixel *= contrast;
pixel += 0.5;
pixel *= 255;
```

```
if (pixel < 0) pixel = 0;
if (pixel > 255) pixel = 255;
p[2] = (byte)pixel;
pixel = green / 255.0;
pixel -= 0.5;
pixel *= contrast;
pixel += 0.5;
pixel *= 255;
if (pixel < 0) pixel = 0;
if (pixel > 255) pixel = 255;
p[1] = (byte)pixel;
pixel = blue / 255.0;
pixel -= 0.5;
pixel *= contrast;
pixel += 0.5;
pixel *= 255;
if (pixel < 0) pixel = 0;
if (pixel > 255) pixel = 255;
p[0] = (byte)pixel;

p += 3;
}
p += nOffset;
}
}

b.UnlockBits(bmData);

returntrue;
}
```