
A Biometric Approach to Prevent False Use of IDs

By

Rupa Patel

*Submitted to the University of Hertfordshire in partial fulfilment of the requirements of
the degree of MSc by Research*

June 2018

ABSTRACT

What is your username? What is your password? What is your PIN number?

These are some of the commonly used key questions users need to answer accurately in order to verify their identity and gain access to systems and their own data. Passwords, Personal Identification Numbers (PINs) and ID cards are different means of tokens used to identify a person, but these can be forgotten, stolen or lost.

Currently, University of Hertfordshire (UH) carries out identity checks by checking the photograph on an ID card during exams. Other processes such as attendance monitoring and door access control require tapping the ID card on a reader. These methods can cause issues such as unauthorised use of ID card on attendance system and door access system if ID card is found, lost or borrowed. During exams, this could lead to interruptions when carrying out manual checks. As the invigilator carries out checks whilst the student is writing an exam, it is often difficult to see the student's face as they face down whilst writing the exam. They cannot be disturbed for the ID check process. Students are also required to sign a manual register as they walk into the exam room. This process is time consuming.

A more robust approach to identification of individuals that can avoid the above mentioned limitations of the traditional means, is the use of biometrics. Fingerprint was the first biometric modality that has been used. In comparison to other biometric modalities such as signature and face recognition, fingerprint is highly unique, accepted and leads to a more accurate matching result. Considering these properties of fingerprint biometrics, it has been explored in the research study presented in this thesis to enhance the efficiency and the reliability of the University's exam process.

This thesis focuses on using fingerprint recognition technology in a novel approach to check identity for exams in a University environment. Identifying a user using fingerprints is not the only aim of this project. Convenience and user experience play vital roles in this project whilst improving speed and processes at UH.

ACKNOWLEDGEMENTS

There has been a lot of interest in this project and I would like to thank all my colleagues, friends and family for their encouragement, faith and good wishes. Huge thanks to:

Research Supervisors, UH

Lily Meng, Iosof Mporas and Sooda Ramalingam

Sponsors, UH

Academic Registry

Additional Funding

Diamond Fund (UH), Postgraduate Conference Bid (UH), Proof of Concept (UH), and Cost ACTION IC1206 Gran Canaria

Exams and Awards Team, UH

Anne Austin, Sarah Beeley and Mick Fowler

Suppliers

Ian Allen, Keri Feeney, Charlie Hicks, Matthew Insley, Kevin Loveman and Daniel Smith

External Organisations

Beaumont School – Julie Wells, Biometrics Institute, InnoEducaTIC 2017, University of Bristol – Tony Blundall, University of South Wales – Liam Bryson and William Callaway

Internal Business Units at UH

Academic Registry – Francesca Coxon, Sharon Harrison-Barker, Liz Hedges, Helena Johnson, Julie Kelly, Tracy Payen -Taylor, Emma Pritchard, Clare Sapsford, Andy Taylor, Susan Warner and Eileen Worby

School of Computer Science – Ian Bradford, William Clocksin, Jo Horridge, Ruth Marsh, Katy Sykes and Simon Trainis

School of Engineering and Technology – Rodney Day, Funlade Sunmola, Anne Passmore, Sarah Flynn and Pandelis Kourtessis

Estates, Hospitality and Contract Services – Alan Reck (Lintel) and Security

School of Humanities – Theo Gilbert and Jeremy Ridgman

School of Law – Katie Barker, Carolyn Back, Penny Carey and Charles Wild

Library and Computing Services – Matt Billows, Daniel Bu, Chris Hill, Tony Crook, David Gee, Colin Manning, Nathan Ruddick and Andy Wroot.

Marketing and Communications – Sarah Koniotis and Jerome Price.

Office of Vice Chancellor – Alex Hall, Ian Hanahoe and Sal Jarvis.

Student Participants – School of Computer Science, Engineering and Technology and Law

Students Union – Phil MacKay and Grainne Meadhbh O'Monghain

Proofreaders

Jyotika Halai and Lily Meng

TABLE OF CONTENTS

ABSTRACT	i
ACKNOWLEDGEMENTS	ii
List of Figures	v
List of Tables.....	vii
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: EXAM REGISTRATION PROCESS.....	4
2.1 Current Process	4
2.2 Benefits of using biometrics	4
CHAPTER 3: SUBJECT REVIEW.....	7
3.1 Why Fingerprints?	7
3.2 Comparison of different biometric modalities.....	7
3.2.1 Iris Recognition	8
3.2.2 Face Recognition	10
3.2.3 Fingerprint Recognition	11
3.3 Subject Review of Fingerprint Recognition Technology.....	13
3.3.1 Image processing	17
3.3.2 Feature extraction	21
3.3.3 Matching process	28
CHAPTER 4: PRACTICAL WORK	34
4.1 Survey	34
4.1.1 Methodology.....	34
4.1.2 Survey Results.....	36
4.2 System Architecture and Processes.....	39
4.3 Databases.....	40
4.3.1 The Exams Database	42
4.3.2 Fingerprint Application Database.....	43
4.3.3 The ID Card Database.....	44
4.3.4 Database Queries	45
4.4 Fingerprint Application Software Development Kit/Programming.....	51
4.4.1 Fingerprint Application Design	52
4.4.2 Fingerprint Application Programming.....	54
4.5 Development of the Integrated Exam Reporting System	59
4.5.1 Design and Functionality	59
4.5.2 Programming.....	64
CHAPTER 5: SYSTEM TESTING	70

5.1 System Deployment via Proof of Concept.....	70
5.2 Enrolment.....	71
5.2.1 Enrolment with the developed system	71
5.2.2 Enrolment with the commercial system	73
5.2.3 Collectability and universality of fingerprints of University student population	76
5.3 Authentication.....	78
5.3.1 Test Results – Performance	80
5.4 User Feedback – Acceptability	84
CHAPTER 6: RECOMMENDATION AND CONCLUSION	88
REFERENCES	91
APPENDIX I: SAMPLE FORM TO RECORD ABSENT AND EXTRA STUDENTS	94
APPENDIX II: UH SCREENS BROADCAST REQUEST	95
APPENDIX III: FINGERPRINT SURVEY METHODS	96
APPENDIX IV: ENTITY RELATIONSHIP DIAGRAM (ERD) V1	98
APPENDIX V: SQL STATEMENTS	99
APPENDIX VI: SAMPLE SDK FINGERPRINT APPLICATIONS	101
APPENDIX VII: FINGERPRINT APPLICATION RECORDLIST.CS CODE.....	104
APPENDIX VIII: FINGERPRINT APPLICATION MSOACQUISITIONSTRATEGY.CS CODE 108	
APPENDIX IX: EXAMS REPORT SYSTEM: CODE FOR LOGINFORM.CS.....	117
APPENDIX X: EXAMS REPORT SYSTEM – CODE FOR SEARCHFORM.CS.....	119
APPENDIX XI: EXAMS REPORT SYSTEM – CODE FOR EXAMSOFFICEREPORTS.CS 121	
APPENDIX XII: EXAMS REPORT SYSTEM – CODE FOR EXAMINVIGILATOR.CS	126
APPENDIX XIII: PROJECT PLAN.....	128
APPENDIX XIV: CONSENT FORM FOR PROOF OF CONCEPT	129
APPENDIX XV: PARTICIPANT INFORMATION SHEET	130
APPENDIX XVI: CONSENT FORM FOR RESEARCH PROJECT	132
APPENDIX XVII: SYSTEM FEEDBACK FORM.....	133

List of Figures

Figure 1 Redundant Access Control Reader at UH	2
Figure 2 Exam Register.....	4
Figure 3 Different Biometric Modalities (taken from [4])	8
Figure 4 Iris (taken from [5])	8
Figure 5 Variations of Irises (taken from [6])	9
Figure 6 Face Recognition Stages (taken from [8])	10
Figure 7 Types of Fingerprint Minutiae (taken from [9])	11
Figure 8 Biometrics Market Shares 2015 by Modality (taken from [11]).....	13
Figure 9 Types of fingerprint patterns (taken from [13]), From left to right: right loop, left loop, whorl, arch, tented arch.....	13
Figure 10 Milestones of Fingerprint Recognition (taken from [18])	14
Figure 11 Evolution of Fingerprint Recognition Sensors from 1800s to 2010 (taken from [18])	14
Figure 12 Fingerprint Scanner Optical Sensor (taken from [20])	15
Figure 13 Fingerprint Scanner Capacitive Sensor (taken from [20])	15
Figure 14 Slap Fingerprint Technology (taken from [21])	16
Figure 15 Near Infrared Technology for Slap Fingerprint Scanners (taken from [22])	16
Figure 16 Touchless Fingerprint Sensor (taken from [24])	17
Figure 17 Fingerprint Recognition System Block Diagram (taken from [11]).....	17
Figure 18 Histogram Equalization Before and After Fingerprint Images (taken from [26])....	18
Figure 19 Before and After Histograms Post Histogram Equalization (taken from [9])	19
Figure 20 Image Enhanced Using Fourier Transform Method (taken from [28])	20
Figure 21 Thinned Fingerprint Image (taken from [28])	21
Figure 22 Types of Minutiae (taken from [34])	22
Figure 23 Binarized and Gray-scale Images (a) Gray-scale Image, (b) Binarized Image, (c) Image obtained after thinning process of binarized image (taken from [38])	23
Figure 24 Fingerprint Feature Extraction Algorithm (images taken from [13]).....	23
Figure 25 (a) Singular points and minutiae with its direction and (b) orientation field shown with unit vector (taken from [40]).....	24
Figure 26 Crossing Number (CN) value for Bifurcation and Ridge Ending (taken from [43])	24
Figure 27 Key Feature Extraction. (a) Feature levels in a fingerprint. (b) Flow chart of a typical minutiae feature (taken from [44])	25
Figure 28 Classification of Key Features Extraction Techniques (taken from [30])	26
Figure 29 A 3x3 pixel window (taken from [32])	27
Figure 30 Fingerprint Matching Techniques	29
Figure 31 Match and Non-match Distributions and Receiver Operating Curve (ROC) (taken from [15])	32
Figure 32 Survey Link on StudyNet.....	35
Figure 33 Survey Promoted on UH Screens.....	35
Figure 34 Fingerprint Recognition System.....	40
Figure 35 Entity Relationship Diagram v2	41
Figure 36 System Databases	42
Figure 37 Original Exams Database	42
Figure 38 Fingerprint Application Database.....	43
Figure 39 ID Card Database.....	44

Figure 40 Sub Query to Retrieve Student ID Numbers of Students Taking Scheduled Exam	45
Figure 41 Query to Retrieve List of Students Taking Exam	46
Figure 42 Query Results Showing Student ID Numbers and Corresponding Exam Room Location	46
Figure 43 Query Results showing Device ID of Exam Room.....	47
Figure 44 Query Results showing Student Numbers of Students who have Taken Exam	47
Figure 45 Query Results showing Absentee Students.....	48
Figure 46 Query Results for Extra Students	48
Figure 47 Query Results - Student Numbers and Names of Students Scheduled for an Exam	49
Figure 48 Query Results Showing List of Student Taken Exam in a Room.....	50
Figure 49 Query Results for List of Absentees for Exams Office	50
Figure 50 Query Results for List of Extra Students for Exams Office	51
Figure 51 Morpho License and USB Scanner Service.....	52
Figure 52 Fingerprint Application Main Form Amended.....	52
Figure 53 Fingerprint Application Amended New Record Form	53
Figure 54 Fingerprint Application Amended Record List Form	53
Figure 55 Fingerprint Application Identification Record without Personal Details.....	58
Figure 56 Fingerprint Application Identification Record with Personal Details	59
Figure 57 System Navigation Diagram	60
Figure 58 Login Form	60
Figure 59 Search Form	61
Figure 60 Instructions to user for fingerprint capture	61
Figure 61 Absentee Students Report for Exams Office Report	62
Figure 62 Extra Students for Exams Office Report.....	62
Figure 63 Absentee Students for Marker Report	63
Figure 64 Extra Students for Marker Report.....	63
Figure 65 Export Exams Office Reports	64
Figure 66 Form for Exam Invigilator	64
Figure 67 Fingerprint Scanner and License Dongle	72
Figure 68 Enrolment using Commercial System	73
Figure 69 Fingerprint Wall Reader	79
Figure 70 Commercial System Reports	80
Figure 71 Feedback of the Developed System from Test 1	85
Figure 72 Feedback of the Developed System from Test 2	86
Figure 73 Feedback from System Users of Commercial System	87

List of Tables

Table 1 Ratings of Desirable Properties of Different Biometric Modalities (taken from [1] [3])	7
Table 2 List of Fingerprint Quality Assessment Algorithms (taken from [29]).....	20
Table 3 Characteristics of Minutiae	21
Table 4 Crossing Number Point System	27
Table 5 Pseudocode for Determining Direction from CN.....	27
Table 6 Comments from Survey Respondents.....	38
Table 7 Key Information from Tests using Commercial System.....	75
Table 8 Live Quality Scores using Developed System.....	77
Table 9 Live Quality Scores using Commercial System	78
Table 10 Matching Scores from Tests of Commercial System	81
Table 11 Matching Scores from Test 1 of Developed System Before Enhancements	81
Table 12 Matching Scores from Test 2 of Developed System after Enhancements	82
Table 13 Matching Scores of Imposter from Developed System	83

CHAPTER 1: INTRODUCTION

Using traditional methods such as passwords, PINs and ID cards to gain access into systems and verify your identity can pose risks of identity fraud/theft if these are lost or stolen. In addition to this, if tokens are forgotten, this can cause inconvenience to users. Passwords and PINs can be recovered by answering some security questions, but ID cards have to be replaced.

The University of Hertfordshire (UH) currently uses Mifare chip on its ID cards which is compatible with most systems such as student check-in system, door access control system and Multi Functional Devices (MFDs).

There are known issues with ID cards as these can be passed around or stolen which can cause issues such as providing false information. For example, when a student swipes another student's ID card to register attendance or misuses an ID card i.e. in the event of an unauthorised user finding another user's ID card and using it to gain access to buildings/rooms/systems.

Biometrics cannot only address the above issues, but also speed up processes if implemented correctly. One such process where the use of biometrics can be beneficial is identifying students during exams. This research project aims to utilise fingerprint recognition techniques to assist exam registration. The choice of fingerprint rather than other forms of biometric modalities is based on its high uniqueness, high acceptability and most importantly its high accuracy [1]. Although fingerprint has medium universality among general public, when the target users are constrained within the University student population, a high universality has been observed in this research work. Extra measures have been suggested in this thesis to improve collectability.

This thesis focuses on the outcomes of the research project entitled "A Biometric Approach to Prevent False Use of IDs" which revolves around the research question: Can biometrics be the alternative and preferred method to prove identity?

In a report published by Big Brother Watch in 2014, it is estimated that 40% of schools in England are using biometric technology and more than 866,000 children had their fingerprint taken [2]. Schools mainly use fingerprint technology for cashless catering to ensure pupils can still have their lunch if they forget their ID card but more importantly not having to carry cash with them.

There are very few Universities that use biometric technology. Two Universities were identified before this project commenced. After identifying that University of South Wales and University of Bristol use fingerprint technology, a site visit was arranged in March 2016 to view their respective working systems. University of South Wales uses fingerprint technology for attendance monitoring of its Tier 4 visa holder students and University of Bristol uses it for door access control. Whilst University of Bristol saves the fingerprint template on ID cards to follow a verification process by carrying out one-to-one matching, University of South Wales stores the templates in a secure database to carry out one-to-many identification. University of South Wales was the first University to implement a fingerprint technology in the UK.

University of Hertfordshire's Sports Village did introduce a fingerprint system for its door access control system but due to hardware issues, this became redundant. The reader

remains on the wall by the door entrance (as shown in Figure 1) as they hope to reintroduce the fingerprint technology for identification instead of using ID cards.



Figure 1 Redundant Access Control Reader at UH

This research project has generated a lot of interest within UH, amongst suppliers and other Universities. UH will be the first University in the UK to implement fingerprint technology for Exams process if successful. Part of this research work has also been highlighted in a book Chapter, "Advances in Fingerprint Technology" written by the author of this Thesis. This will be published by Springer in a book entitled "Biometric-Based Physical and Cybersecurity Systems".

Due to privacy concerns especially with the General Data Protection Regulations (GDPR) enforced on 25th May 2018, Universities are still hesitant to invest in biometric technology.

There are forums where industry experts meet, and best practice guides are shared. One such forum is the Biometrics Institute: <https://www.biometricsinstitute.org/>. Following funding received from the UH Postgraduate Research (PGR) Conference Funding Bid, the author and three other UH Staff became members of the Biometrics Institute.

Biometrics Institute provides an independent and impartial international platform for biometric users and other interested parties to promote the responsible use of biometrics. Some of its members include UK Home Office, Barclays Bank and Mastercard. UH has benefited from becoming a member as we have received updates on GDPR, best practice guides and news as and when they happen. Biometrics Institute also hosts various expert groups including one in security and privacy called the Privacy Expert Group.

This research work has closely followed the development of data protection regulations and the best practice. For example, the University ethics process has been followed, consent was always obtained from students, encryption of data and fingerprint templates, data

storage on secured servers and a proposal for the deletion process have also been put in place.

A novel approach of using a fingerprint recognition system instead of ID cards in exams processes has been demonstrated in this research project by involving key stakeholders i.e. students, staff and suppliers. A fingerprint application has also been enhanced to link to the University's exams database and ID database and tested during exam sessions.

The rest of this thesis is structured as follows:

- Chapter 2 details the existing UH exam registration process.
- Chapter 3 discusses various biometric modalities and justifies why fingerprint recognition is the chosen modality for this project.
- Chapter 4 presents the work carried out over the past two years in order to produce two working systems (the Fingerprint Application and the Exams Reporting system).
- Chapter 5 highlights the success of Proof of Concept deployment and sheds light on:
 - The fingerprint enrolment and exam registration processes carried out at UH.
 - Full test results obtained from various tests carried out throughout the project.
- Chapter 6 provides recommendations for deployment of the developed systems and processes, draws conclusions from the project outcomes whilst reflecting on objectives met or changed.

CHAPTER 2: EXAM REGISTRATION PROCESS

The current ID card based identity verification process used during exams is lengthy and time consuming. This section details the current process and benefits of using biometrics.

2.1 Current Process

Currently, the exam registration process involves manual checks of the IDs where the invigilator has to check the photograph on an ID card to see if it matches the face of the cardholder presenting the ID card.

When a student enters the room where their exam takes place, they are required to sign a manual register as shown in Figure 2. All personal data i.e. ID numbers, signatures and email addresses have been redacted in all images used in this thesis to protect identity.

The image shows a pink 'University of Hertfordshire - Examination Register' form. The form includes the following information:

- Day: Friday
- Room: C152 - College Lane
- Total Candidates for exam: 18
- Date: 15 Jan 16
- Time: 14:15
- Module Code & Title: 6NMH1031-0909 Transitions to Professional Practice
- Date collected for marking: 17/1/16
- Nos of scripts collected by invigilator: 14
- Marker's signature: [Redacted]
- Please print name: [Redacted]
- Seat Name & Student No.: 1 [Redacted] 2 [Redacted]
- Signature: [Redacted]

Figure 2 Exam Register

The ID check results are firstly recorded on paper by the invigilator. When additional candidates are verified, their details have to be recorded on another paper form as well, as shown in Appendix I. This paper-based process is time consuming and not always reliable.

Training of invigilators and exam office staff as well as post-exam processing of the forms all add to the costs of running an exam. In addition, human error can cost not only time and money, it can also have a huge impact on the individual being affected.

When a student forgets to bring his/her ID to the exam, currently they are required to visit an administration office to get a temporary ID slip. Replacing lost ID cards or getting a temporary one can also be time consuming especially when the exams are due to start within minutes of a student realising they have forgotten/lost his/her ID card.

In a crucial process such as exams, the absence of an ID card can have a huge impact on students leading to unnecessary stress for both students and staff, as well as invigilators. This is why the use of biometrics can be beneficial for the exams process.

2.2 Benefits of using biometrics

The ID Office system already captures photographs when a student registers on their course. If UH introduces biometrics, fingerprints can be captured at the same time which can be linked to the student's record. For any student joining the University at any other time, this

can be captured by the ID Office team at the time of issuing an ID card. This means the enrolment of fingerprints at UH would not cause significant inconvenience to the students and the introduction cost would be low.

This data can then be used for various processes by different teams within UH, e.g. the exam registration process run by the exam office through the invigilator(s).

If a student forgets his/her ID card at an exam, instead of using a temporary paper ID slip, UH can use fingerprint recognition readers attached to a mobile device, PC or wall to identify the student against the student's record and photographs stored in the ID Office database. To reduce the number of matching required per student and hence achieve high efficiency required by the exam registration process. The fingerprint system developed in this research work has achieved a significant reduction of population size by matching against students in the exam class list only, as opposed to carrying out an identification against all student population within our database. The current student population at UH is over 27000 while the maximum number of students at a particular exam room is 200.

Biometrics can be used for various other processes by different teams within UH:

- **Door Entry System:**
Use of biometrics system will allow members of the University to gain access into authorised areas. This will tighten security across UH as it will eliminate unauthorised access into restricted areas by means of lending ID card to someone else. In the case of lost ID card, it can be used by anyone who may find it if not reported lost to the ID Office.
- **Library System:**
Library and Computing Services department can move away from scanning barcodes on ID cards for issuing and returning of books. Use of fingerprint recognition reader would stop members of the University using someone else's ID card if found.
- **Security:**
Use of fingerprint or face recognition at the entry point of libraries would help security officers to identify individuals who have left UH or those who have been banned. These individuals can then be denied access and appropriate action can be taken as required.

Face recognition technology can also be used to match facial features from ID photo database with CCTV images to quickly identify details of any UH member who would have been caught carrying out an offence on UH premises.

- **Student Attendance Monitoring:**
ID cards can get lost, stolen or misused. A student can give an ID card to another student to swipe on attendance system to mark their attendance. With integration of biometrics to the attendance system, students will have no means to bypass the current system hence provide real and more accurate data.
- **Students' Union:**

If the ID Office system captures fingerprint data, this can be linked to the Students' Union database which would allow SU security staff to give entry to UH events to authorised students only. Face recognition technology could be an alternative for this process.

- Award Ceremonies and other similar events at the University:
The fingerprint technology can be used by students at their award ceremony to collect tickets for their guests, so they don't have to wait in a queue to show a form of identity.

While the use of biometrics will bring benefits to various processes at UH, the research work presented in this thesis, as a proof of concept, has focused on the use of fingerprint technologies during exam registration.

CHAPTER 3: SUBJECT REVIEW

At the initial stage of this research study, investigation was carried out into different biometric modalities, which has led to the conclusion that fingerprint recognition system would be best suited for this project. This chapter details why fingerprint recognition technology was the chosen modality in comparison to different modalities followed by a detailed subject review of fingerprint recognition technology.

3.1 Why Fingerprints?

Fingerprints have been associated with crime investigations since the 1800s, which has had a negative impact on the acceptability of fingerprint recognition. But the integration of fingerprint scanners in mobile devices over the last five years is changing the perception of users and making identity authentication processes easy and convenient, i.e. providing good user experience whilst still meeting security requirements.

As shown in Table 1, fingerprint technology is known to have a very good balance of desirable properties: Distinctiveness, Performance, Circumvention, Universality, Acceptability, Permanence and Collectability [1]. Fingerprints are highly unique; no two individuals have the same fingerprints – even twins! This has led to the high accuracy of fingerprint recognition. High quality images can be captured very quickly which makes fingerprint a popular choice as time is essential in many processes such as exam registration. Everyone has fingerprints making them universal. With technology evolving and the incorporation of fingerprint scanners in mobile devices, the capture and subsequent use of fingerprint for authentication are being more accepted now than many years ago.

Facial features change over time but fingerprints don't change, hence it scores highly in the permanence category. With fingerprint scanners becoming more affordable, fingerprints can be easily collected making it one of the most desirable modalities.

Table 1 Ratings of Desirable Properties of Different Biometric Modalities (taken from [1] [3])

Biometric Identifier	Universality	Distinctiveness	Permanence	Collectability	Performance	Acceptability	Circumvention	Introduction Cost
DNA	H	H	H	L	H	L	L	H
Face	H	L	M	H	L	H	H	M
Fingerprint	M	H	H	M	H	M	M	L
Hand geometry	M	M	M	H	M	M	M	L
Hand vein	M	M	M	M	M	M	L	M
Iris	H	H	H	M	H	L	L	H
Retina	H	H	M	L	H	L	L	H
Signature	L	L	L	H	L	H	H	L
Voice	M	L	L	M	L	H	H	M

Key: H-High, M-Medium, L-Low

3.2 Comparison of different biometric modalities

There are various biometric modalities as shown in Figure 3. This subsection compares the potential three modality candidates for this project:

- Iris recognition is considered as it rates the highest in almost all desirable properties.
- Face recognition as it is widely used and was the first thought of modality for this research project.
- Fingerprint recognition as it is the chosen modality for this project.

Other modalities are not considered due to various factors such as: low acceptance and collectability of DNA, low performance of signature recognition and low acceptability and collectability of retina.

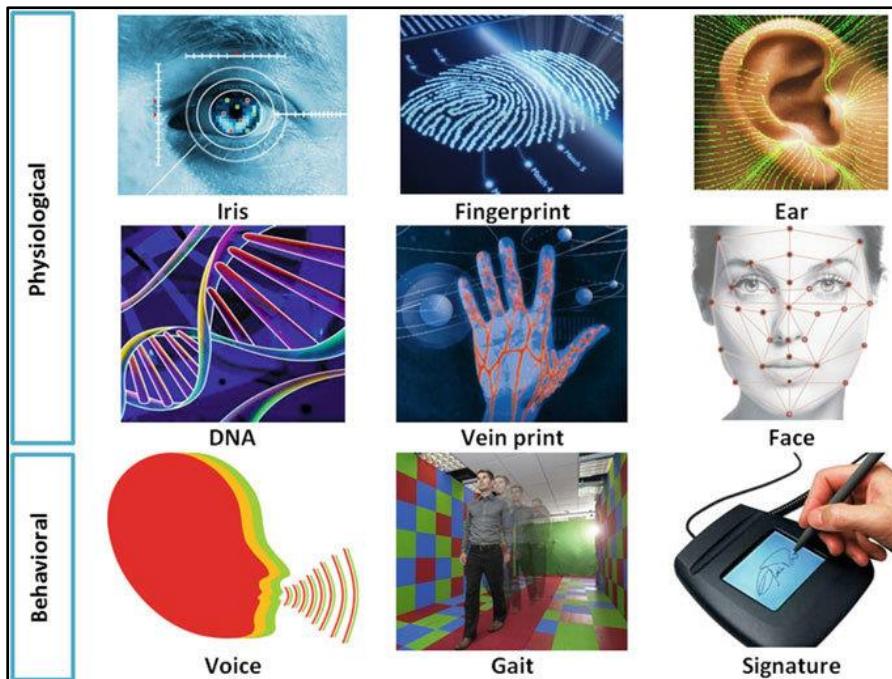


Figure 3 Different Biometric Modalities (taken from [4])

As shown in Figure 3, there are physiological modalities which rely on a person's physical features i.e. iris, face or fingers and there are behavioural modalities such as voice, gait and signature which rely on a person's actions.

3.2.1 Iris Recognition

Iris recognition analyses random patterns of the iris (Figure 4). In 1985, it was argued by Dr Leonard Flom and Dr Aran Safir that no two irises are alike, even in twins. The first commercial products using iris recognition became available in 1995 [5].

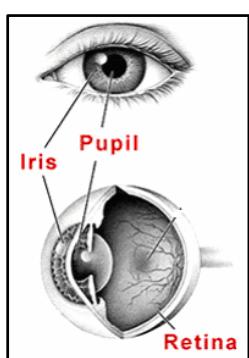


Figure 4 Iris (taken from [5])

Irides can have distinct features such as the “many collagenous fibres, contraction furrows, coronas, crypts, colour, serpentine vasculature, striations, freckles, rifts and pits” but “the primary visible characteristic is the trabecular meshwork which gives the appearance of dividing the iris in a radial fashion which is permanently formed by the eighth month of gestation” [6].

An iris has very low probability of getting damaged as it is protected by the eyelid unlike fingers which are prone to cuts and scars. An iris is not subject to aging either and the use of glasses/contact lenses has very little effect on its representation which results to low impact to its accuracy unlike face recognition which is prone to occlusions. Figure 5 shows different variations of irises.

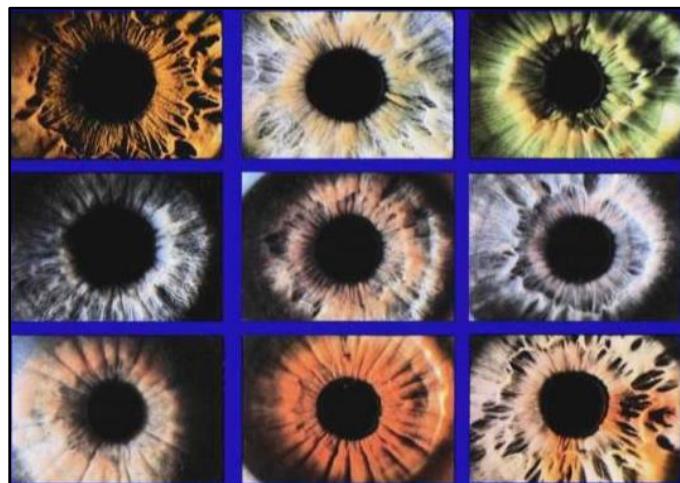


Figure 5 Variations of Irises (taken from [6])

Images of irises can be captured using high quality digital cameras with the commercial ones using infrared light to brighten the iris without causing discomfort or harm to the user [5]. An iris template is obtained by first capturing the image, then segmenting and localizing it, normalizing it, encoding the features and finally storing it and comparing the template against stored ones in the database.

Iris recognition has many advantages. Some of these include [7]:

- High accuracy – Iris is known to be the most accurate modality.
- High stability and permanence – As an iris is protected by an eyelid and cornea, the pattern remains stable throughout an individual’s life. An iris doesn’t change hence rating high in permanence.
- Universality – Everyone has an iris hence it is universally available.

Although iris recognition is rated the most accurate and hence one of the most widely used modalities, it does have a few disadvantages [7]:

- Medium collectability – Irises can be scanned from a normal distance like taking a regular photo.
- Expensive – Iris scanners costs five times more than fingerprint scanners making it less affordable for smaller organisations or members of the public.
- Reflection – In rare cases where there is presence of obscurity or reflections i.e. eyelashes or lenses, it can be difficult to scan the iris.

- Transformation – Medical or other conditions may change the size of the pupil causing the iris to deform non-elaistically.
- Infrared light – Repeated use of the iris system may cause harm to iris due to exposure to infrared light.

As a user of a Samsung mobile phone which allows the phone to be unlocked using iris recognition, face recognition or fingerprint recognition, the disclaimer for iris recognition is a major put-off to me as it displays five precautions to keep in mind i.e. "Anyone who experiences dizziness, seizures, loss of awareness, blackouts, or other symptoms linked to an epileptic condition, or a family history of such symptoms or conditions, should see a doctor before using iris recognition". Another precaution which does not demonstrate confidence as a user is: "Do not use iris recognition with infants. Doing so may damage their eyesight". This was a huge contributing factor to iris recognition not being the chosen modality for this project. This is also reflected by its low acceptability with the general public.

3.2.2 Face Recognition

Face recognition technology analyses shape, pattern and location of facial features. For example, shape of cheekbones, length of jawline, distance between the eyes and nose width, and textures in these local areas. These are used to create a template/faceprint which is then compared against the templates of faces registered within a database [8].

As shown in Figure 6, the four stages of face recognition are:

- (i) Image capture using any conventional camera.
- (ii) Extracting key features to create a template.
- (iii) Comparing the data with existing templates.
- (iv) Provide a decision on the identity of the captured face image.

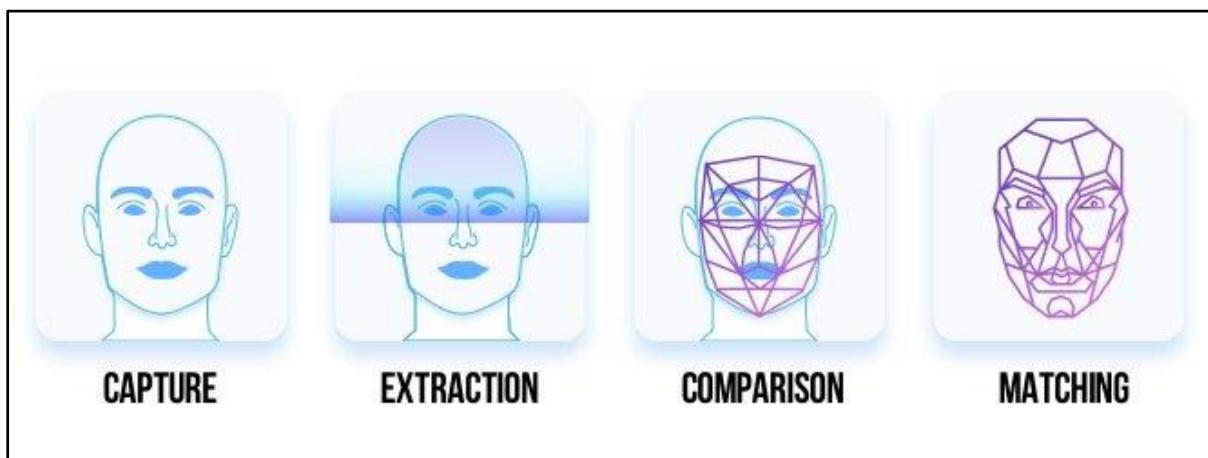


Figure 6 Face Recognition Stages (taken from [8])

Face recognition has the following advantages and disadvantages [3].

Advantages:

- Non-intrusive – As the process involves capturing an image at a distance, it is not intrusive for users, hence rates high in collectability. It is fast and can be parallel as it does not require user's cooperation.
- Low implementation costs – Conventional cameras are cheap in comparison to the devices used for other modalities.

- High acceptability – Thanks to social media, photographs are more widely shared. As a result, users are more willing to use face recognition technology compared to fingerprint recognition.

Disadvantages:

- Low accuracy – Faces are known to have low uniqueness. Twins have nearly identical faces. Siblings and family members have similar faces. In addition, many factors can change or cause significant changes in facial features (e.g. expression, head pose, makeup, illumination). All these have a noticeable impact on the matching results. Recognition of faces in the wild has always been a challenge.
- Medium permanance – Faces do change over time, especially with children. This means that face images stored in database would have to be updated every few years.

The medium permanance of face is not a concern to this research project as the target users are the university students whose faces are not expected to change significantly over their stay at the University. Face recognition would have been the perfect solution for this project if it had a higher performance rate. As mentioned previously, the exam registration process needs to be quick and accurate. Having used face recognition system at airports which has failed to recognise facial features against template stored on passports, implementing such technology for a crucial process such as exams would be risky.

3.2.3 Fingerprint Recognition

A detailed subject review is carried out in the next subsection, while this subsection summarises the key features of fingerprint recognition technology and lists its advantages and disadvantages.

Fingerprints are captured using fingerprint scanners which can cost around £700 for an antispoof/live detection, fingervein reader. The prints captured are pre-processed, then go through thinning process. Once the image is processed, key features called minutiae as shown in Figure 7 are extracted. Some of the key features include island, delta and pore but the most commonly used ones are ridge endings and bifurcations.

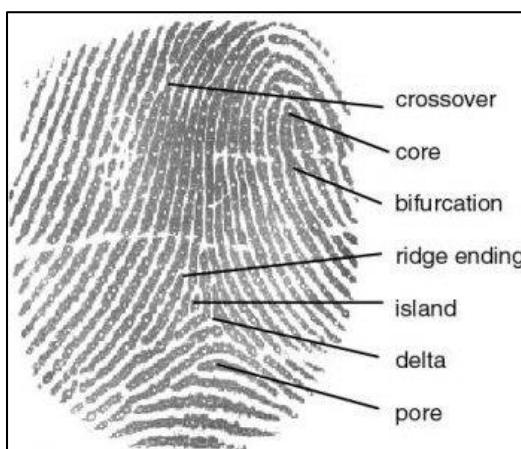


Figure 7 Types of Fingerprint Minutiae (taken from [9])

Once these key features are extracted, the data collected is compared with stored data to carry out the matching process. The matching process is discussed in the next subsection.

Like other biometric modalities, fingerprint recognition also has its own advantages and disadvantages [3].

Advantages:

- High accuracy – Fingerprint technology has always had accurate results hence rating high in accuracy as mentioned previously.
- Easy to use – Unlike retina scan, fingerprints scanners are easy to use.
- Small storage space required – Compared to photographs used for face recognition, biometric templates are relatively small requiring less memory space on a server.
- Low costs – Fingerprint scanners are cheaper compared to scanners for other modalities such as iris scanners.
- High permanence – Various researchers share the same understanding that a fingerprint is a pattern consisting of ridges and valleys. Muzhir Shaban Al-Ani makes a useful point that these are formed during the third to fourth month of fetal development [10] and does not change over one's lifetime.

Disadvantages:

- Intrusive – Unfortunately, some people still associate it with criminal investigation and find fingerprint technology intrusive.
- Hygiene – This is the most talked about issue when fingerprint is mentioned. Even during survey and pilot study for this project, students have mentioned this as a concern.
- Medium collectability – Due to dirt on scanners, moisture and cuts on fingers, this could result to issues collecting fingerprints and also requires user's cooperation.

Fingerprint recognition technology is considered one of the most popular modalities which does not score low on any of the desirable properties. Some examples of fingerprint technology being used include: door locks, laptop login and phone login. With low implementation costs and high accuracy score, this was the most favourable and well-suited modality for this project.

To conclude this section, Figure 8 shows that fingerprint recognition modality is the most adopted technology in terms of market shares, based on revenue [11]. This report was published by ABI Research in 2015.

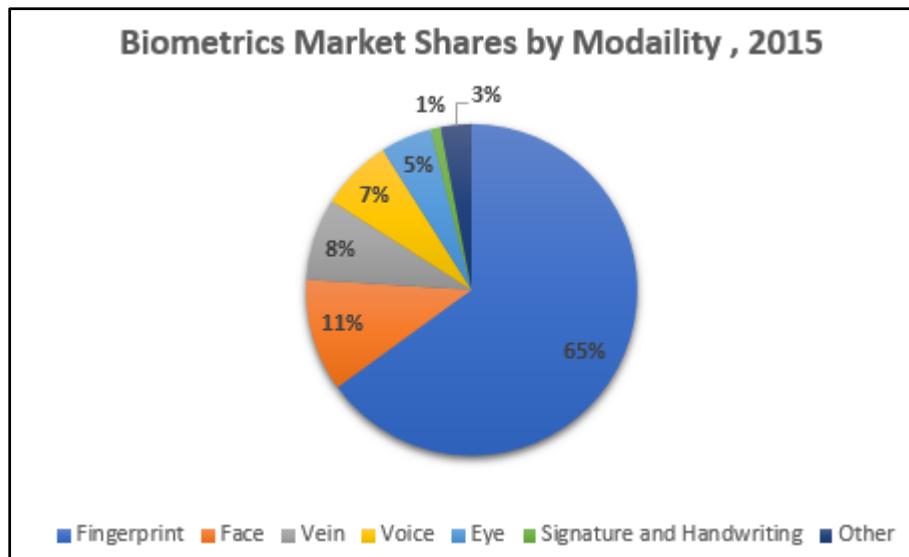


Figure 8 Biometrics Market Shares 2015 by Modality (taken from [11])

3.3 Subject Review of Fingerprint Recognition Technology

This subsection presents a detailed subject review of fingerprint recognition technology, especially focusing on: image processing, key feature extraction and matching process.

As stated by Uchida, “A *fingerprint* is a pattern of fine ridges and valleys (spaces between ridges) on the surface of a finger, and a *fingerprint sensor* makes a digitized image of it.” [12].

The patterns on a fingerprint can be identified by recognising the type of class it belongs to: Whorl, Left Loop, Right Loop, Arch and Tented Arch [13] [14] [15] as shown in Figure 9. This classification system is called Henry classification system as it was introduced by Sir Edward Henry, a British Policeman in 1897 [13].

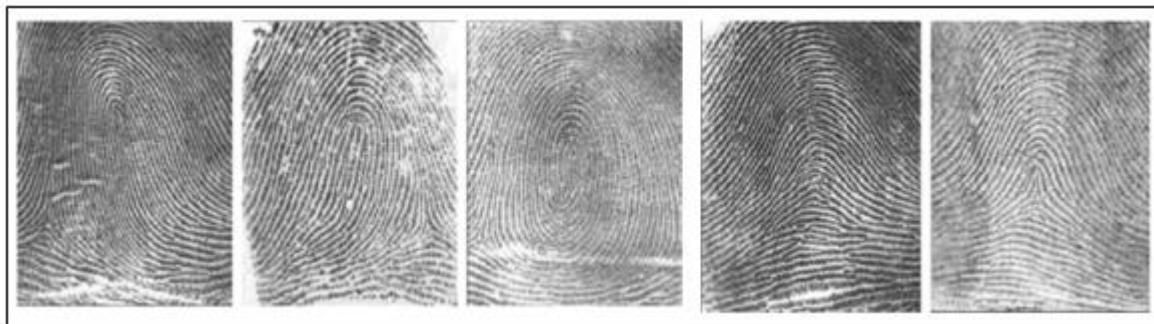


Figure 9 Types of fingerprint patterns (taken from [13]), From left to right: right loop, left loop, whorl, arch, tented arch

Each of these classes have unique characteristics [16] [17]:

- Arch – Ridges enter on one side, form an arch in the centre and exit on the other side. This class consists of Tented Arch and Plain Arch.
- Loop – Ridges enter, form a curve and exit on the same side. This class consists of Left Loop and Right Loop.
- Whorl – Consists of circles, more than one loop or a mixture of pattern types forming: Plain Whorl, Double Loop Whorl, Central Pocket Whorl and Accidental Whorl.

Approximately 65 percent of the population has loops, 30 percent have whorls and 5 percent have arches [10]. It is not clear from the finding which population this refers to though.

Figure 10 shows how the timeline of sensors compare with the milestones of fingerprint recognition [18], from the introduction of Sir Henry's fingerprint classification in 1800s to Scotland Yard adopting fingerprint technology in 1901 and Federal Bureau of Investigation (FBI) initiating the Automated Fingerprint Identification System (AFIS) in 1970s, these systems all use a process of identifying one or many unknown fingerprints against a database.

Two decades later, the swipe sensors were introduced with touchless sensors being invented in 2005. In 2009, the Unique Identification Authority of India project commenced and its prime purpose was to provide a Unique Identity (Aadhaar) to all Indian residents.

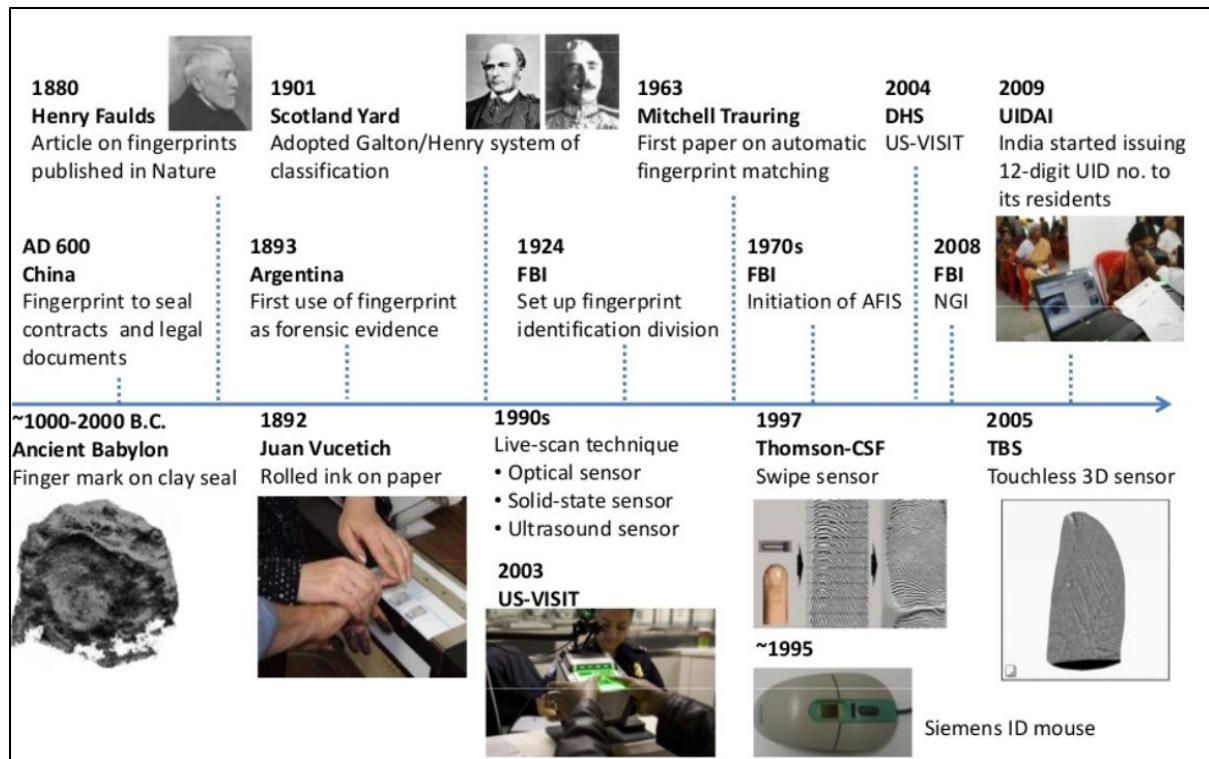


Figure 10 Milestones of Fingerprint Recognition (taken from [18])

To enhance the performance of the system and later to facilitate a more user-friendly experience, manufacturers have revolutionised fingerprint scanners from traditional ink and paper method used in 1800s to introduction of optical sensors in 1990s to the most recent touchless swipe sensor as shown in Figure 11.



Figure 11 Evolution of Fingerprint Recognition Sensors from 1800s to 2010 (taken from [18])

Qiu describes optical sensors as “*the oldest ‘live-scan’ fingerprint sensors that use frustrated refraction over a glass prism (when the skin touches the glass, the light is not reflected but absorbed)*” [19]. The optical scanner reads the ridges and valleys from the fingerprint with the help of light that shines through from the side as shown in Figure 12. When this research project commenced, one of this was purchased to gain an understanding of how fingerprint scanners work and integrate with software.

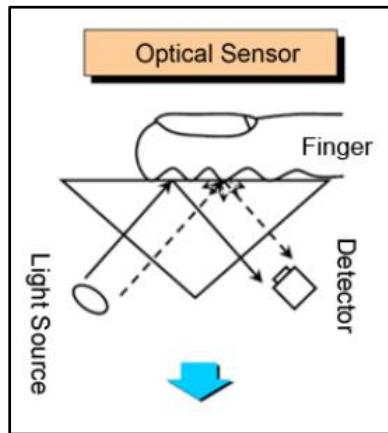


Figure 12 Fingerprint Scanner Optical Sensor (taken from [20])

These are low in cost but big in size as it needs to accommodate the light. It also has high power consumption rate. To address these issues, a capacitive sensor was developed. These are usually used in smartphones. It transfers electricity from a capacitor (a two-terminal electrical component that stores potential energy) and the skin via a short distance as shown in Figure 13. These are small and have low power consumption rate. These cannot be fooled with print of a piece of paper but they do not work with dirty or wet fingers.

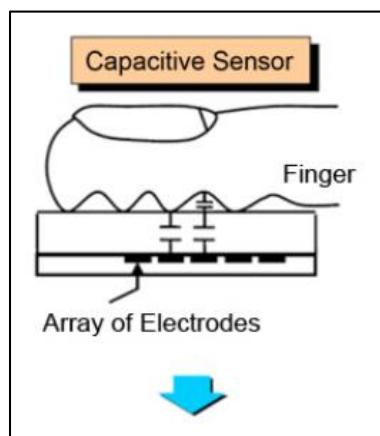


Figure 13 Fingerprint Scanner Capacitive Sensor (taken from [20])

Slap fingerprint technology uses charge-coupled device (CCD) optical sensor to capture multiple fingerprints at once and then separates them into individual segments to extract key features as shown in Figure 14. The slap technology is very secure and quick but only captures plain information instead of detailed data such as nail-to-nail. A typical size of the area which captures the fingerprints is approximately 3.2 x 3.0 in.

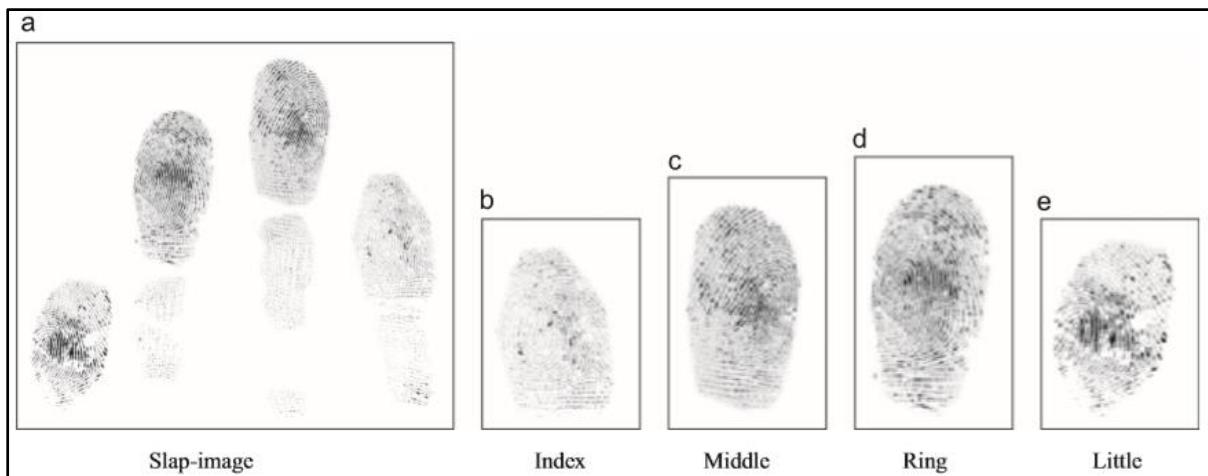


Figure 14 Slap Fingerprint Technology (taken from [21])

The slap fingerprint scanners use near infrared light to capture images [22]. As veins are visible, this type of scanner can also be used to capture vein prints as shown in Figure 15.

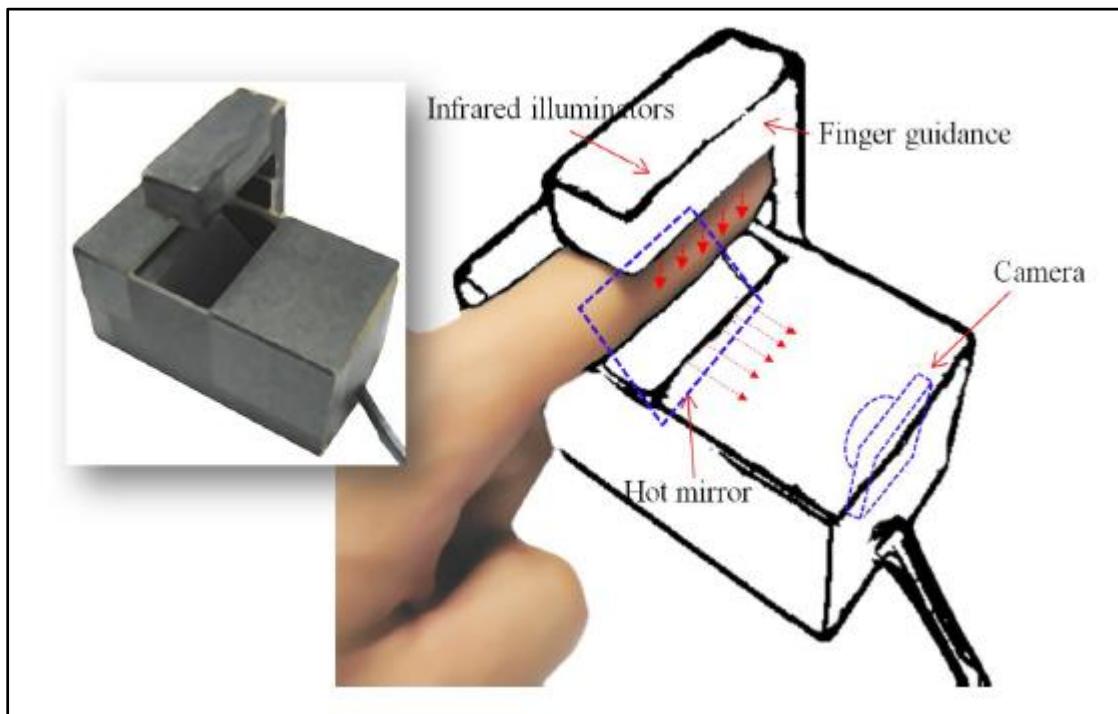


Figure 15 Near Infrared Technology for Slap Fingerprint Scanners (taken from [22])

The touchless swipe sensor is a very welcomed product in the world of Biometrics. This technology addresses the hygiene issue as it does not require contact between skin and surface of scanner as it uses digital camera to acquire a 3D image as shown in Figure 16 . Like most sensors, it also has its own issues [23]:

- It has low contrast between the valley and the ridge pattern of a fingerprint image.
- Lighting is non-uniformed.
- The lack of Depth of Field (DoF) in digital cameras leads to motion blurriness and defocus. DoF refers to the zone of sharpness within a photograph which appears in focus.

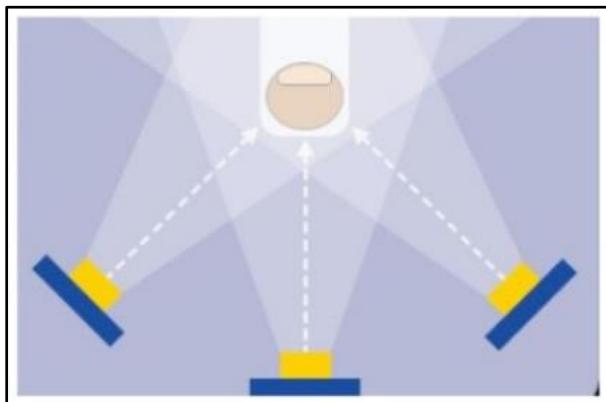


Figure 16 Touchless Fingerprint Sensor (taken from [24])

Touchless fingerprint sensors with large capture area can produce superior image quality and is non-intrusive compared to other fingerprint scanners. However, these can be costly. In 2016, UH was quoted £10,000 for one of the touchless fingerprint scanners which is 93% higher than the standard USB Fingerprint/Fingervein purchased for this project.

The rest of this section looks at how fingerprints are processed once captured as per the system architecture shown in Figure 17.

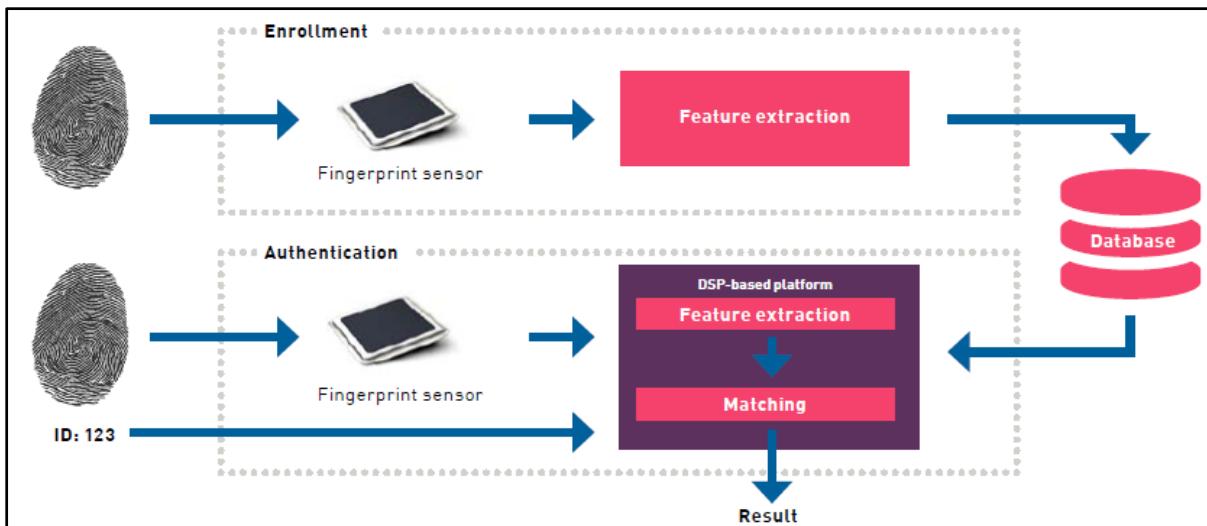


Figure 17 Fingerprint Recognition System Block Diagram (taken from [11])

To summarize, the stages of a fingerprint recognition system are:

- Acquisition – Process where fingerprint is captured using a fingerprint scanner.
- Pre-processing – Images captured are enhanced, converted to grayscale images, go through thinning process and error correction.
- Feature extraction – Key features called minutiae are extracted.
- Matching – Matching against the templates stored in the database/on a device.

3.3.1 Image processing

Once a fingerprint is captured, it needs to be pre-processed before extracting key features. As the image is grayscale, it requires enhancement to improve contrast and remove noise.

Eliminating this process could lead to extraction of incorrect features hence causing issues during matching and identification stages.

Improved scanners reject poor quality images at the time of acquisition, but even high quality fingerprint images require pre-processing before extracting features.

There are two common methods used for fingerprint image enhancement: Histogram Equalization and Fourier Transform [9].

3.3.1.1 Histogram Equalization

Histogram image is a graphical representation of the tonal distribution in a digital image. In the method of histogram equalization, the cumulative density operation is applied to flatten the histogram and extend the dynamic selection of intensity values. The processing of histogram equalization relies on the use of the cumulative distribution function (cdf). The cdf is a cumulative sum of all the probabilities lying in its domain and defined by [25]:

$$cdf(x) = \sum_{k=-\infty}^x P(k) \quad (1)$$

Further image enhancement is performed depending on the global content of the image. The edges and borders between different objects are highlighted. For image $I(x, y)$ with K discrete level, the gray values histogram is determined using the occurrence probability of the gray level i , as shown in the equation below [26]:

$$Prob(i) = \frac{n_i}{N} \quad \text{Where } 0 \leq i \leq K - 1 \quad (2)$$

where N is the total number of pixels in the image, n is the total number of pixels with the same intensity level, and K is the total number of gray level in the image. Before and after fingerprint images following histogram equalization can be seen in Figure 18.



Figure 18 Histogram Equalization Before and After Fingerprint Images (taken from [26])

Figure 19 shows the histograms of before and after histogram equalization is carried out on a fingerprint image.

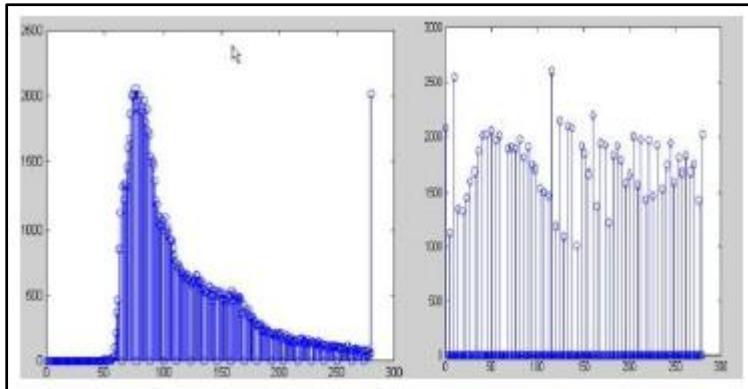


Figure 19 Before and After Histograms Post Histogram Equalization (taken from [9])

3.3.1.2 Fourier Transform

Images are in spatial domain which means that the RGB components of the image vary with their intensity in space, x-axis and y-axis represented as $f(x, y)$. Fourier transform method breaks the image into frequencies. In other words, transforms a spatial image to another domain called the frequency domain or the Fourier domain represented as $F(u, v)$, where u is the frequency change along the x-axis and v is the frequency change along the y-axis [27].

The image is divided into small processing blocks (32 by 32 pixels) and Fourier Transform is performed as follows [28]:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \times \exp \left\{ -j2\pi \times \left(\frac{ux}{M} + \frac{vy}{N} \right) \right\} \quad (3)$$

for $u = 0, 1, 2, \dots, 31$ and $v = 0, 1, 2, \dots, 31$.

In order to enhance a specific block by its dominant frequencies, we multiply the *FFT* of the block by its magnitude a set of times. Where the magnitude of the original *FFT* = $\text{abs}(F(u, v)) = |F(u, v)|$.

An enhanced block can be obtained as:

$$g(x, y) = F^{-1} \{ F(u, v) \times |F(u, v)|^K \} \quad (4)$$

where $F^{-1}(F(u, v))$ is calculated by:

$$F(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u, v) \times \exp \left\{ j2\pi \times \left(\frac{ux}{M} + \frac{vy}{N} \right) \right\} \quad (5)$$

for $x = 0, 1, 2, \dots, 31$ and $y = 0, 1, 2, \dots, 31$.

The K in equation (4) is an experimentally determined constant and it was set to 0.45 in Singh et al's study [28]. While having a higher K improves the appearance of the ridges by

filling up small holes in ridges; having a larger value of K can result in false joining of ridges and a termination might become a bifurcation. Figure 20 shows a sample image enhanced using the Fourier Transform method.

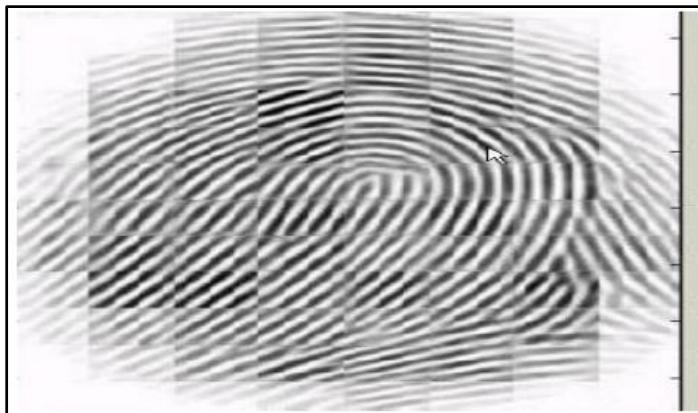


Figure 20 Image Enhanced Using Fourier Transform Method (taken from [28])

There are various other fingerprint quality assessment algorithms as listed in Table 2. The National Institute of Standards and Technology (NIST) Fingerprint Image Quality (NFIQ) is the most popular fingerprint quality assessment algorithm in literature [29].

Table 2 List of Fingerprint Quality Assessment Algorithms (taken from [29])

Category	Authors	Description	Type
Pixel intensity	Chen et al	Grey level distributions of segmented ridges	Local
Wavelet transform	Vatsa et al	Combined response from RDWT for dominant edge information	Local
Power spectrum	Chen et al	In a ring-shaped region of the spectrum	Global
Combined features	NFIQ	Amplitude, frequency and variance of sinusoid to model valid ridges	Global
Orientation tensor	Fronthaler et al	Encode orientation with parabolic symmetry features	Global

3.3.1.2 Thinning

To facilitate an accurate estimation of the locations of key features, fingerprint images also go through a thinning process. This process involves reducing the thickness of the lines as much as possible with minimum loss of data. More precisely, this process aims to achieve the following [10]:

- The lines in the output fingerprint image should be as close to as a single pixel.
- The lines in the output fingerprint image should not have any discontinuity.
- The lines of the output fingerprint image should return to its centre pixel.
- All redundant and unwanted pixels should be eliminated.

Thinning is the final step before extracting minutiae. Figure 21 shows a sample image after thinning.

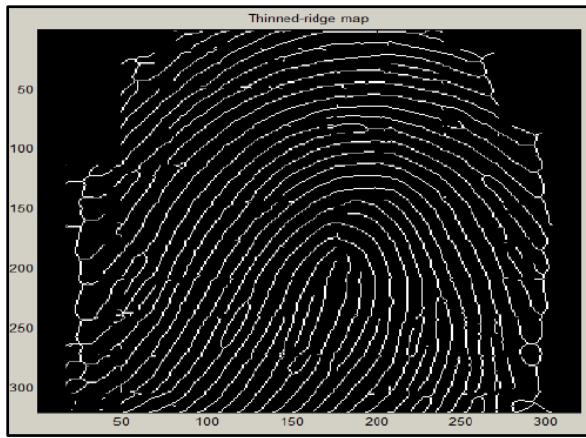


Figure 21 Thinned Fingerprint Image (taken from [28])

3.3.2 Feature extraction

Once an image is enhanced and the ridges are thinned, key features called minutiae are extracted from the fingerprint to create templates during enrolment and authentication stages.

Minutiae are referred to as points of interest in a fingerprint, mainly, bifurcation and ridge endings. Other points of interest include core point, delta point, crossing point and island. Table 3 shows the definition of characteristics of different types of minutiae. A good quality image has around 40 to 100 minutiae [30].

Table 3 Characteristics of Minutiae

Type of Minutia	Characteristics that define it
Termination/Ridge endings	Ridge comes to an end
Bifurcation	A ridge divided into two ridges
Island/Short ridges or independent ridge	A ridge that commences, travels a short distance and then ends [31]
Crossover or bridge	A short ridge that runs between two parallel ridges [31]
Core	Topmost point on the innermost upwardly curving ridge line (approximately centre of the fingerprint) [32]
Delta	Triangular region located near a loop [33]
Ponds/Lake	Empty spaces between two ridges [9]

Clear design structure of each of these minutiae can be seen in Figure 22.

	Termination
	Bifurcation
	Lake
	Independent ridge
	Point or island
	Spur
	Crossover

Figure 22 Types of Minutiae (taken from [34])

3.3.2.1 Key Features Extraction Methods

Minutiae extraction methods generally fall within two categories [30]:

- Ones that work on binarized fingerprint images – Binarized fingerprint images are those that have been converted into binary data via a binarization process, which transforms an enhanced gray-level image into a binary image [35]. A binary image consists of values 0s and 1s. ‘1’ is assigned to a ridge pixel and ‘0’ is assigned to non-ridge pixels [36].
- Ones that work directly on grayscale fingerprint images – Does not use binarization process [35].

Minutiae extraction on binarized fingerprint images can be further classified into unthinned and thinned binarized images. Thinned binarized images are obtained from a skeletonization process that reduces the width of binarized ridgeline to 1 pixel [37]. This is based on the number of neighbouring pixels in the way the minutiae points are detected by location of end points on the thinned ridge skeleton. Bifurcation points are selected if they have more than two neighbouring ridge pixels and end points are selected if they have one neighbouring ridge pixel [30]. Loss of information, being time-consuming and observation of spurious minutiae due to breaks, holes and undesired spikes are some of the problems identified with the binarization and thinning process [30] [38]. Figure 23 shows the result of binarization and thinning process on a gray-scale image.

With gray-scale fingerprint images, the sinusoidal-shapes formed by the combination of ridges and valleys in a local neighbourhood has a well-defined frequency and orientation [36].

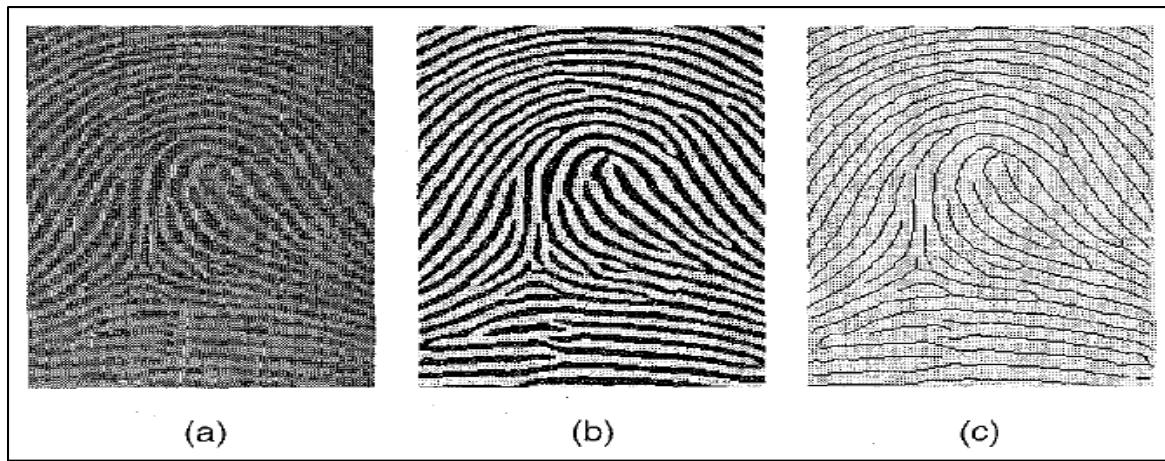


Figure 23 Binarized and Gray-scale Images (a) Gray-scale Image, (b) Binarized Image, (c) Image obtained after thinning process of binarized image (taken from [38])

3.3.2.2 Key Features Extraction Algorithms

As shown in Figure 24, a typical feature extraction algorithm involves five operations [13]:

- 1) Orientation estimation: to estimate local ridge directions.
- 2) Ridge detection: separate ridges from the valleys by using the orientation estimation, resulting in a binary image.
- 3) Thinning algorithm/skeletonization: reducing the ridge width to one pixel.
- 4) Minutiae detection: identifying ridge bifurcations (those with three ridge pixels in its neighbourhood) and ridge endings (those with one ridge pixel in its neighbourhood).
- 5) Post processing: removes spurious minutiae.

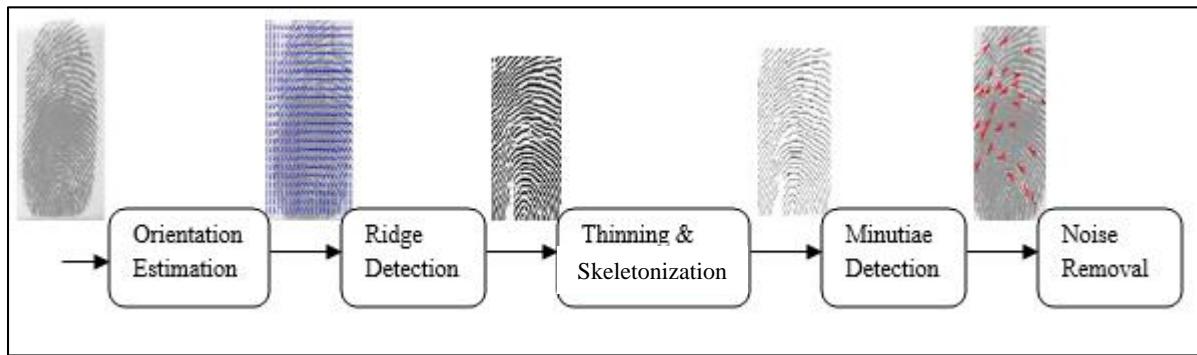


Figure 24 Fingerprint Feature Extraction Algorithm (images taken from [13])

Orientation Estimation: Orientation field estimation plays a vital role in fingerprint recognition system. A good orientation estimation can help improve the performance of a fingerprint recognition system and a poor estimation often weakens the performance of a fingerprint recognition system. Orientation field reveals the global pattern of a fingerprint and meticulously illustrates the basic structure, directional feature and shape of a fingerprint [39].

Various algorithms which are based on filter-bank approaches, gradient-based approaches and micro-patterns as binary gradients, as well as 2D spectral estimation methods and high frequency power in 3D space have been proposed in literature for orientation field estimation [40]. The filter-bank and gradient-based approaches are the most popular ones to be used. Gradient-based approaches are more accurate compared to filter-bank based approaches due to their limited number of filters. Nevertheless, filter-bank based approaches are more

resistant to noise, i.e. breaks and smudges in comparison to gradient-based approaches. The filter-bank approaches are also computationally expensive as it needs to carry out comparison of all filters' output. Figure 25 shows two fingerprints: one with direction of minutiae (a) and the other shows the orientation field (b).

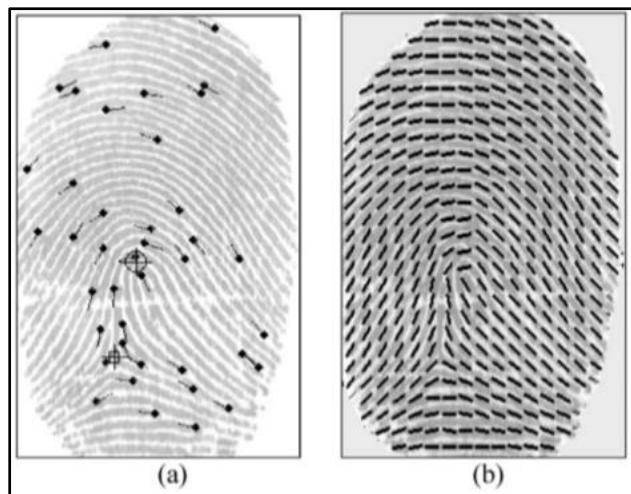


Figure 25 (a) Singular points and minutiae with its direction and (b) orientation field shown with unit vector (taken from [40])

Ridge Detection: The next stage is to separate ridges from the valleys in a given fingerprint image. This is the main objective of a ridge detection algorithm. The gray-level values on ridges attain their local minima along a direction normal to the local ridge orientation making it a more reliable property to utilise for ridge detection [41].

Thinning: Thinning or skeletonization is a process of reducing thick fingerprints into 1 pixel wide ridge lines. After this stage, each minutia is described by a feature vector containing its (x,y) coordinates, type and orientation.

Minutiae Detection: Ridge endings and bifurcation points are identified using a minutiae detection process by examining a local neighbourhood of window size 3x3. In [42], pixels in a fingerprint are examined within their 3x3 neighbourhoods for their connectivity as shown in Figure 26 and the number of ridge pixels in a neighbourhood is referred to as Crossing Number (CN). If the CN value for a central pixel is 3, it is identified as a bifurcation point. If the CN value is 1, it is identified as a ridge ending point. More details on CN is covered later on in this subsection. In the minutiae detection process, spurious minutiae results are also produced in the thinned images. These need to be post-processed.

 Bifurcation Point (CN=3)	 Ridge Ending Point (CN=1)
---------------------------------	----------------------------------

Figure 26 Crossing Number (CN) value for Bifurcation and Ridge Ending (taken from [43])

Post-processing: Any other unwanted patterns of minutiae are removed from the image. These include spikes, breaks and holes. Morphological operators may be used for this purpose.

Figure 27 shows a flow chart of the results of a typical minutiae extraction algorithm. Following extraction of minutiae, a stored template could typically contain the minutiae position (x, y) minutiae direction (angle) and minutiae type (ridge ending or bifurcation) [28]. During the enrolment and authentication stages, the template is stored in the database to then be used in the matching process as a reference template.

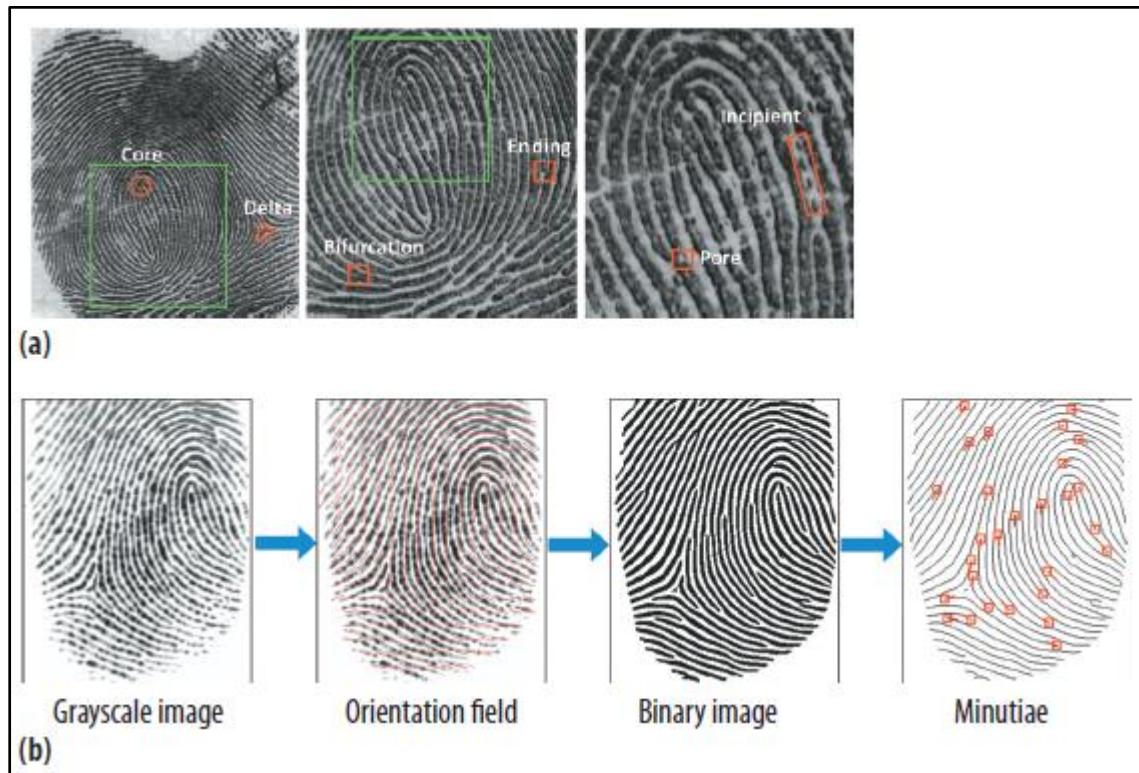


Figure 27 Key Feature Extraction. (a) Feature levels in a fingerprint. (b) Flow chart of a typical minutiae feature (taken from [44])

As mentioned at the beginning of this section, the feature extraction algorithms differ depending on whether binary or gray-scale images are used and whether skeletonization is applied or not. Categorization of these algorithms is shown in Figure 28.

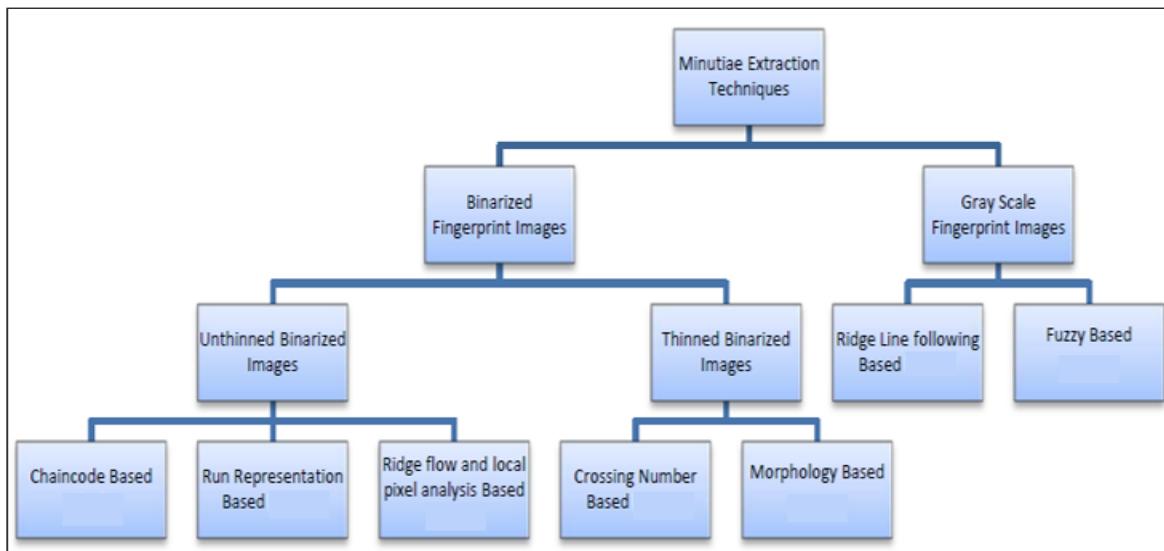


Figure 28 Classification of Key Features Extraction Techniques (taken from [30])

As methods based on thinning are sensitive to noise, techniques such as chaincode processing, run based methods and ridge flow and local pixel analysis based can be used to replace the thinning process [30]. The chaincode processing method uses object contours by scanning image from top to bottom and right to left to identify the transitions from white background to black foreground. This follows a representation of an array of contour elements which are traced counter clockwise and each element denotes a pixel on the contour. A minutia ending is located when the ridge makes a left turn by tracing a ridge line along the boundary counter clockwise. If it makes a right turn, a bifurcation minutia is identified [30] [45].

The run representation based method results in fast extraction of fingerprint minutiae that are based on the horizontal and vertical run-length encoding from binary images. Characteristic images are then found by checking the runs adjacency of the runs [45]. This minutiae extraction technique does not require a computationally expensive thinning process. The ridge flow and local pixel analysis technique uses a 3x3 square mask to compute the average of pixels around each pixel in the fingerprint image. A ridge termination minutia is identified if the average is less than 0.25 and those greater than 0.75, determine a bifurcation minutia [30].

In the morphology based method which falls under thinned binarized images feature extraction technique, the image is pre-processed to reduce the effort in the post processing stage. Spurs, bridges etc. are removed with morphological operators followed by use of morphological hit or miss transform to extract true minutiae. Morphological operators are shape operators whose composition allows the natural manipulation of shapes for the identification and the composition of objects and object features [30].

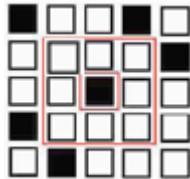
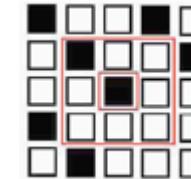
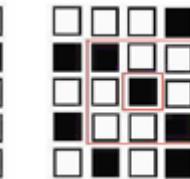
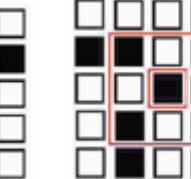
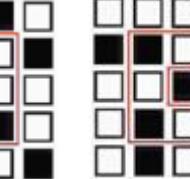
Within fingerprint recognition systems, the concept of a Crossing Number (CN) as a feature extraction process is widely used. The CN method uses a skeleton image where the ridge pattern is connected by eight neighbouring pixels (denoted by N8). This refers to an investigation of the 8 neighbouring pixels of the central pixel, P. The minutiae are extracted by scanning the local neighbourhood of each ridge pixel in the image using a 3x3 window as shown in Figure 29. It then assigns a CN value based on type of central pixel and its neighbouring pixels as shown in Table 4. The grid in the first row of the table shows sub-

images of size 5x5. In (a), the central pixel highlighted in black has no ridge pixel in its 3x3 neighbourhood. Hence, the CN value returned is 0. In the remaining grids, there is an increasing value of CN due to the number of connected ridge pixels in its neighbourhood.

P4	P3	P2
P5	P	P1
P6	P7	P8

Figure 29 A 3x3 pixel window (taken from [32])

Table 4 Crossing Number Point System

				
(a) Non-Minutiae N8=0, CN=0	(b) Ridge ending N8=1, CN=1	(c) Continuing ridge N8=2, CN=2	(d) Bifurcation point N8=3, CN=3	(e) Complex minutiae N8>3, CN=4

The CN values for a ridge and bifurcation is calculated using a formal definition for CN as follows:

$$CN = 0.5 \sum_{i=1}^8 |P_i - P_{i+1}|, P_9 = P_1 \quad [32] \quad (6)$$

This is translated as half the sum of the differences between pairs of adjacent pixels in the eight neighbouring pixels. P_i is the pixel value that is in the neighbourhood of the central pixel, P where $P_i = 0$ or 1 and $P_9 = P_1$ [23]. This method also involves recording of locations of the minutiae points and their considered directions: N, S, W, E, NE, NW, SE and SW. The most important minutiae points are the ridge endings and bifurcation hence the angle and direction of these minutiae are very important. The directions can be determined for the conditions of CN=1 and CN=3 [46] using the following pseudocode in Table 5.

Table 5 Pseudocode for Determining Direction from CN

% Direction for ridge ending point if CN=1 then if P1=1 then direction =W if P3=1 then direction =S if P7=1 then direction =N if P5=1 then direction =E if P4=1 then direction =SE if P2=1 then direction =SW if P6=1 then direction =NE if P8=1 then direction =NW end if	% Direction for ridge bifurcation point if CN=3 then if P1 and P3 and P7=1 then direction =W if P1 and P3 and P5=1 then direction =S if P1 and P7 and P5=1 then direction =N if P3 and P5 and P7 =1 then direction =E if P4 and P3 and P5=1 then direction =SE if P3 and P2 and P1=1 then direction =SW if P3 and P5 and P6=1 then direction =NE if P4 and P8 and P5=1 then direction =NW end if
--	--

3.3.3 Matching process

Biometric authentication can be split into two categories: verification and identification. Verification is a one-to-one matching process where the template may be stored in a token such as ID card or passport. The token is presented to the system and checked against the relevant physiological characteristic presented at time of authentication. The other form is identification which carries out one-to-many match within a database of stored templates. Verification is much quicker than identification.

In fingerprint verification, decision making process considers if the matching score exceeds a threshold. If so, it 'Accepts' the fingerprint, otherwise it 'Rejects' the fingerprint. A user is required to present their token which has the template stored on it, followed by presenting a finger to the fingerprint reader to verify if the two templates match. When a user presents their finger to a reader, the key features are extracted to then perform a verification check against the template stored on the token.

In fingerprint identification, the decision making process considers two possibilities:

Close-set identification which returns the identity associated with the highest matching score.

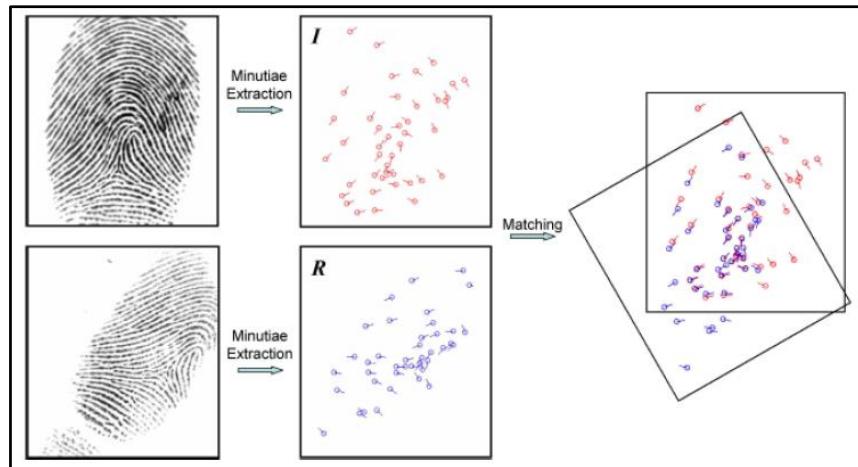
Open-set identification which returns the identity associated with this score if the score is higher than a threshold or otherwise declare that the test biometric data does not belong to any of the registered individuals.

The Automatic Fingerprint Identification Systems (AFIS) have been developed for human identification in which the matching module computes a matching score between two fingerprints. A high score is assigned to fingerprints from the same finger and a low score to those that do not match. To claim that two fingerprints are from the same finger, the following factors are evaluated [1]:

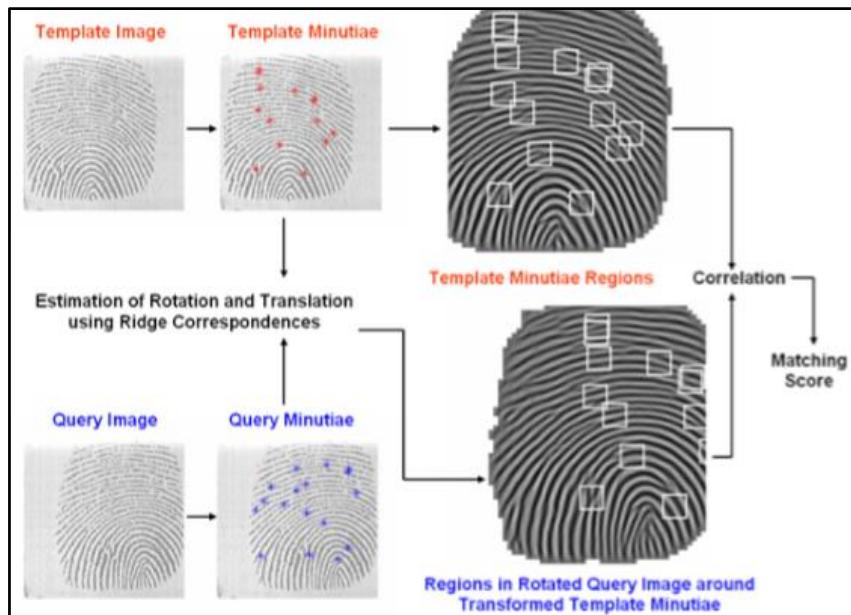
- Global pattern configuration agreement – two fingerprints must be of the same type.
- Qualitative concordance – corresponding minutiae must be identical.
- Quantitative factor – at least a certain number of minutiae details must be found.
- Corresponding minutiae details – must be identically inter-related.

There are various fingerprint matching techniques discussed by researchers and experts but the most commonly used in fingerprint recognition systems are [1] [12]:

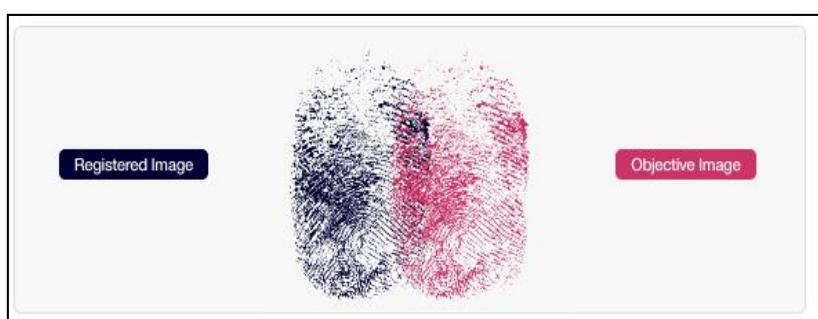
- **Minutiae-based:** matches local features such as bifurcations and ridge endings based on location and direction of each point as shown in Figure 30(a). This is the most accurate technique [47].
- **Correlation-based:** two fingerprint images are superimposed and the correlation between corresponding pixels is computed for different alignments such as, various displacements and rotations as shown in Figure 30(b). This approach requires less computation but is less robust against image distortions [12].
- **Pattern-based or ridge feature-based:** compares two images for similarity such as shape, texture information, local orientation, ridge shape and frequency as shown in Figure 30(c). This technique is faster compared to minutiae-based approach but it includes every macro and micro part of the fingerprint [48].



(a) Minutiae-based Matching (taken from [49])



(b) Correlation-based Matching (taken from [50])



(c) Pattern-based Matching (taken from [51])

Figure 30 Fingerprint Matching Techniques

3.3.3.1 Minutiae-based Matching

In this algorithm, each minutia may be described by several attributes including its location in the fingerprint image, orientation, type (i.e. ridge termination or ridge bifurcation) and a weight based on the quality of the fingerprint image in the neighbourhood of the minutia etc. Let T and I represent the template and input fingerprints respectively, each represented as a

feature vector as given by equation (7) and equation (8) and whose size is determined by the number of minutiae m and n respectively.

$$T = (m_1, \dots, m_m), m_i = (x_i, y_i, \theta_i), i = 1..m [1] \quad (7)$$

$$I = (m'_1, \dots, m'_n), m'_j = (x'_j, y'_j, \theta'_j), j = 1..n [1] \quad (8)$$

Minutiae m'_j in the input image and m_i in the template image are considered matching if the spatial distance sd between them is smaller than a given tolerance r_0 and the direction difference dd between them is smaller than an angular tolerance θ_0 :

$$sd(m'_j, m_i) = \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2} \leq r_0 [1] \quad (9)$$

$$dd(m'_j, m_i) = \min(|\theta'_j - \theta_i|, 360^\circ - |\theta'_j - \theta_i|) \leq \theta_0 [1] \quad (10)$$

Equation (10) takes the minimum of $|\theta'_j - \theta_i|$ and $360^\circ - |\theta'_j - \theta_i|$ because of the circularity of angles (e.g. the difference between angles of 2° and 358° is only 4°).

3.3.3.2 Correlation-based Matching

The correlation based matching technique uses a sum of squares approach to determine the match between the test input and the templates in the database. This is given by:

$$SSD(T, I) = |T - I|^2 = (T - I)^T (T - I) = |T|^2 + |I|^2 - 2T^T I [1] \quad (11)$$

where SSD represents the sum of squared differences between the template and input images and is an indicator of the diversity between these two images, T represents the template image and I represents the input fingerprint, and the superscript T represents the transpose of a vector.

If the terms $|T|^2$ and $|I|^2$ are constant, the diversity between the two images is minimised when the cross-correlation CC is maximised:

$$CC(T, I) = T^T I [1] \quad (12)$$

where $CC(T, I)$ is a measure of image similarity.

The similarity cannot be simply computed by simply superimposing T and I and applying the above equation due to the displacement and rotation. The similarity between two fingerprint images T and I can be measured as:

$$S(T, I) = \max_{(\Delta x, \Delta y, \theta)} CC(T, I^{(\Delta x, \Delta y, \theta)}) [1] \quad (13)$$

$I^{(\Delta x, \Delta y, \theta)}$ represents a rotation of the input image I by an angle θ around the origin (often the image centre) and shifted by $\Delta x, \Delta y$ pixels in directions x and y , respectively.

3.3.3.3 Pattern Matching

With the pattern matching technique, each fingerprint is represented by a feature vector of size $80 \times 8 = 640$ and is called the *Finger Code*. The formula for pattern matching is given by:

$$V_{ij} = \frac{1}{n_i} \left(\sum_{C_i} \left| g\left(x, y; \theta_j, \frac{1}{10}\right) - \bar{g}_i \right| \right) [1] \quad (14)$$

where

- V_{ij} of the vector ($i = 1.80$ is the cell index, $j = 1.8$ is the filter index) denotes the energy revealed by the filter j in cell i , and is computed as the average absolute deviation from the mean of the responses of the filter j over all the pixels of the cell i as above.
- C_i is the i th cell of the tessellation (tiling of fingerprint area of interest with respect to the core point).
- n_i is the number of pixels in C_i .
- The local texture information in each sector is decomposed into separate channels by using a Gabor filter bank (fingerprint enhancement method) so $g(\cdot)$ is defined by the Gabor filter equation.
- \bar{g}_i is the mean value of g over the cell C_i .

Research shows that due to the large variability in different impressions of the same finger, matching fingerprint images is an extremely difficult problem [1]. Some of these variations are due to:

- **Displacement:** The same finger may be placed in different locations on the fingerprint capture device sensor.
- **Rotation:** As per displacement, the same finger may be placed at different angles on the sensor.
- **Pressure and skin condition:** Finger pressure, dryness of the skin, skin disease, grease, dirt, sweat and humidity in the air can all result in a non-uniform contact on the sensor.
- **Noise:** Introduced by the fingerprint sensing system. For example, residues left over from the previous fingerprint capture.

3.3.3.4 Fingerprint Performance Metrics

With fingerprint biometrics authentication, a similarity measure is estimated between an input and the template. The system accepts or rejects the claimed identity based on whether the similarity score is above a pre-determined threshold. Such a threshold is determined through experimentation of the distributions of similarity measures for the specific database. In general, the distribution of similarity measures of genuine and imposter attempts do not have a crisp boundary of thresholds. An overlap exists between these two categories of attempts as shown in Figure 31. The matching performance is thus measured by two error measures [50]:

- False Accept Rate (FAR) or False Match Rate (FMR) – the likelihood of a fingerprint system incorrectly matching an input pattern to a non-matching template in the database. This is shown by the dark gray shaded area in the Figure 31.
- False Reject Rate (FRR) or False Non-Match Rate (FNMR) – the likelihood of the system failing to detect a match between the input pattern and a matching template in the database as shown by the light gray shaded area in the Figure 31.

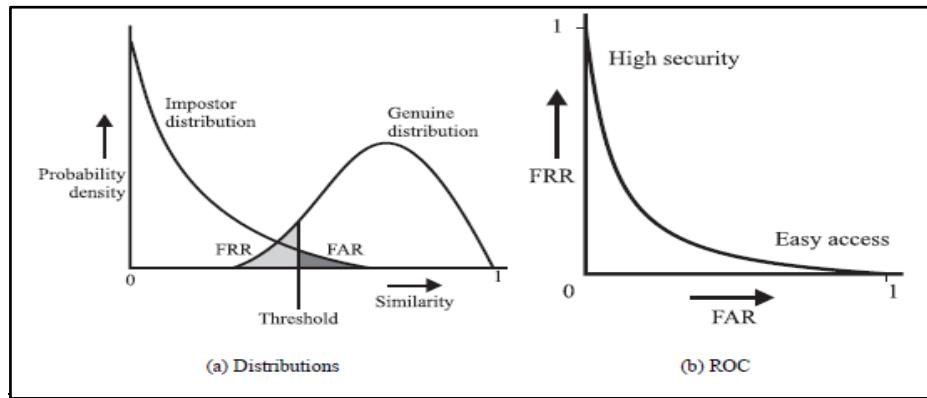


Figure 31 Match and Non-match Distributions and Receiver Operating Curve (ROC) (taken from [15])

It is reported in [15] that advanced systems have $FAR = 10^{-4}$ and $FRR = 10^{-2}$. More realistic systems have error rates of have $FAR = 10^{-2}$ and $FRR = 5 \times 10^{-2}$. A receiver operating curve (ROC) provides insight into the system performance by plotting the FAR against FRR. This enables to tune the threshold to meet the requirements of an application. Systems requiring high security such as airports can set up a high threshold where FAR is very low whilst others as in a classroom access may prefer easy access and therefore lower thresholds with low FRR. Identification systems often use an equal error rate as a performance metric and this defines the specific point on the ROC where FAR is equal to FRR.

3.3.3.5 Fingerprint Databases

Several databases are available for training and testing a fingerprint system largely due to the research community generating its own databases as well as international organisations that run competitions in the field. Some databases are listed below:

- National Institute of Standards and Technology (NIST) Fingerprint Database 4** [52]: This database consists of 2000, 8-bit gray scale fingerprint image pairs. Each image is 512x512 pixels with resolution 19.7 pixels/mm and 32 rows of white space. There are five classes of images namely arch, left loop, right loop, tented arch and whorl.
- Chinese Academy of Sciences' Institute of Automation (CASIA)-FingerprintV5** [53]: There are two datasets, one with the subject's co-operation and the other without the co-operation. Dataset 1 has 17 users with fingerprints from middle fingers of both hands and a total of 4 samples. Three different sensors are used. The dataset dimensionality is 68 fingers x 4 samples x 3 sensors = 816 image samples. Similarly, the non-co-operative dataset dimensionality is 64 fingers x 4 samples x 3 sensors = 768 images.
- FingerDOS** [54]: This is a collection of fingerprint images acquired using an optical sensor. The database consists of 60 subjects with a total of 3600 fingerprint images acquired from thumb, index finger and middle finger of both left and right hands. Each subject has 10 samples and saved as 8-bit gray scale images.
- Spoofed Fingerphoto Database** [55] [56]: This database consists of spoofed fingerprints where co-operative fingerprints were captured with two different sensors and the spoofed fingerprints were acquired using iPad, Laptop and printout. It

consists of 64 subjects x 2 fingers = 128 classes. Further the database varies in illumination and backgrounds two different illumination conditions.

Both systems, one which has been developed and commercial system used in this project use minutiae-based matching on grayscale images, with the matching process carried out using open-set identification.

Following subject review, key objectives of this research project were outlined:

- i. Developing a reliable fingerprint recognition system.
- ii. Deploying such a system within a University environment by linking it to existing University data sources.
- iii. Making University's exam registration process more efficient and reliable.

CHAPTER 4: PRACTICAL WORK

After gaining knowledge about fingerprint technology, some of the learning was applied to the applications used in this project. This chapter covers details of practical work carried out since March 2016.

4.1 Survey

A survey was issued to students of University of Hertfordshire (UH) to gather their thoughts on using fingerprint technology within and outside UH.

The survey consisted of 7 questions created on the Survey Monkey website using University's account: <https://www.surveymonkey.co.uk/r/FingerprintTechnology> Most of the questions were closed and the survey was open from 05th December 2016 to 22nd December 2016 and re-opened during Semester A exams: 03rd January 2017 to 06th January 2017. There was no incentive offered to complete the survey. The survey was checked by the English Language Team/Academic Skills Humanities and Pro Vice-Chancellor (Education and Student Experience) before it was issued. Ethics approval was obtained before conducting the survey.

4.1.1 Methodology

The survey was issued to students via various methods:

- Link to survey was posted as a news item on our internal StudyNet site which is visible to all UH students.
- The survey was also promoted on our screens. A 'UH Screens Broadcast Request Form' was required by the Marketing and Communications team. Appendix II shows the completed request form.
- University's Facebook page managed by our Marketing and Communications department.
- A4 table signs within student areas such as Student Centre.
- Link was sent to Schools following advice from Deans who also placed the link on the School's StudyNet pages.
- Face-to-face approach was also taken by approaching students visiting the Student Centre and Library.

Figure 32 and Figure 33 show two of the methods used to promote the survey. The rest are shown in Appendix III.

University News



Attention All Students!

Rupa_4 Patel on the 05/12/2016 18:51:24

Fingerprint Recognition System

As part of a research project, we would like to gather your thoughts on using fingerprint recognition system to improve processes at UH.

It will take less than 5 minutes to complete. Your feedback is greatly appreciated.
Thank you!

[Share your thoughts now!](#)

Share your thoughts now!

This research project has received Ethics approval from the Science and Technology ECDA committee. Protocol number: ENT/PGR/UH/02043

<https://www.surveymonkey.co.uk/r/FingerprintTechnology>

Figure 32 Survey Link on StudyNet

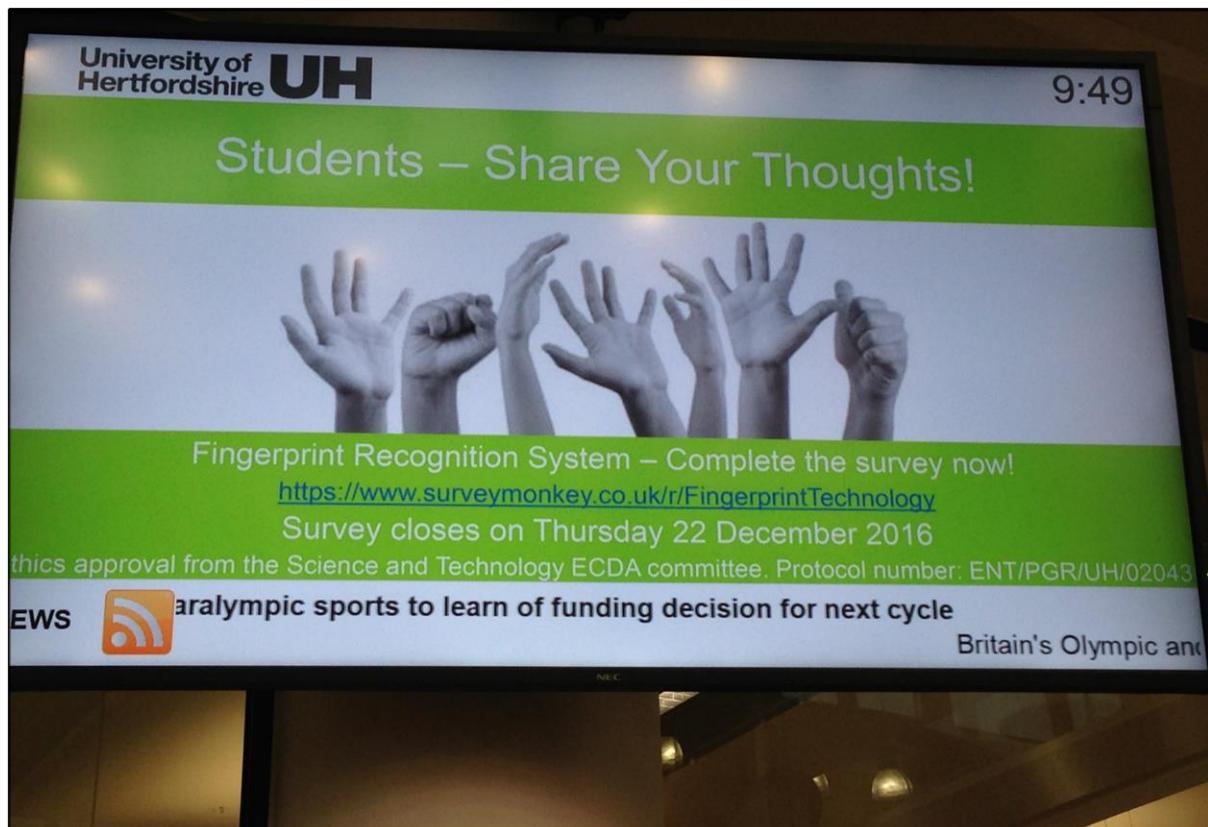


Figure 33 Survey Promoted on UH Screens

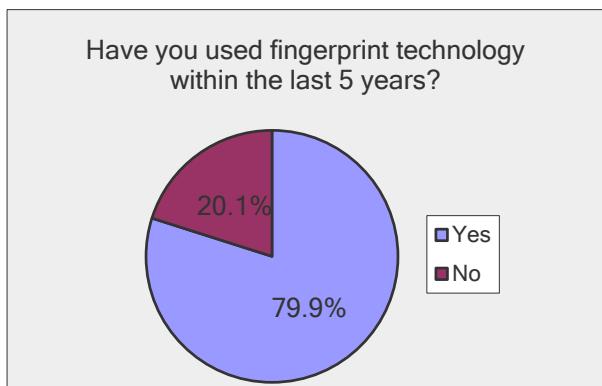
4.1.2 Survey Results

There were 817 respondents with 754 complete responses and 186 comments received for Question 6 on storage of fingerprints.

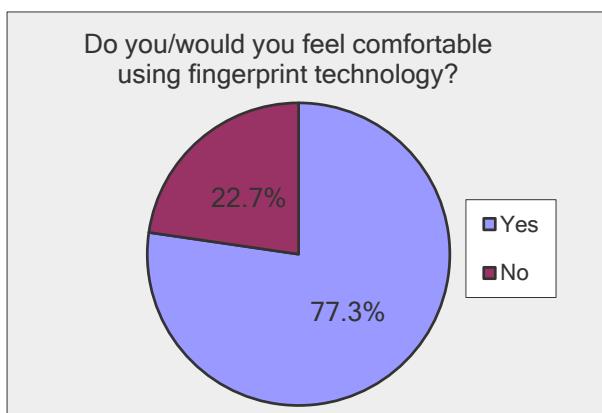
The following results were also shared with colleagues across UH on 27th February 2017. Some of these include Exams and Awards team, Deans of Schools, Library and Computing Services, Students Union and Office of the Vice-Chancellor.

Overall, the survey received around 70% positive response for majority of the questions asked. 79 respondents answered 'No' to all questions/were not in favour of this proposal.

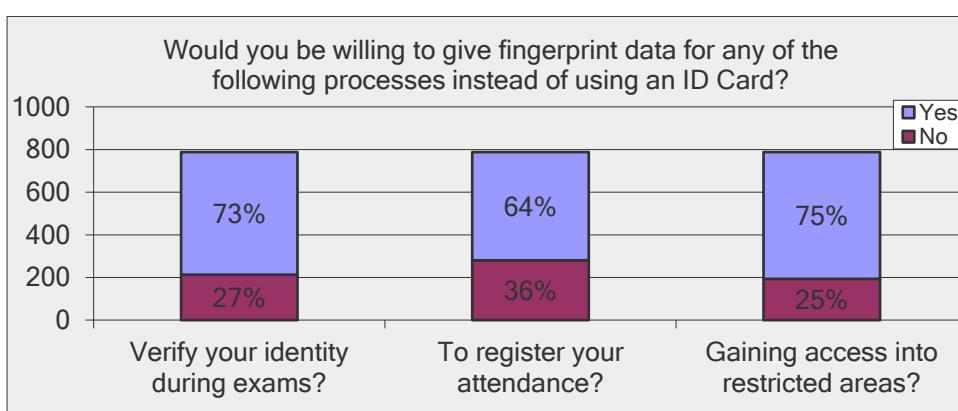
Question 1:



Question 2:

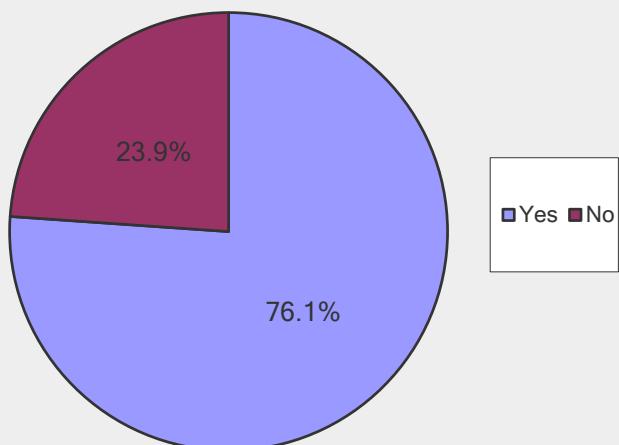


Question 3:

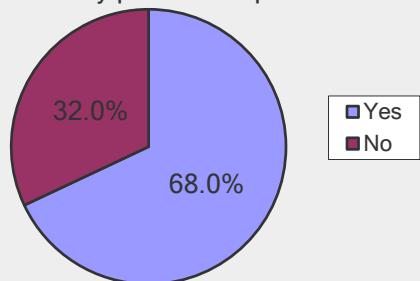


Question 4:

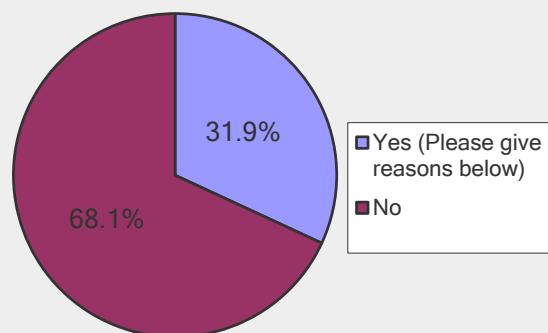
Considering that the cost of replacing an ID card is £10 (for lost ID card), would you prefer using fingerprint recognition system instead of paying for a replacement one in urgent cases i.e. during exams, checking in for attendance, gaining access

**Question 5:**

Do you think that using a fingerprint recognition system would make University processes quicker?

**Question 6:**

Would you have any concerns with the University of Hertfordshire storing your encrypted fingerprint data for the duration of your course at UH?



186 respondents commented on Question 6 and 141 of these provided comments relating to data security, hacking and privacy rights/issues. 18 respondents provided comments such as: "If they were taken off private record once we leave university, I dont see this being a problem, good idea!". 27 respondents provided neutral comments such as: "Injured/burnt finger?".

Table 6 summarizes some of the comments shared by respondents.

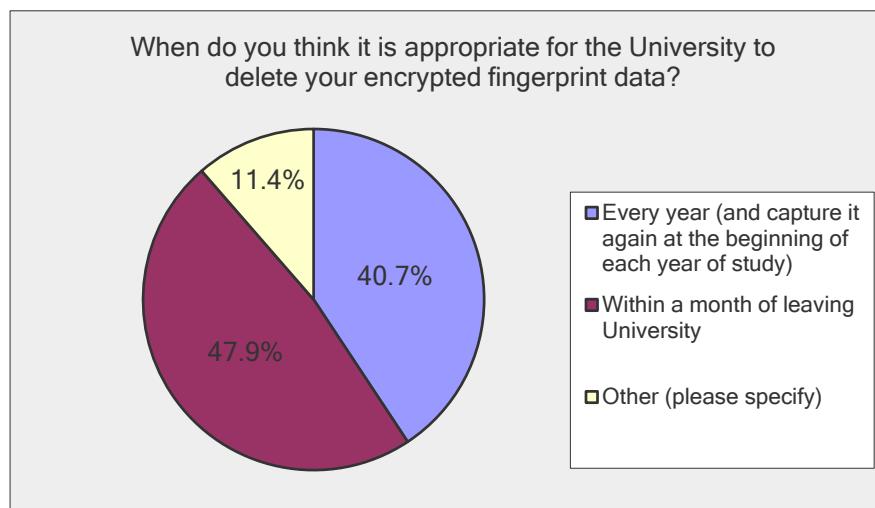
Table 6 Comments from Survey Respondents

Support	Reluctance
<ul style="list-style-type: none"> • Good idea • "Acceptable for exams" • "Could it possible be used in the Learning Resource Centre" • Prevents "using other people's ID" • More convenient and efficient • "You can't forget your finger" • For duration of course, it's alright 	<ul style="list-style-type: none"> • Privacy concerns – data storage/sharing of this information, hacking/theft, "risk" • Hygiene • System tends to fail – Network • Associate it with crime/being Tracked • Cost – "Expensive" • "Might be slower than ID" • "ID process is efficient", "Still need ID" – Student discounts

To address some of the concerns raised in the survey, appropriate measures have been taken. For example, to maintain privacy, only encrypted templates will be stored in a secure database. To mitigate risks of network failure, the exam class list will be uploaded to the local device and a local copy of the data collected during exam registration will be stored.

In terms of "Might be slower than ID", the average identification time is 4 seconds according to our pilot study.

Question 7:



These survey results were also published in a paper entitled "Defining a Pilot Experiment to Enhance Exam Registration with Fingerprint Biometrics" which was selected for an oral presentation at the InnoEducaTIC 2017 Conference held at University of Las Palmas, Gran Canaria on Thursday, 16th November 2017.

4.2 System Architecture and Processes

Following the survey, works on the system were commenced. This subsection covers details of the system.

Like all biometrics systems, the use of our fingerprint system involves two phases: enrolment and authentication. Figure 34 shows an overview of the system for the purpose of exam registration.

The enrolment process in the completed system would require a user to present his/her finger to a fingerprint sensor. The system enrols one fingerprint from each hand. The reason for capturing two fingerprints is to have a backup in place in case one finger cannot be used at authentication stage (in case of bruise or cut). The relevant features would be extracted and converted to an encrypted template. This would then be stored, with the provided University ID number, finger ID (i.e. 2 for right index finger) and enrolment date into a database on a secure server. This enrolment process is independent of any other systems or University databases.

Unlike the enrolment process, the fingerprint authentication process of this system always relies on the information provided by other UH systems. As shown in Figure 34, in the case of exam registration, the system relies on the information from both the ID card system and the exams system of our University which already exist and have not been amended for this project. During the authentication process, the user will be required to present their finger to a fingerprint scanner to be identified before they can be allowed to take their exam. The relevant features will be extracted and checked using feature matching process to generate matching scores against some of the templates enrolled in the fingerprint database, with the highest matching score and its associated ID number being returned.

If the highest score is greater than the open-set identification threshold, then the ID number from the fingerprint database will be returned and the user will be registered for the exam as an expected candidate. Otherwise, another process (for example the current ID card check process or a fingerprint matching between the test template against all the templates enrolled in the fingerprint database) will take place to obtain the ID number of the user and the user will be registered for the exam as an unexpected additional candidate. As the IDs in the fingerprint database, the ID card database and the exams database are identical of the same user, the ID number returned by the fingerprint matching process (either via the 'yes' or the 'no' branch) will be used to retrieve the full personal details of the identified user from the University's ID card database. These user details will finally be stored in the Fingerprint Application database, along with the exam information such as the location, and the unit Id of the exam registration unit used.

This system architecture is unique as the standard fingerprint system architecture mostly relies on one database only: the one in which the fingerprints are stored. In comparison, our system retrieves data from three databases. It is important to link the ID database to retrieve photographs, first name and last name and the Exams database is used to retrieve the student's exam information i.e. exam date and time, room number and candidate number. This information is then used to update the fingerprint application database to generate reports used by Exams Office.

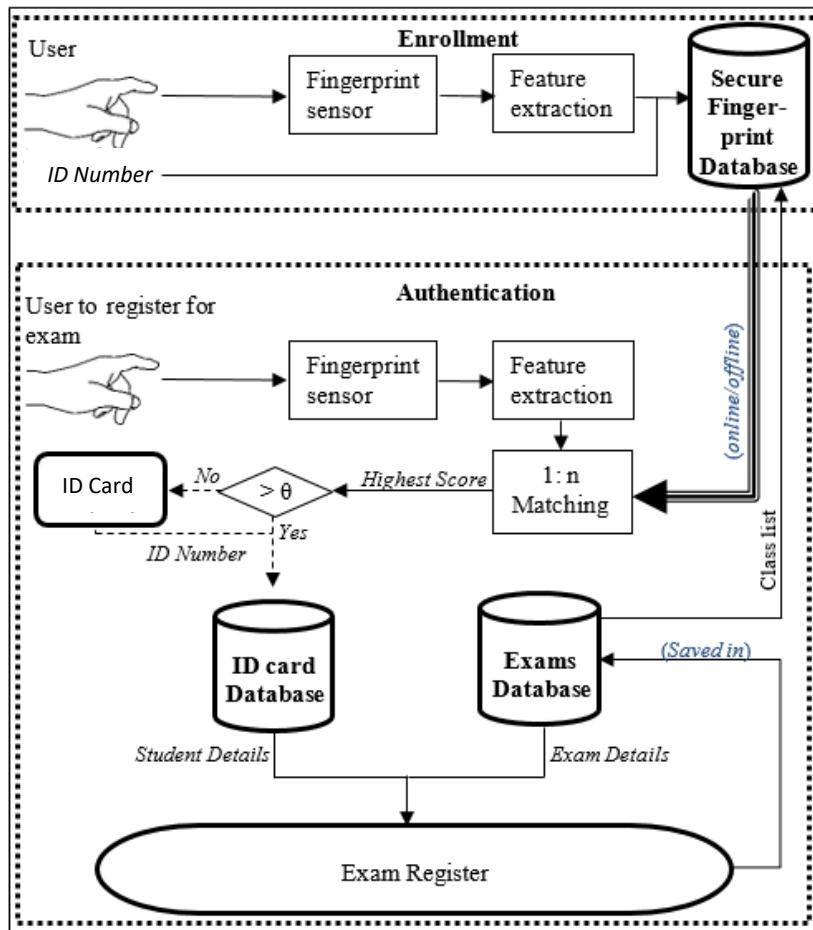


Figure 34 Fingerprint Recognition System

The registration process before an exam must be as efficient as possible and is expected to last less than 15 minutes for the entire exam room. The entire student population size of our University is more than 27,000. This means that the fingerprint matching for each exam candidate would require a 1-to-27000 matching and there are often more than 100 candidates in an exam room. To make the fingerprint-based exam registration more efficient and achievable within the required timeframe, the system retrieves an exam class list from the University exams database. This exam class list shows which users are expected in the particular room of examination. This list is then used to retrieve the fingerprint templates of the expected exam candidates. These fingerprint templates can be retrieved from the fingerprint application database online in real time or they can be downloaded offline and stored on the local fingerprint terminal beforehand. This additional operation can significantly reduce the number of fingerprint matching per candidate from 27,000 (size of the student population at the University) down to less than 30 (number of candidates in a small exam room).

4.3 Databases

Before commencing the project, an Entity Relationship Diagram (ERD) was created which was used as a guide to create the databases for two systems: Fingerprint Application system which has been covered in detail in Section 4.4 and the other is an application written to be used by ID Office staff and Exams and Awards team. Appendix IV shows version 1 of the ERD.

As the project progressed, the ERD was modified as shown in Figure 35.

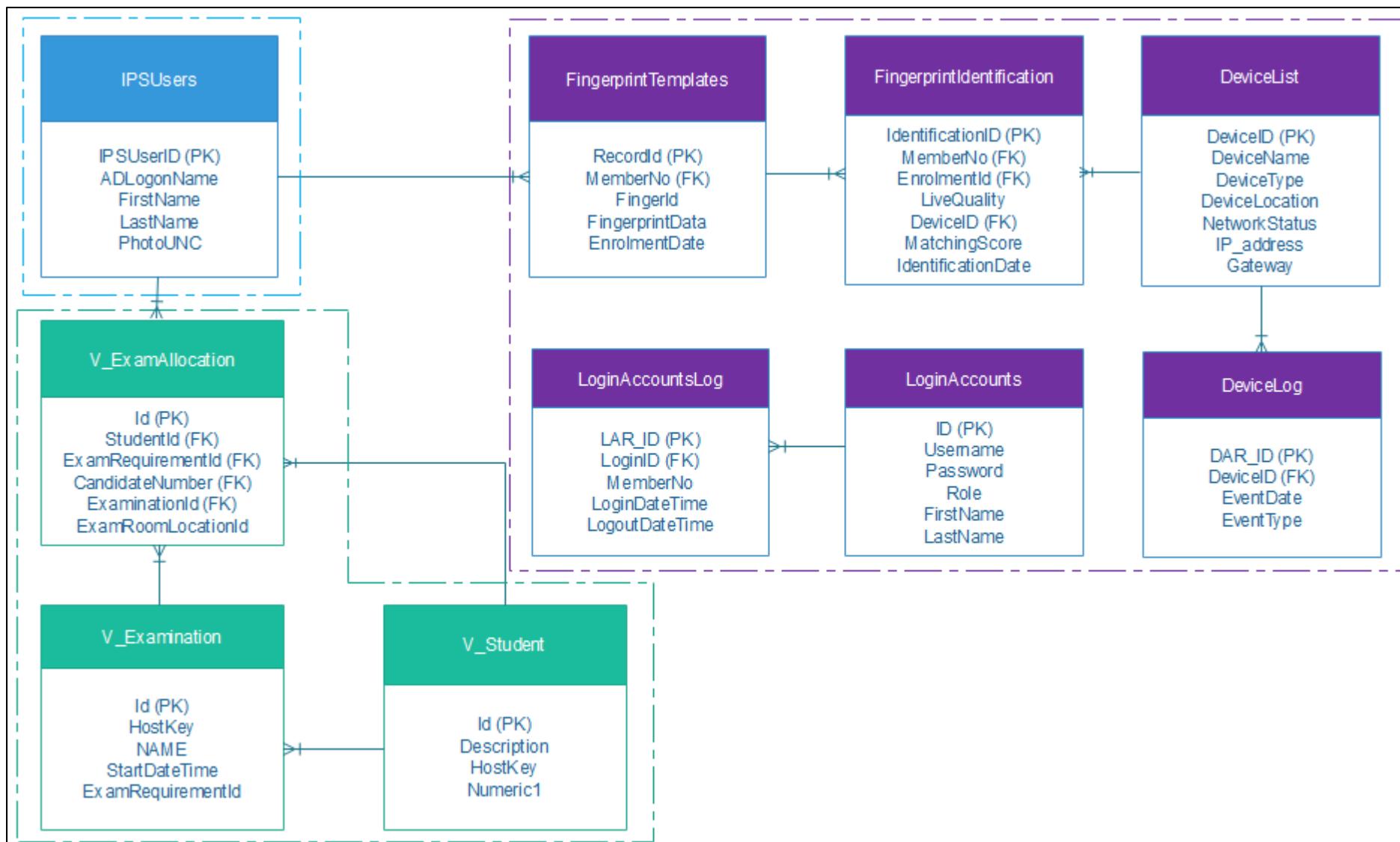


Figure 35 Entity Relationship Diagram v2

The table in light blue exists within the ID Card database, the ones in purple have been created for the Fingerprint Application database and the views in teal exist within the Exams database. This section details the current database structure and how it links with other databases and the main Fingerprint Application systems.

Microsoft SQL Server 2014 Management Studio was also installed with an SQL instance called ‘RUPA-TOSH\SQL Express’ created and three databases ‘esdev’, ‘FingerprintApplication’, and ‘IDCardSystem’ were attached/created (as shown in Figure 36).

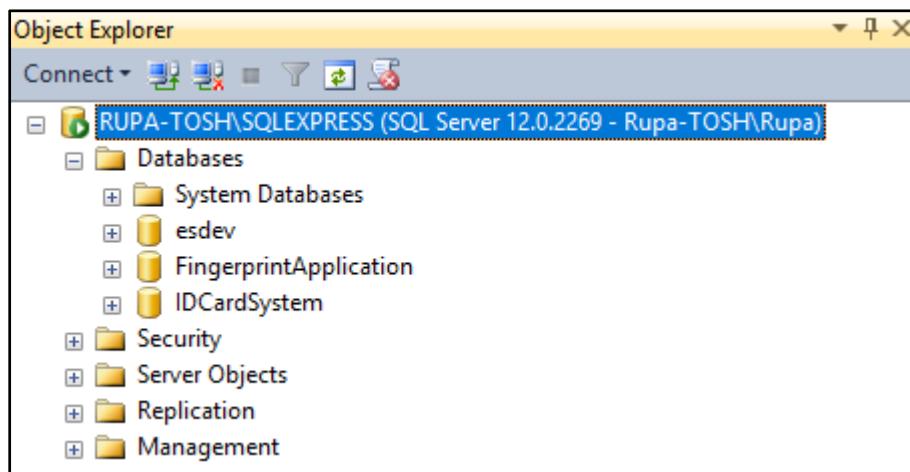


Figure 36 System Databases

4.3.1 The Exams Database

An existing exams database called ‘esdev’ along with its schema was obtained from the University’s development team. The database was then attached to the SQL Express instance. The exams database which is currently used for exams processes consists of 111 tables, 86 views and 19 stored procedures as shown in Figure 37.

The screenshot shows the SQL Query Editor window with the title 'SQLQuery1.sql - R...pa-TOSH\Rupa (52)*'. The code executed is:

```

USE esdev
SELECT count(*) AS Exams_DB_Tables
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE'

SELECT count(*) AS Exams_DB_Views
FROM INFORMATION_SCHEMA.VIEWS

SELECT count(*) AS Exams_DB_Stored_Procedures
FROM sys.objects
WHERE type='P'
    
```

The results pane displays the counts of objects:

Exams_DB_Tables	111
Exams_DB_Views	86
Exams_DB_Stored_Procedures	19

Figure 37 Original Exams Database

Each record in the exams database consists of data relating to an exam, including module number, room number, exam date, start time, exam duration, a class list (in terms of ID number and candidate names), invigilator name, and module information, etc. The fields that are relevant and utilised in this project include StudentId, ExamRequirementId, CandidateNumber, ExaminationId, ExamRoomLocationId, Hostkey, NAME, StartDateTime, Description and Numeric1. As shown in Figure 35, StudentId in the esdev database has been used as the foreign key to build links to the IDCardsystem and FingerprintApplication databases. As mentioned in section 4.2, the class list has been used to reduce the population size of fingerprint identification process for each exam candidate.

The following tables and views were used to create reports for Exams Office staff:

- esdev.rdreader.V_Student – This view consists of student details such as Student ID number, candidate number, email address and other personal details. This table was used to retrieve student ID number and candidate number.
- esdev.rdreader.V_ExamAllocation – This view consists of Student ID number, Candidate Number, Exam Requirement ID and Location ID. This view was used to retrieve student ID number to then match with student ID number in V_Student where the Exam Requirement ID is the same as Exam Requirement ID in V_Examination.
- esdev.rdreader.V_Examination – This consists of details relevant to the actual exam i.e. module code, module description, exam start date, duration etc. This view was used to obtain the exam requirement ID for a particular exam module on a particular date.

4.3.2 Fingerprint Application Database

The ‘FingerprintApplication’ database is the main database which has been created with 6 tables as shown in Figure 38.

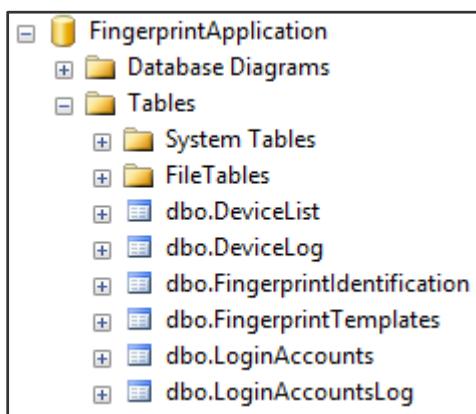


Figure 38 Fingerprint Application Database

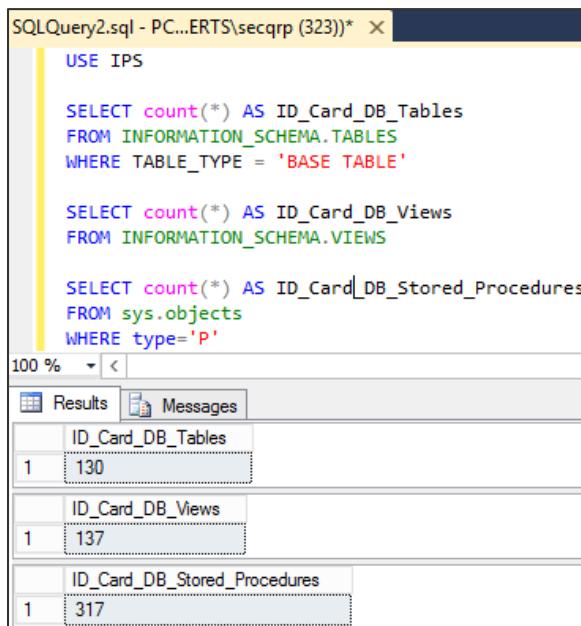
All tables are used as follow:

- dbo.DeviceList – Records details about fingerprint scanners: device ID (primary key), device name, type (whether wall reader or USB), location, network status (whether offline or online), IP address (for online readers only) and Gateway (for online readers only).
- dbo.DeviceLog – Records operation details of each fingerprint scanner i.e. when a reader goes offline or online.

- dbo.FingerprintIdentification – Records details of each fingerprint identification that has been carried out, including member number (student ID number), enrolment ID, live quality score, device ID, matching score and identification date/time when a student is identified on the fingerprint system.
- dbo.FingerprintTemplates – Records each fingerprint template with member number, finger ID (e.g. '2' for right index finger, '7' for left index finger) and date and time when the fingerprint was enrolled. Here enrolment ID is defined as the primary key as the same student can enrol multiple fingers.
- dbo.LoginAccounts – Records details of system operators for exams reporting system. These include username, password, role (Admin, Exams or Invigilator), First Name and Last Name.
- dbo.LoginAccountsLog – Records exams reporting system login activities. It records the LoginID which is the foreign key from the 'LoginAccounts' table, time user logged in and logged out.

4.3.3 The ID Card Database

For the convenience of local project development, a copy of the ID card database structure has been obtained from the University's ID server and attached to the local database instance. This database consists of 130 tables, 137 views and 317 stored procedures as shown in Figure 39.



```
SQLQuery2.sql - PC...ERTS\secqrp (323)* X
USE IPS

SELECT count(*) AS ID_Card_DB_Tables
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE'

SELECT count(*) AS ID_Card_DB_Views
FROM INFORMATION_SCHEMA.VIEWS

SELECT count(*) AS ID_Card_DB_Stored_Procedures
FROM sys.objects
WHERE type='P'
100 % < >
Results Messages
ID_Card_DB_Tables
1 130
ID_Card_DB_Views
1 137
ID_Card_DB_Stored_Procedures
1 317
```

Figure 39 ID Card Database

Only one table was copied to the local instance which has been used to retrieve details of a student:

- dbo.IPSUsers – This table consists of ID number, first name, last name, card serial number, link to the student photograph and other fields relevant to an ID card record for students/staff. Details on how the table is used for the fingerprint application system can be found in Section 4.4.

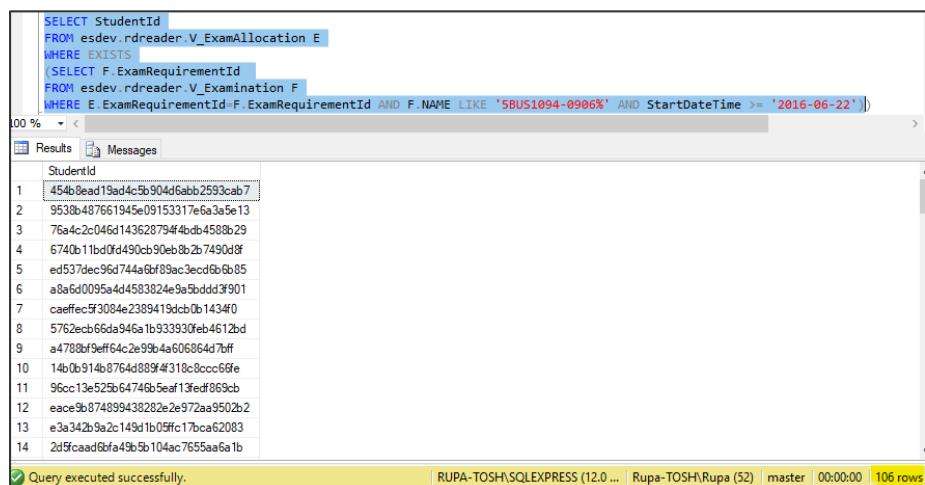
4.3.4 Database Queries

As shown in Figure 35, all three databases are linked by MemberNo (Student Number). The rest of this subsection expands on the queries written to link the three databases.

To generate four reports for exams office, eight SQL queries have been written:

- For Exam Marker:
 1. Query to retrieve list of students scheduled to take exam for a particular module (e.g. '5BUS1094-0906') during a particular exam period (e.g. referred/deferred exams on or after '2016-06-22'). Note that the exam of a module often take place in multiple rooms. The exam marker(s) must be able to access all the exam candidates for his/her module.

When the two queries as shown in Appendix V were run, 106 rows were returned as shown in Figure 40.



StudentId
454b8ead19ad4c5b904d6abb2593cab7
953b487661945e091533176a3a5e13
76a4c2c046d143628794f4bd458b829
6740b1bd0fd490cb90eb8c2b7490d8
ed537dec96d744a6bf89ac3ecd6b6b85
a8a6d0095a4d4583024e9a5bdd3f901
caefeecef3084e2389419dcbb0b1434f0
5762ecb66da946a1b93330feba4612bd
a4788cf9eff64c2e99a4606864d7ff
14b0b914b8764d8894f318c8cc56fe
96cc13e525b64746b5eaf13edf869cb
eacc9b87489943822e2e972aa9502b2
e3a342b9a2c149d1b05ffc17bca62083
2d5fcfaad6fa49b5b104ac7655aa6a1b

Figure 40 Sub Query to Retrieve Student ID Numbers of Students Taking Scheduled Exam

Finally, the top level of the query selects the ID number and candidate number of the expected candidates and insert the results into a table called '#TempTableList' (as shown in Figure 41).

```

SELECT HostKey AS ID_NUMBER, Numeric1 AS Candidate_Number
INTO #TempTableList
FROM esdev.rdreader.V_Student
WHERE Id IN (
    SELECT StudentId
    FROM esdev.rdreader.V_ExamAllocation E
    WHERE EXISTS
        (SELECT F.ExamRequirementId
        FROM esdev.rdreader.V_Examination F
        WHERE E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '5BUS1094-0906%' AND StartDateTime >= '2016-06-22'))

```

Results

ID_NUMBER	Candidate_Number
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	

Query executed successfully.

Figure 41 Query to Retrieve List of Students Taking Exam

2. Query to retrieve list of students who have taken the relevant exam (i.e. '5BUS1094-0906') in the relevant location (i.e. 'Club DH').

When the last two queries are run as shown in Appendix V, 106 rows are returned as shown in Figure 42.

```

SELECT StudentId, ExamRoomLocationId
FROM esdev.rdreader.V_ExamAllocation E
WHERE EXISTS
    (SELECT F.ExamRequirementId
    FROM esdev.rdreader.V_Examination F
    WHERE E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '5BUS1094-0906%' AND StartDateTime >= '2016-06-22'))

```

Results

StudentId	ExamRoomLocationId
45b8ead19ad4c5b904d6abb2593cab7	fa1028876093468db9f79e420bbe2053
953b487661945e09153317e6a3a5e13	fa1028876093468db9f79e420bbe2053
76a4c2c046d143628794f4bd458b29	fa1028876093468db9f79e420bbe2053
6740c11b0f4d490cb90eb8b2b7490d8f	fa1028876093468db9f79e420bbe2053
ed537dec96d744a6fb89ac3ecd6b6b85	fa1028876093468db9f79e420bbe2053
a8a6d0095a4dd4583824e9a5dddf3f001	fa1028876093468db9f79e420bbe2053
caeffec5f3084e2389419dcba1434f0	fa1028876093468db9f79e420bbe2053
5762ecb66da946a1b933930eb4612bd	fa1028876093468db9f79e420bbe2053
a4798b9ef64c2e99a4606064d7bf	1cb60225199346a9b631c75d7c6c...
140b0914b8764d8894f318c0cc6f6	fa1028876093468db9f79e420bbe2053
96cc13e525b64746b5eaef13fed869cb	fa1028876093468db9f79e420bbe2053
eace9b874899438282e2e972aa9502b2	fa1028876093468db9f79e420bbe2053
e3a342b9a2c149d1b05ff17bca62083	fa1028876093468db9f79e420bbe2053
2d95caad6fb49b5b104ac7655aa6a1b	fa1028876093468db9f79e420bbe2053
2eebad22d8a34f5b8c5bb29039794822	fa1028876093468db9f79e420bbe2053
db5be8d469d4d6a7bb3c23d9c22	fa1028876093468db9f79e420bbe2053
e0dd9597f324bf4aa48ea5427b26c5a	fa1028876093468db9f79e420bbe2053
a591a5e4efb4d4d982e1fd75a7e8116	fa1028876093468db9f79e420bbe2053
7a9c24a6609349719a413d8743f2cc9	fa1028876093468db9f79e420bbe2053
a53956290124ca29264a71a1cdf08c	fa1028876093468db9f79e420bbe2053

Query executed successfully.

Figure 42 Query Results Showing Student ID Numbers and Corresponding Exam Room Location

The third subquery returns the Device ID from the Fingerprint Application database where the location name of the scheduled room is the same as where the exam has taken place as shown in Figure 43.

```

SELECT DeviceID
FROM FingerprintApplication.dbo.DeviceList d INNER JOIN esdev.rreader.V_Location l
ON d.DeviceLocation=l.Description
WHERE EXISTS(
    SELECT StudentId, ExamRoomLocationId
    FROM esdev.rreader.V_ExamAllocation E
    WHERE EXISTS
        (SELECT F.ExamRequirementId
        FROM esdev.rreader.V_Examination F
        WHERE E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '5BUS1094-0906%' AND StartDateTime >= '2016-06-22'))))
```

The screenshot shows a SQL query being run in SSMS. The results pane displays a single row with DeviceID 293569866-16075008729. The status bar at the bottom indicates the query executed successfully.

Figure 43 Query Results showing Device ID of Exam Room

The second subquery returns student numbers of those who have taken exams in the queried location for the queried module (i.e. 5BUS1094-0906):

```

SELECT DISTINCT MemberNo
FROM FingerprintApplication.dbo.FingerprintIdentification
WHERE DeviceId IN(
    SELECT DeviceID
    FROM FingerprintApplication.dbo.DeviceList d INNER JOIN esdev.rreader.V_Location l
    ON d.DeviceLocation=l.Description
    WHERE EXISTS(
        SELECT StudentId, ExamRoomLocationId
        FROM esdev.rreader.V_ExamAllocation E
        WHERE EXISTS
            (SELECT F.ExamRequirementId
            FROM esdev.rreader.V_Examination F
            WHERE E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '5BUS1094-0906%' AND StartDateTime >= '2016-06-22'))))
```

The screenshot shows a SQL query being run in SSMS. The results pane displays student numbers from 1 to 9. The status bar at the bottom indicates the query executed successfully.

Figure 44 Query Results showing Student Numbers of Students who have Taken Exam

When the full query is executed, the Student Number, First Name and Last Name of the students who took the exam is then retrieved from the ID card database and inserted into a temporary table called '#TempTable'.

3. Query to retrieve list of absent students – those who didn't turn up for the exam.

Figure 45 shows an example of the results of the query run (as shown in Appendix V), which will then be used to generate a relevant exam report.

The screenshot shows a SQL query window with the following code:

```

SELECT Candidate_Number FROM #TempTableList WHERE
ID_Number NOT IN(SELECT ID_NUMBER FROM #TempTable)

```

The results pane displays a table with one column, "Candidate_Number", containing values from 1 to 12. The first row (value 1) has its content redacted.

At the bottom of the window, the status bar shows: OS\SQLEXPRESS (12.0 ... | Rupa-TOSH\Rupa (52) | master | 00:00:00 | 105 rows

Figure 45 Query Results showing Absentee Students

4. Final query to retrieve list of extra students – those who have not been scheduled to take the exam in the queried location but did end up taking the exam in that room/location.

Figure 46 shows an example of the results of the query run as shown in Appendix V, which will then be used to generate a relevant exam report.

The screenshot shows a SQL query window with the following code:

```

SELECT DISTINCT Numeric1 AS CANDIDATE_NUMBER
INTO #TempTableC
FROM esdev.dbo.V_Student
WHERE HostKey IN(
SELECT ID_Number
FROM #TempTable
WHERE EXISTS(
SELECT DISTINCT Numeric1, HostKey
FROM esdev.dbo.V_Student
WHERE HostKey=ID_Number))
select * From #TempTableC

```

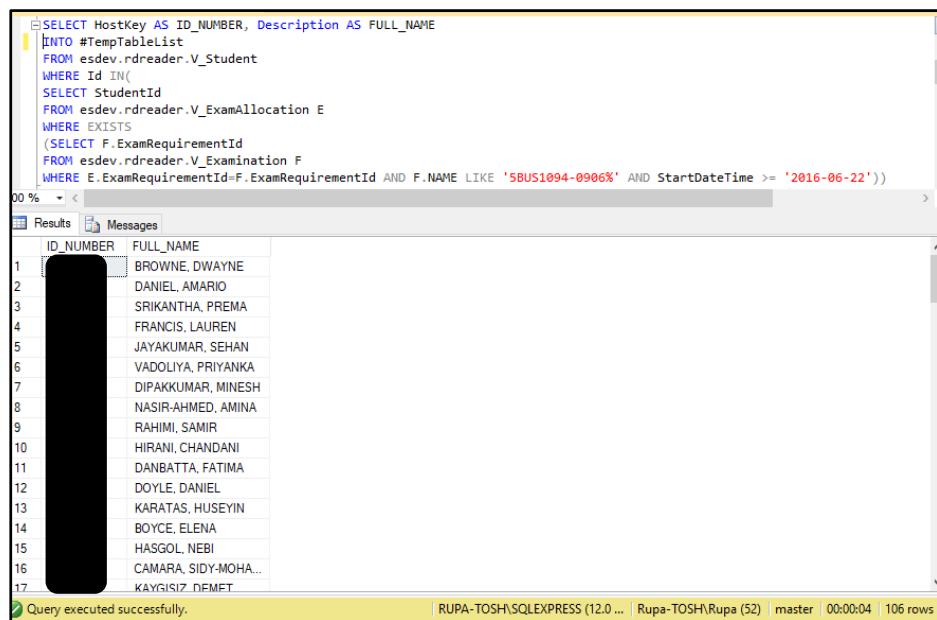
The results pane displays a table with one column, "CANDIDATE_NUMBER", containing values from 1 to 9. The first row (value 1) has its content redacted.

At the bottom of the window, the status bar shows: -TOSH\SQLEXPRESS (12.0 ... | Rupa-TOSH\Rupa (52) | master | 00:00:00 | 9 rows

Figure 46 Query Results for Extra Students

- For Exams Office: Same as the queries run for Exam Markers but instead of retrieving candidate numbers, the following four queries return Student Numbers and Name.
 1. Query to retrieve list of students scheduled to take exam for a particular module (e.g. '5BUS1094-0906') during a particular exam period (e.g. referred/deferred exams on or after '2016-06-22').

Figure 47 shows the results when the query shown in Appendix V is executed.



```

SELECT HostKey AS ID_NUMBER, Description AS FULL_NAME
INTO #TempTableList
FROM esdev.rdreader.V_Student
WHERE Id IN(
SELECT StudentId
FROM esdev.rdreader.V_ExamAllocation E
WHERE EXISTS
(SELECT F.ExamRequirementId
FROM esdev.rdreader.V_Examination F
WHERE E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '5BUS1094-0906%' AND StartDateTime >= '2016-06-22'))

```

The screenshot shows a SQL query being run in SSMS. The query selects student IDs and names from a temporary table based on exam requirements. The results grid shows 106 rows of student names. The status bar at the bottom indicates the query was executed successfully.

ID_NUMBER	FULL_NAME
1	BROWNE, DWAYNE
2	DANIEL, AMARIO
3	SRIKANTHA, PREMA
4	FRANCIS, LAUREN
5	JAYAKUMAR, SEHAN
6	VADOLIYA, PRIYANKA
7	DIPAKKUMAR, MINESH
8	NASIR-AHMED, AMINA
9	RAHIMI, SAMIR
10	HIRANI, CHANDANI
11	DANBATTI, FATIMA
12	DOYLE, DANIEL
13	KARATAS, HUSEYIN
14	BOYCE, ELENA
15	HASGOL, NEBI
16	CAMARA, SIDY-MOHA...
17	KAYGISIZ, DEMET

Query executed successfully. | RUPA-TOSH\SQLEXPRESS (12.0 ... | Rupa-TOSH\Rupa (52) | master | 00:00:04 | 106 rows

Figure 47 Query Results - Student Numbers and Names of Students Scheduled for an Exam

2. Query to retrieve list of students who have taken the relevant exam (i.e. '5BUS1094-0906') in the relevant location (i.e. 'Club DH').

Figure 48 shows rows returned when the query shown in Appendix V is executed.

The screenshot shows a SQL query window with the following code:

```

SELECT ADLogonName AS ID_Number, FirstName AS First_Name, LastName AS Last_Name
INTO #TempTable
FROM IDCardSystem.dbo.IPSUsers
WHERE ADLogonName IN(
    SELECT DISTINCT MemberNo
    FROM FingerprintApplication.dbo.FingerprintIdentification
    WHERE DeviceId IN(
        SELECT DeviceID
        FROM FingerprintApplication.dbo.DeviceList d INNER JOIN esdev.rdreader.V_Location l
        ON d.DeviceLocation=l.Description
        WHERE EXISTS(
            SELECT StudentId, ExamRoomLocationId
            FROM esdev.rdreader.V_ExamAllocation E
            WHERE EXISTS(
                SELECT F.ExamRequirementId
                FROM esdev.rdreader.V_Examination F
                WHERE E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '5BUS1094-0906%' AND StartDateTime >= '2016-06-22')))
)

```

The results grid displays 10 rows of student information:

ID_Number	First_Name	Last_Name
1	Asad	Kamal
2	Vishal	Dinescumar
3	Yiran	Yang
4	Z	ChenJiajie
5	Shehroze	Khan
6	Elliott	Cable
7	Yuting	Liu
8	Haotian	Wu
9	Said	Ettahery
10	Test	Record

Message bar at the bottom: Query executed successfully.

Figure 48 Query Results Showing List of Student Taken Exam in a Room

3. Query to retrieve list of absent students – those who didn't turn up for the exam.

Figure 49 shows records retrieved when query shown in Appendix V is executed.

The screenshot shows a SQL query window with the following code:

```

SELECT * FROM #TempTableList WHERE
ID_Number NOT IN(SELECT ID_NUMBER FROM #TempTable)

```

The results grid displays 105 rows of student information:

ID NUMBER	FULL_NAME
1	JAYAKUMAR, SEHAN
2	FRANCIS, LAUREN
3	SRIKANTHA, PREMA
4	DANIEL, AMARIO
5	HASGOL, NEBI
6	BOYCE, ELENA
7	VADOLIYA, PRIYANKA
8	RAHIMI, SAMIR
9	DOYLE, DANIEL
10	DIPAKKUMAR, MINESH
11	NASIR-AHMED, AMINA
12	HIRANI, CHANDANI
13	KAYGISIZ, DEMET
14	OZALTUN, ERDEM
15	CAMARA, SIDY-MOHA...
16	DANBATTI, FATIMA
17	KARATAS, HUSEYIN
18	TESORO, GLENN
19	SHAMSI, FAIZAN
20	BAKHTIAR, SANA
21	HENRY, LEAH
22	LIBORO, LEE
23	PATEL, ANKIT
...	...

Message bar at the bottom: Query executed successfully.

Figure 49 Query Results for List of Absentees for Exams Office

4. Final query to retrieve list of extra students – those who weren't scheduled to take the exam in the queried location but did end up taking the exam in that room/location.

Figure 50 shows results retrieved when the query shown in Appendix V is executed.

```

SELECT * FROM #TempTable
WHERE ID_NUMBER NOT IN(
SELECT ID_NUMBER FROM #TempTableList)

```

ID_Number	First_Name	Last_Name
1	Asad	Kamal
2	Vishal	Dinescumar
3	Yiran	Yang
4	Z	Chen-Jiajie
5	Shehroze	Khan
6	Elliott	Cable
7	Yuting	Liu
8	Haotian	Wu
9	Said	Ettahery

TOSH\SQLEXPRESS (12.0 ... | Rupa-TOSH\Rupa (52) | master | 00:00:00 | 9 rows

Figure 50 Query Results for List of Extra Students for Exams Office

Section 4.5 includes details and code in which all these queries have been used to create customised reports for both the marker and the Exams office.

4.4 Fingerprint Application Software Development Kit/Programming

For the Fingerprint Application, a Software Development Kit (SDK) [57] provided by Morpho, Safran was used to further develop and customise the fingerprint application to meet the UH requirements. This subsection covers the design and programming of this application.

The SDK and identification license dongle and a USB Fingervein Reader was purchased following a successful funding of £1,368 acquired from Diamond Fund Award in March 2016.

Once the SDK, the license dongle and the USB Fingervein Enrolment Reader arrived, Microsoft Visual Studio 2017 was installed to allow amendment of the files within the SDK and to create a new application for the exam reporting system (as detailed in Section 4.5). The scanner drivers and license were installed as shown in Figure 51.

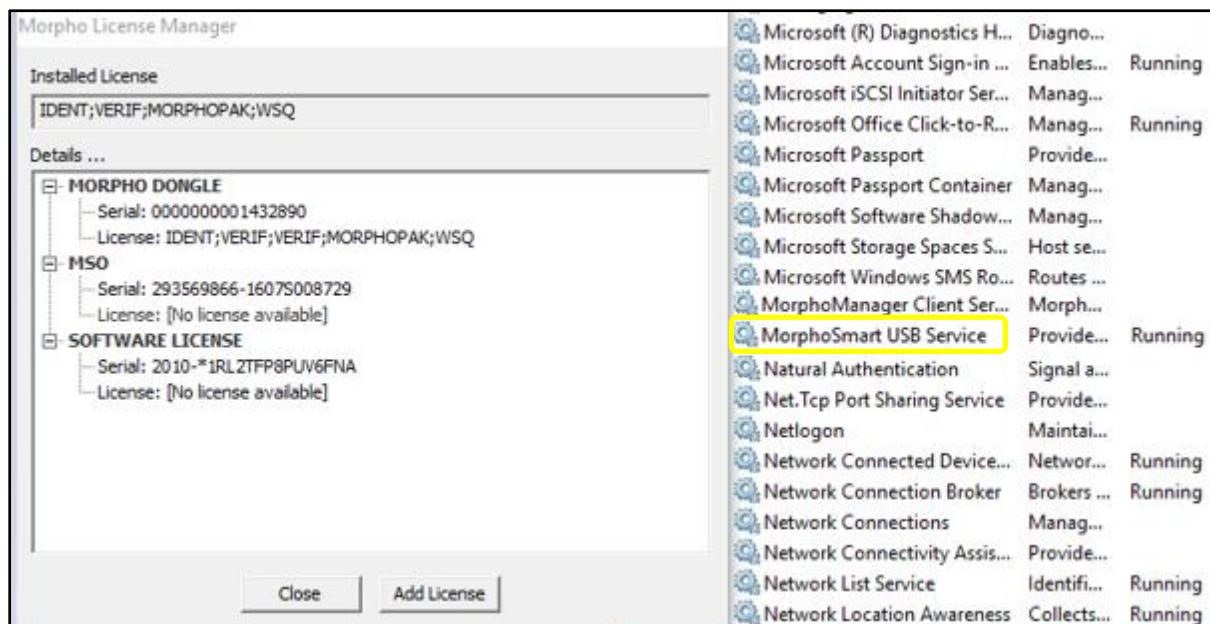


Figure 51 Morpho License and USB Scanner Service

The SDK comes with 6 sample fingerprint/fingervein applications as shown in Appendix VI. Following some comparison tests, it was decided that the 'MorphoKitDotNETSample' application will be used for this project. This is because the application had the identification functionality required for this project.

4.4.1 Fingerprint Application Design

The MorphoKitDotNETSample's design was amended as shown in Figure 52, Figure 53 and Figure 54 where changes have been circled in blue.

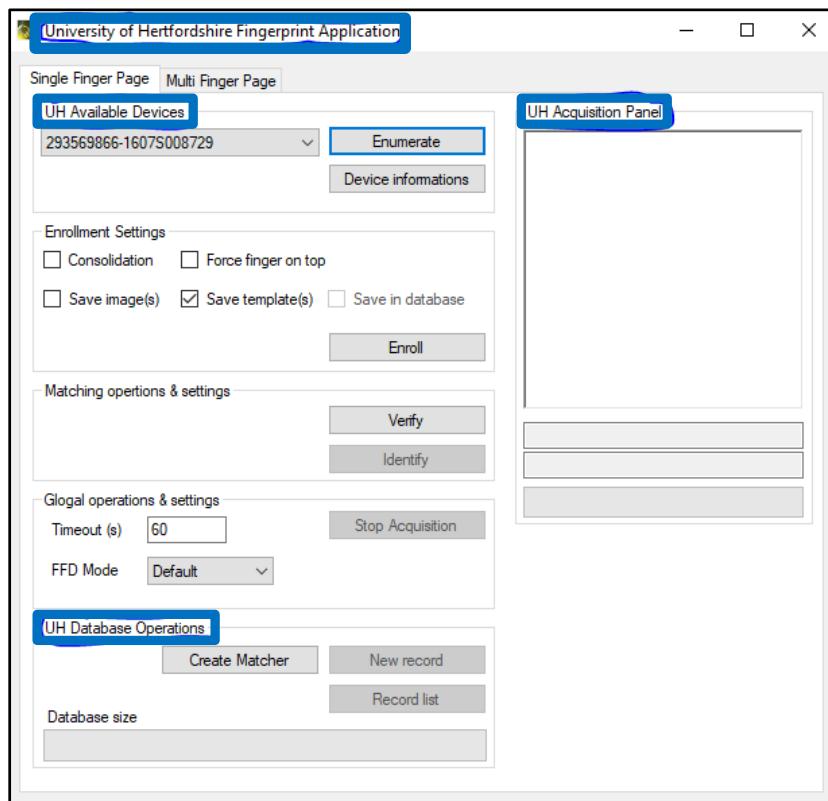


Figure 52 Fingerprint Application Main Form Amended

New Record

Record Id	22
ID Number	[REDACTED]

The record id is required. ID Number is used as payload.

Templates

Finger position	RIGHT_INDEX	+ -
Finger Position	File path	
RIGHT_INDEX	C:\Users\Rupa\Documents\EngD\...	

Save Cancel

Figure 53 Fingerprint Application Amended New Record Form

Record List

Record id	ID Number	Templates
1	[REDACTED]	1
10	[REDACTED]	1
11	[REDACTED]	1
20	[REDACTED]	1
21	[REDACTED]	1
24	[REDACTED]	1
41	[REDACTED]	1
49	[REDACTED]	1
5	[REDACTED]	1

Select a record id to remove Remove Refresh

Available records in database

Export Import Close

Figure 54 Fingerprint Application Amended Record List Form

The forms amended are:

- 'MainForm.Designer.cs'

```
this.Text = "University of Hertfordshire Fingerprint Application";
this.groupBox3.Text = "UH Acquisition Panel";
this.groupBox6.Text = "UH Database Operations";
this.groupBox9.Text = "UH Available Devices";
```

- 'NewRecord.Designer.cs'

```
this.label3.Text = "ID Number"; //Was "Full Name". Changed on 27/01/2018
```

```

this.label5.Text = "The record id is required. ID Number is used as payload .";
//Was "The record id is required. Both first and last name are used as payload
." Changed on 27/01/2018

```

- 'RecordList.Designer.cs'

```

this.columnHeader3.Text = "ID Number"; //Was "Payload", changed to "ID Number"
on 27/01/2018

```

4.4.2 Fingerprint Application Programming

Classes/files have been amended using C# programming language with SQL code to connect to the Fingerprint Application database.

4.4.2.1 Adding Fingerprint Templates into SQL Database – RecordList.cs

Before modifying the code on the Record List form (Figure 54), the application stored all fingerprints captured as a cvf template in an XML file only:

```

private void btn_db_export_Click(object sender, EventArgs e)
{
    SaveFileDialog fileDlg = new SaveFileDialog();
    fileDlg.AddExtension = true;

    fileDlg.FileName = "matcher_records";
    fileDlg.Title = "Export database records";
    fileDlg.DefaultExt = "xml";
    fileDlg.Filter = "XML file (*.xml)|*.xml|All Files (*.*)|*.*";
    if (fileDlg.ShowDialog() == DialogResult.OK)
    {
        try
        {
            XmlSerializer serializer = new XmlSerializer(typeof(DatabaseXml));
            StreamWriter writer = new StreamWriter(fileDlg.FileName);

            DatabaseXml db = new DatabaseXml();
            List<RecordXml> records = new List<RecordXml>();
            string[] ids = _matchingContext.GetRecordIds();
            foreach (string id in ids)
            {
                RecordXml record = new RecordXml();
                IRecord irecord = _matchingContext.FindRecord(id);
                record.Id = irecord.Id;
                record.Payload = Encoding.ASCII.GetString(irecord.Payload);
                List<TemplateXml> templates = new
List<TemplateXml>(irecord.NumberOfTemplates);
                for (int i = 0; i < irecord.NumberOfTemplates; ++i)
                {
                    TemplateXml template = new TemplateXml();
                    IFingerTemplate fingertemplate = irecord.GetTemplate(i);
                    template.FingerId = fingertemplate.Id;
                    template.Data = fingertemplate.Buffer;
                    templates.Add(template);
                }
                record.TemplateItems = templates.ToArray();
                records.Add(record);
            }
            db.RecordList = records.ToArray();
            serializer.Serialize(writer, db);
            writer.Close();
        }
        catch (Exception exc)
        {

```

```

        MessageBox.Show(exc.Message, "Export Database",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

The code was amended within the ‘RecordList.cs’ file to save the templates in the FingerprintTemplates table in the FingerprintApplication database:

```

SqlConnection sqlConnection1 = new SqlConnection("Data Source=RUPA-
TOSH\SQLEXPRESS;Initial Catalog=FingerprintApplication;Integrated
Security=True");
SqlCommand cmd = new SqlCommand();
Int32 rowsAffected;

cmd.CommandText = "SP_ImportFingerprintData_XML";
cmd.CommandType = CommandType.StoredProcedure;
cmd.Connection = sqlConnection1;

sqlConnection1.Open();

rowsAffected = cmd.ExecuteNonQuery();

sqlConnection1.Close();

```

The ‘SP_ImportFingerprintData_XML’ procedure has been written to import templates from XML file into the FingerprintApplication database:

```

USE [FingerprintApplication]
GO
***** Object: StoredProcedure [dbo].[SP_ImportFingerprintData_XML] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[SP_ImportFingerprintData_XML]
AS
BEGIN
    -- Creating a Temporary Table for importing the data from xml file.
    CREATE TABLE #Fingerprints(
        [RecordID] [varchar](50) NOT NULL,
        [MemberNo] [varchar](50) NOT NULL,
        [FingerId] [varchar](50) NULL,
        [FingerprintData] [varchar](5000) NULL,
        [EnrolmentDate] [datetime] NULL,
        CONSTRAINT [PK_Fingerprints1] PRIMARY KEY CLUSTERED ( [RecordID] ASC )
    )

    -- Inserting all the rows from xml to temporary table using OPENROWSET
    ;with XMLNAMESPACES (DEFAULT 'http://www.morpho.com')
    INSERT INTO
    #Fingerprints(RecordId,MemberNo,FingerId,FingerprintData,EnrolmentDate)

    SELECT record.value('@Id','varchar(max)') AS RecordId,
           X.record.query('Payload').value('.','varchar(50)') AS MemberNo,
           data.value('@FingerId','varchar(50)') AS FingerprintId,
           X.record.query('TemplateList/Template/Data').value('.','varchar(5000)') AS
FingerprintData,
           getdate() AS EnrolmentDate
    FROM (
    SELECT CAST(x AS XML)
    FROM OPENROWSET(
        BULK 'C:\Users\Rupa\Documents\EngD\Thesis\Test_FPs\matcher_records.xml',

```

```

    SINGLE_BLOB) AS T(x)
) AS T(x)

CROSS APPLY x.nodes('MorphoKitDatabase/RecordList/Record') AS X(record)
CROSS APPLY X.record.nodes('TemplateList/Template') AS Y(data)

    -- Selecting the records from temporary table. This is just to know the records
    inserted or not.
    --SELECT * FROM #Fingerprints;

    -- By using MERGE statement, inserting the record if not present and updating if
exists.
    MERGE FingerprintTemplates AS TargetTable      -- Inserting or Updating the table.
    USING #Fingerprints AS SourceTable            -- Records from the temporary table
(records from xml file).
    ON (TargetTable.RecordId = SourceTable.RecordId) -- Defining condition to decide
which records are already present
    WHEN NOT MATCHED BY TARGET                  -- If the records in the
FingerprintTemplates table is not matched?
        THEN INSERT (RecordId, MemberNo, FingerId, FingerprintData, EnrolmentDate)
-- then INSERT the record
        VALUES(SourceTable.RecordId, SourceTable.MemberNo, SourceTable.FingerId,
SourceTable.FingerprintData, SourceTable.EnrolmentDate)
    WHEN MATCHED                                -- If not matched then UPDATE
        THEN UPDATE SET
            TargetTable.RecordId = SourceTable.RecordId,
            TargetTable.MemberNo = SourceTable.MemberNo,
            TargetTable.FingerId = SourceTable.FingerId,
            TargetTable.FingerprintData = SourceTable.FingerprintData,
            TargetTable.EnrolmentDate = SourceTable.EnrolmentDate;

    SELECT * FROM FingerprintTemplates;
END

```

To allow execution of queries, the following libraries were also added at the top of the file:

```

using System.Data.SqlClient;
using System.Data.OleDb;
using System.Data.SqlClient;

```

Full code has been added as Appendix VII.

4.4.4.2 Identifying registered template – MSOAcquisitionStrategy.cs

As shown in Figure 55, the default application only showed the Candidate ID (which is Record ID) and Matching Score. The MSOAcquisitionStrategy.cs file was amended to add code to connect to database and display Member Number, First Name and Last Name which is retrieved from the ID card database based on the ID number match. Figure 56 shows an example of this.

To allow execution of queries, the following libraries were also added at the top of the file:

```

using System.Data.SqlClient;
using System.Data.OleDb;
using System.Data.SqlClient;

```

The below code which has been added to the existing code, first makes connection to two databases: FingerprintApplication to obtain Member Number for those whose Record ID is the same as Candidate ID. Once this has been retrieved, a connection to the IDCardSystem database is opened to then retrieve the First Name and Last Name from the IPSUsers table

where the Member Number from the Fingerprint Application matches the ADLogonName in the IDCardSystem database. If a record is found, the Candidate ID, Matching Score, Member Number, First Name and Last Name is displayed in a message box. If no record is retrieved, a “User not found” message is displayed.

```

if (candidate.Score > 2000) //if matching score is greater than 2000
{
    System.Windows.Forms.Form f =
System.Windows.Forms.Application.OpenForms[" MainForm"];
    SqlConnection con = new SqlConnection();
    SqlConnection con1 = new SqlConnection();
    con.ConnectionString = "Data Source=RUPA-
TOSH\\SQLEXPRESS;Initial Catalog=FingerprintApplication;Integrated Security=True";
    con1.ConnectionString = "Data Source=RUPA-
TOSH\\SQLEXPRESS;Initial Catalog=IDCardSystem;Integrated Security=True";
    con.Open();
    con1.Open();
    SqlCommand cmd = new SqlCommand("Select MemberNo from
FingerprintTemplates where RecordId=' " + candidate.Id + "' ", con);
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    DataTable dt = new DataTable();
    da.Fill(dt);
    SqlCommand cmd1 = new SqlCommand("Select FirstName, LastName
from IPSUsers where ADLogonName=' " + dt.Rows[0][0].ToString() + "' ", con1);
    SqlDataAdapter da1 = new SqlDataAdapter(cmd1);
    DataTable dt1 = new DataTable();
    da1.Fill(dt1);
    SqlCommand cmd2 = new SqlCommand("Select LastName from
IPSUsers where ADLogonName=' " + dt.Rows[0][0].ToString() + "' ", con1);
    SqlDataAdapter da2 = new SqlDataAdapter(cmd2);
    DataTable dt2 = new DataTable();
    da2.Fill(dt2);
    if (dt.Rows.Count == 1)
        if (dt1.Rows.Count == 1)
            if (dt2.Rows.Count == 1)
                MessageBox.Show(String.Format("Candidate id :
{0}\nMatching Score : {1}\nMember Number : {2}\nFirst Name : {3}\nLast Name : {4}",
candidate.Id, candidate.Score, dt.Rows[0][0].ToString(),
dt1.Rows[0][0].ToString(), dt2.Rows[0][0].ToString()));

```

Once a user has been identified successfully, the code below inserts the identified Member Number along with EnrolmentID, LiveQuality, DeviceID and IdentificationDate (and time) into the FingerprintIdentification table:

```

SqlDataAdapter da3 = new SqlDataAdapter();
da3.InsertCommand = new SqlCommand("INSERT INTO
FingerprintIdentification
(MemberNo,EnrolmentID,LiveQuality,DeviceID,MatchingScore,IdentificationDate)
VALUES(@MemberNo,@EnrolmentID,@LiveQuality,'293569866-
1607S008729',@MatchingScore,@IdentificationDate)", con);
    da3.InsertCommand.Parameters.Add("@MemberNo",
SqlDbType.VarChar).Value = dt.Rows[0][0].ToString();
    da3.InsertCommand.Parameters.Add("@EnrolmentID",
SqlDbType.VarChar).Value = candidate.Id;
    da3.InsertCommand.Parameters.Add("@LiveQuality",
SqlDbType.VarChar).Value = enrollResult.Quality;
    da3.InsertCommand.Parameters.Add("@MatchingScore",
SqlDbType.Int).Value = candidate.Score;
    da3.InsertCommand.Parameters.Add("@IdentificationDate",
SqlDbType.DateTime).Value = DateTime.Now;

    da3.InsertCommand.ExecuteNonQuery();

```

```

        con.Close();
        con1.Close();
    }
    else MessageBox.Show("User not found");
}
else
{
    DisplayStatusMessage(acqresult.Status, "Identify");
}
}

```

Full code from the MSOAcquisitionStrategy.cs file has been added to Appendix VIII.

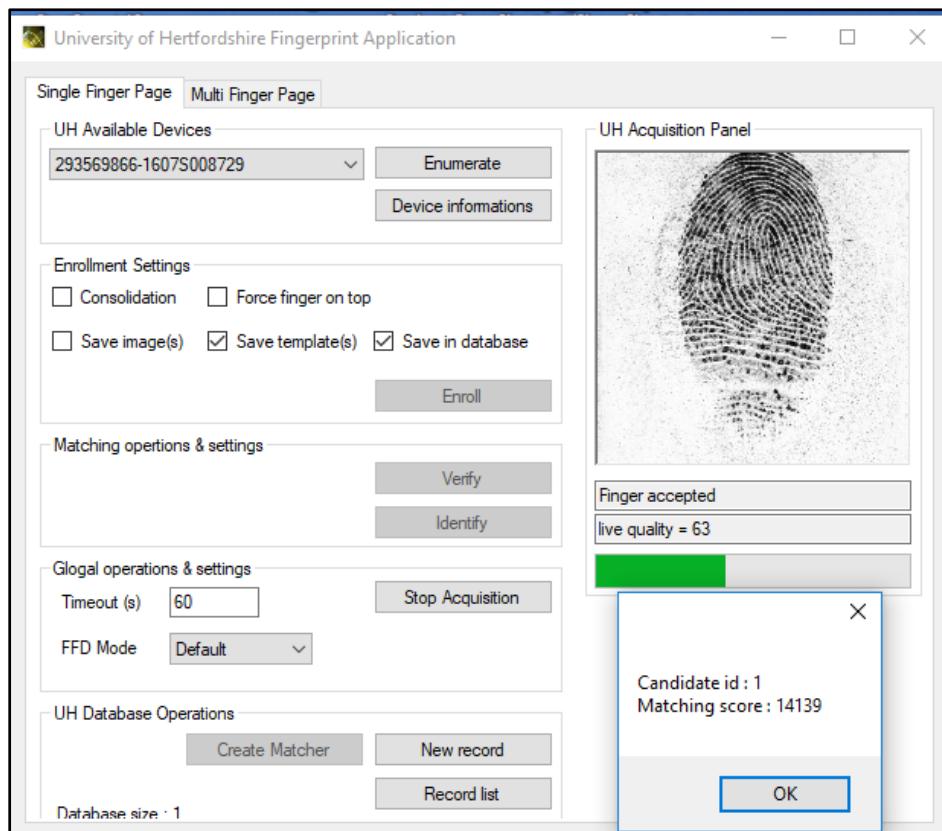


Figure 55 Fingerprint Application Identification Record without Personal Details

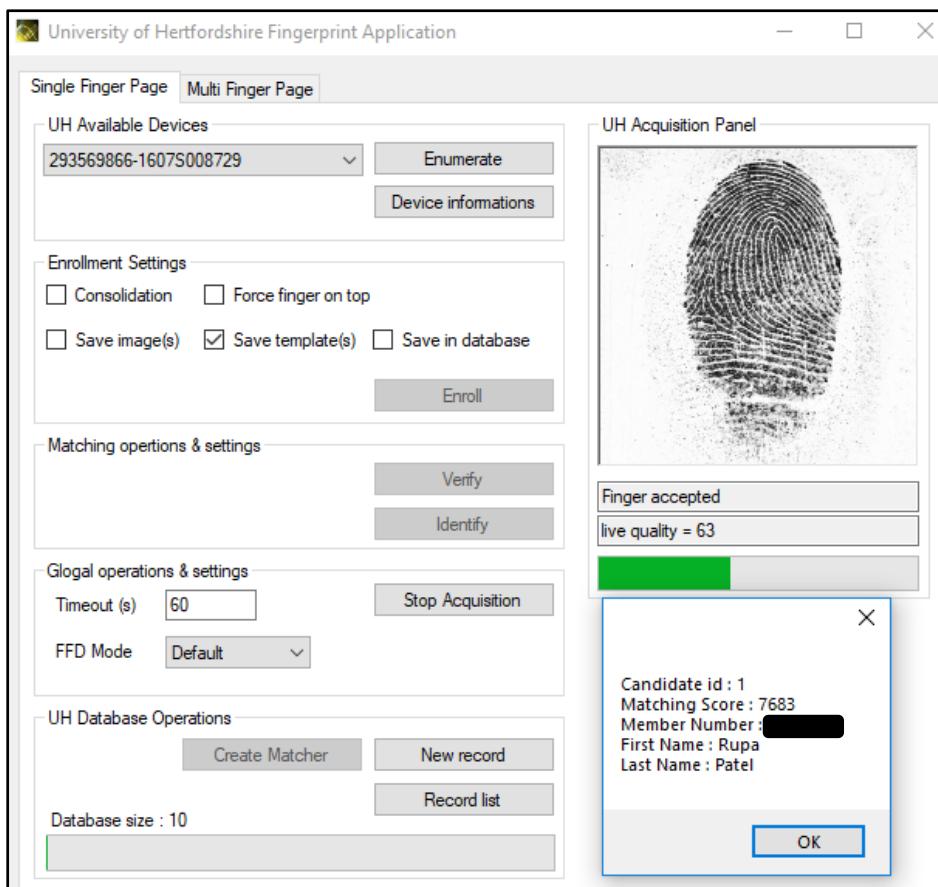


Figure 56 Fingerprint Application Identification Record with Personal Details

4.5 Development of the Integrated Exam Reporting System

A new application called ‘LoginForm’ has been created using Visual Studio with C# as the programming language. This application serves the following purposes:

- Allows ID Office staff to enrol fingerprints against a student record which is linked to the application developed as per Section 4.4.
- Enables an invigilator to check student record including photograph and identify students via linked fingerprint application as detailed in Section 4.4.
- Allows the University’s Exams Office to generate reports.

4.5.1 Design and Functionality

The application consists of 4 user interaction forms and involves 3 databases as shown in the Navigation Diagram (Figure 57):

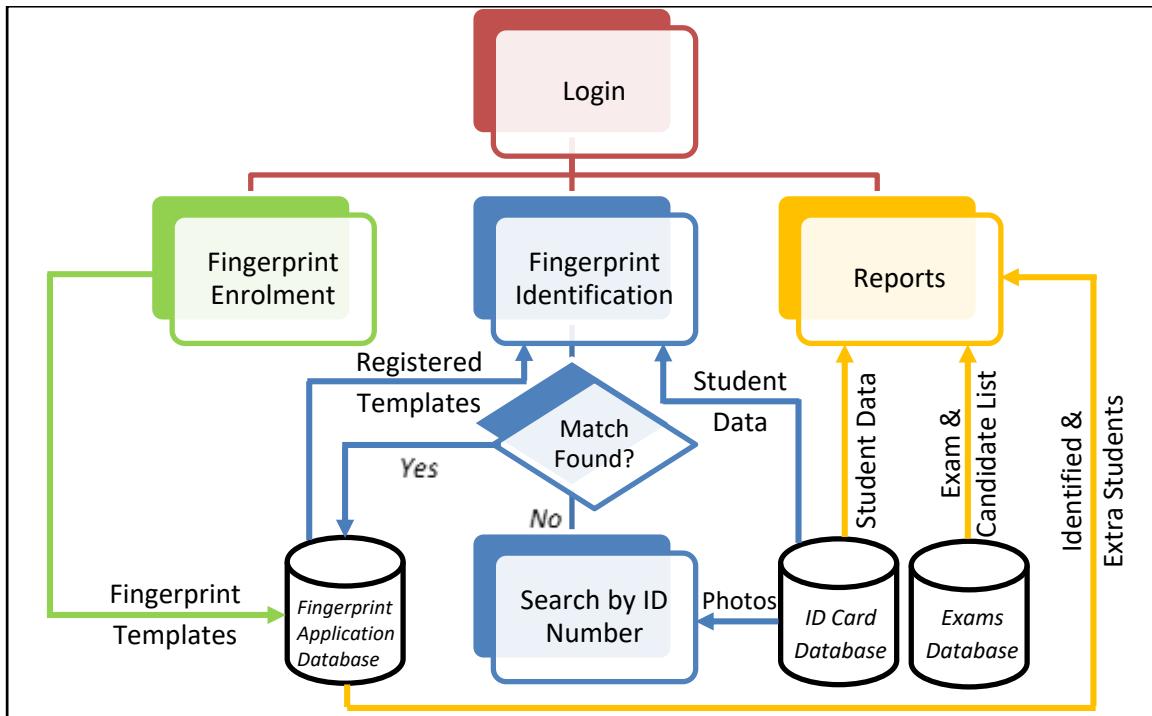


Figure 57 System Navigation Diagram

- `LoginForm.cs` (Figure 58) – Used to login to the system. It has three user roles setup in the `LoginAccounts` table within the `FingerprintApplication` database. These are: ‘Admin’ used by ID Office staff to enrol fingerprints (marked in green in the figure), ‘Invigilator’ used by exam invigilators to search for and identify students (marked in blue in the figure) and ‘Exams’ used by Exams Office staff to generate reports (marked in orange in the figure).

The screenshot shows a Windows-style application window titled 'Login'. In the top right corner are standard window controls: a minus sign for minimize, a square for maximize/minimize, and an X for close. The top left corner features a small icon followed by the word 'Login'. To the right of the title bar is the University of Hertfordshire logo with the letters 'UH'. The main content area is a light gray box with a blue border. At the top center, the text 'Fingerprint Recognition System - Login to continue' is displayed in blue. Below this, there are two input fields: 'Username' on the left with a corresponding text input box to its right, and 'Password' on the left with a corresponding text input box to its right. At the bottom of the input area are two buttons: a blue-outlined 'Login' button on the left and a white 'Cancel' button on the right. At the very bottom of the window, centered, is the text 'Designed by Rupa Patel, ID Office'.

Figure 58 Login Form

- `SearchForm.cs` (Figure 59) – Connects to `IPSUsers` table in the `IDCardSystem` database to retrieve student record and photograph.

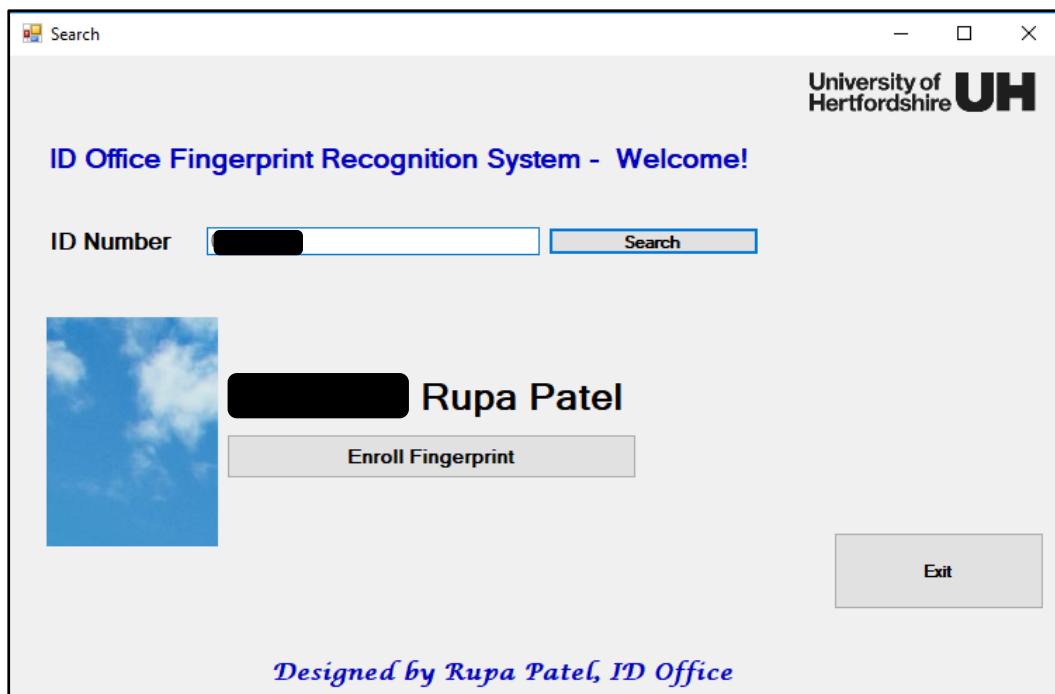


Figure 59 Search Form

Once the 'Enroll Fingerprint' button is pressed, the student sees the screen as shown in Figure 60. A similar message is displayed for the left finger.

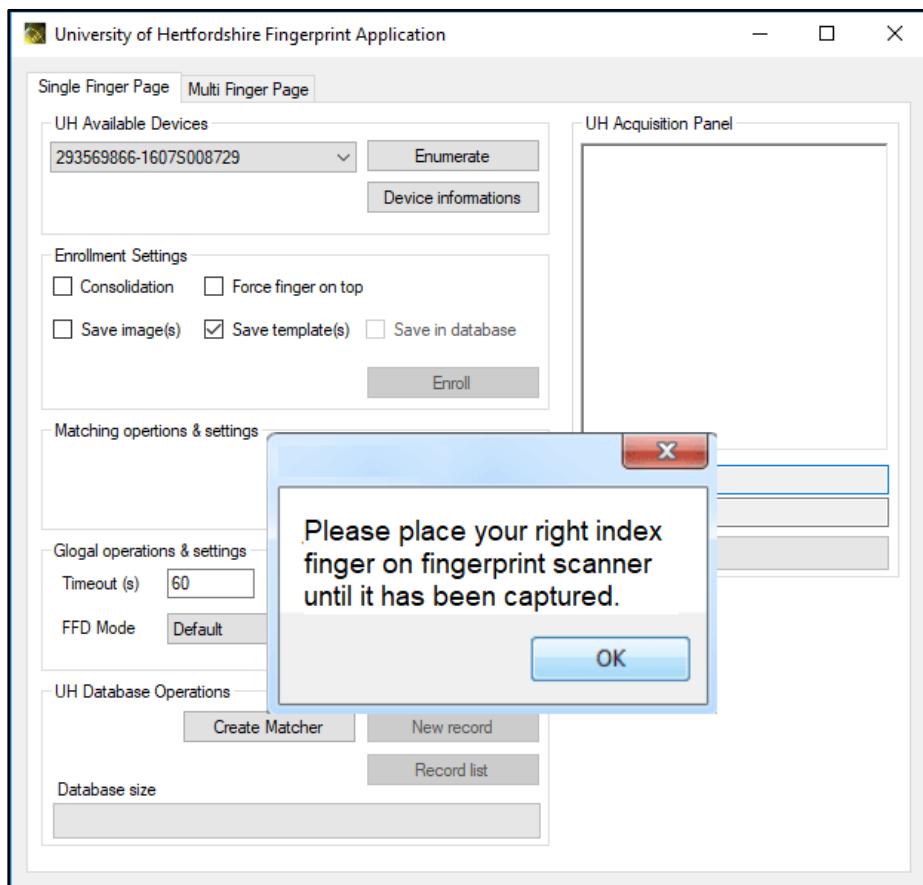


Figure 60 Instructions to user for fingerprint capture

- ExamsOfficeReports.cs – Connects to the exams database ‘esdev’, the ‘FingerprintApplication’ database and the ‘IDCardSystem’ database to retrieve 4 reports, 2 for Exams Office staff: ‘Absentee Students for Exams Office’ (Figure 61) and ‘Extra Students for Exams Office’ (Figure 62) and 2 for Exams Marker: ‘Absentee Students for Marker’ (Figure 63) and ‘Extra Students for Marker’ (Figure 64).

The screenshot shows a Windows application window titled "ExamsOfficeReports". The main title bar says "Exams Office Fingerprint Recognition System - Welcome!". Below the title bar, there are three input fields: "Select report to run" (dropdown menu showing "Absentee Students for Exams Office"), "Enter Exam Code:" (text input field containing "5BUS1094-0906"), and "Enter Date of Exam:" (text input field containing "2016-06-22") with a "Format: YYYY-MM-DD" label next to it. A "Search" button is located below these fields. Below the search area is a table with five rows of data. The table has columns: ID_NUMBER, FULL_NAME, Module_Code, and Exam_Date. The data is as follows:

ID_NUMBER	FULL_NAME	Module_Code	Exam_Date
[REDACTED]	FRANCIS, LAUREN	5BUS1094-0906	2016-06-22
[REDACTED]	SRIKANTHA, PREMA	5BUS1094-0906	2016-06-22
[REDACTED]	HASGOL, NEBI	5BUS1094-0906	2016-06-22
[REDACTED]	VADOLIYA, PRIYANKA	5BUS1094-0906	2016-06-22
[REDACTED]	NASIR-AHMED, AMINA	5BUS1094-0906	2016-06-22

At the bottom of the table is an "Export" button.

Figure 61 Absentee Students Report for Exams Office Report

The screenshot shows the same application window as Figure 61. The "Select report to run" dropdown now shows "Extra Students for Exams Office". The "Enter Exam Code:" field contains "5BUS1094-0906" and the "Enter Date of Exam:" field contains "2016-06-22". A "Search" button is present. Below the search area is a table with five rows of data. The table has columns: ID_NUMBER, First_Name, Module_Code, Exam_Date, and Last_Name. The data is as follows:

ID_NUMBER	First_Name	Module_Code	Exam_Date	Last_Name
[REDACTED]	Rupa	5BUS1094-0906	2016-06-22	Patel
[REDACTED]	Vishal	5BUS1094-0906	2016-06-22	Dinescumar
[REDACTED]	Z	5BUS1094-0906	2016-06-22	Chen-Jiajie
[REDACTED]	Elliott	5BUS1094-0906	2016-06-22	Cable
[REDACTED]	Haotian	5BUS1094-0906	2016-06-22	Wu

At the bottom of the table is an "Export" button.

Figure 62 Extra Students for Exams Office Report

ExamsOfficeReports

Exams Office Fingerprint Recognition System - Welcome!

Select report to run	<input type="button" value="Absentee Students for Marker"/>																		
Enter Exam Code:	<input type="text" value="5BUS1094-0906"/>																		
Enter Date of Exam:	<input type="text" value="2016-06-22"/> Format: YYYY-MM-DD																		
<input type="button" value="Search"/>																			
<table border="1"> <thead> <tr> <th>Candidate_Number</th> <th>Module_Code</th> <th>Exam_Date</th> </tr> </thead> <tbody> <tr> <td>[REDACTED]</td> <td>5BUS1094-0906</td> <td>2016-06-22</td> </tr> </tbody> </table>		Candidate_Number	Module_Code	Exam_Date	[REDACTED]	5BUS1094-0906	2016-06-22												
Candidate_Number	Module_Code	Exam_Date																	
[REDACTED]	5BUS1094-0906	2016-06-22																	
[REDACTED]	5BUS1094-0906	2016-06-22																	
[REDACTED]	5BUS1094-0906	2016-06-22																	
[REDACTED]	5BUS1094-0906	2016-06-22																	
[REDACTED]	5BUS1094-0906	2016-06-22																	
<input type="button" value="Export"/>																			

Figure 63 Absentee Students for Marker Report

ExamsOfficeReports

Exams Office Fingerprint Recognition System - Welcome!

Select report to run	<input type="button" value="Extra Students for Marker"/>																		
Enter Exam Code:	<input type="text" value="5BUS1094-0906"/>																		
Enter Date of Exam:	<input type="text" value="2016-06-22"/> Format: YYYY-MM-DD																		
<input type="button" value="Search"/>																			
<table border="1"> <thead> <tr> <th>Candidate_Number</th> <th>Module_Code</th> <th>Exam_Date</th> </tr> </thead> <tbody> <tr> <td>[REDACTED]</td> <td>5BUS1094-0906</td> <td>2016-06-22</td> </tr> </tbody> </table>		Candidate_Number	Module_Code	Exam_Date	[REDACTED]	5BUS1094-0906	2016-06-22												
Candidate_Number	Module_Code	Exam_Date																	
[REDACTED]	5BUS1094-0906	2016-06-22																	
[REDACTED]	5BUS1094-0906	2016-06-22																	
[REDACTED]	5BUS1094-0906	2016-06-22																	
[REDACTED]	5BUS1094-0906	2016-06-22																	
[REDACTED]	5BUS1094-0906	2016-06-22																	
<input type="button" value="Export"/>																			

Figure 64 Extra Students for Marker Report

The system also allows for the search results to be exported to an excel file as required (Figure 65).

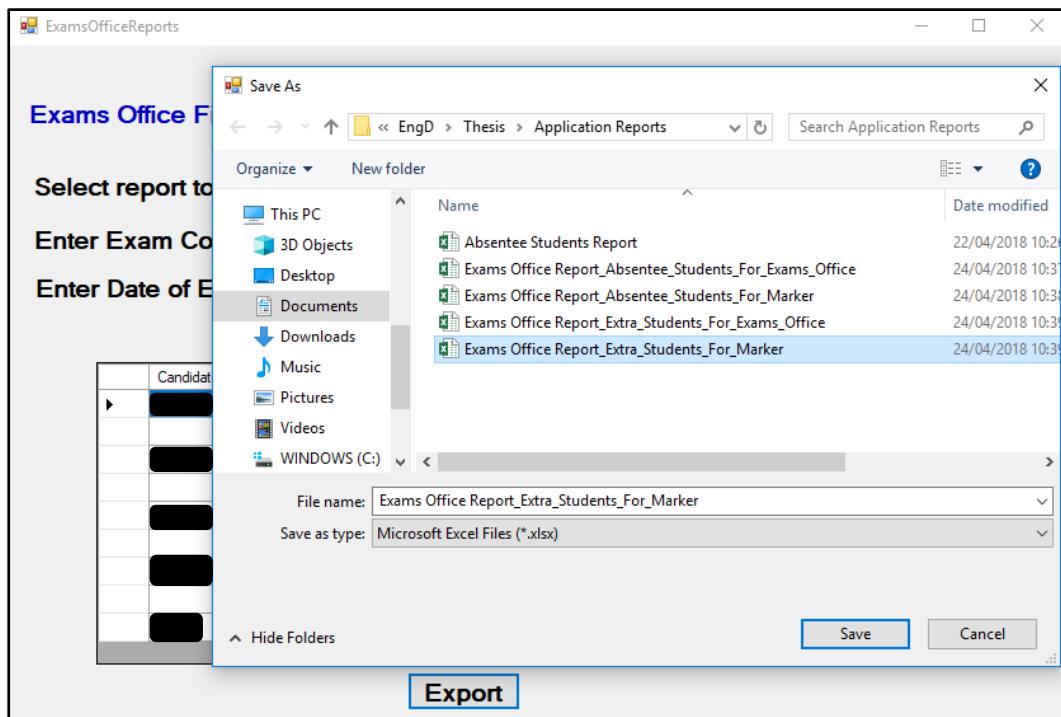


Figure 65 Export Exams Office Reports

- ExamInvigilator.cs (Figure 66) – Connects to the ‘IDCardSystem’ database and loads the Fingerprint Application when invigilator clicks on the ‘Sign in For Exam’ button.

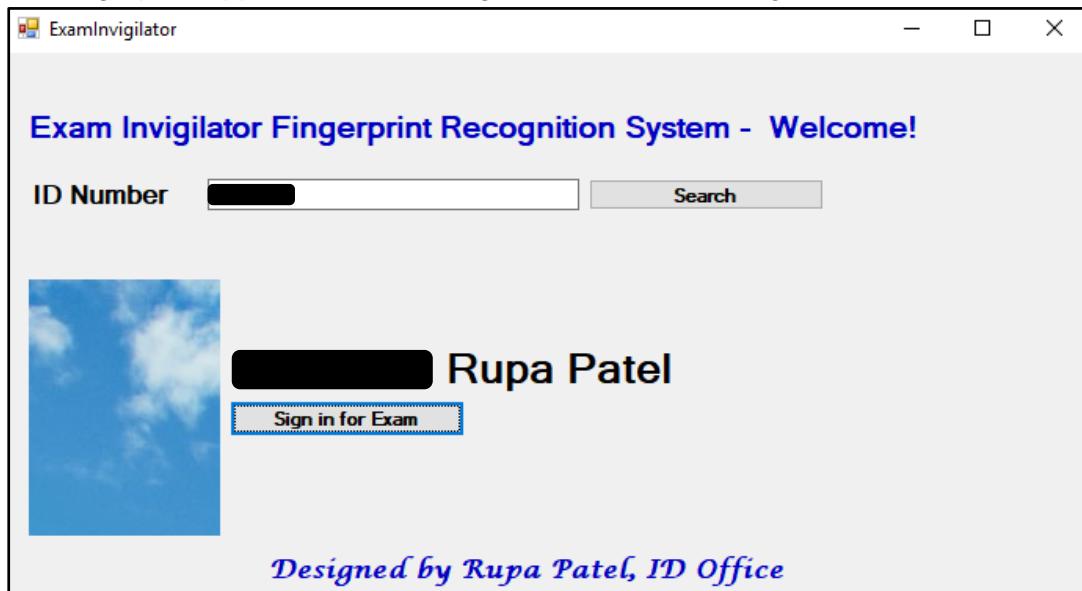


Figure 66 Form for Exam Invigilator

4.5.2 Programming

The database queries mentioned in Section 4.3.4 have been amended to allow system users to run customised searches and reports. The system links to the fingerprint application and all 3 databases using SQL and C#. This section covers the programming element of the Login Form reporting system.

- LoginForm.cs – Following input of username and password, when the ‘Login’ button is pressed, or the user hits the ‘Enter’ key on their keyboard, the username is

checked against ‘Role’ field in the LoginAccounts table within the FingerprintApplication database. If the role is ‘Admin’, the ‘SearchForm.cs’ form loads, if it’s ‘Exams’, the ‘ExamsOfficeReports.cs’ form opens and if it’s ‘Invigilator’, the ‘ExamInvigilator.cs’ form loads and if user is not found, a message box is displayed as shown in the code below. Full code for this form is shown in Appendix IX.

```

private void Login_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection();
    con.ConnectionString = "Data Source=RUPA-TOSH\SQLEXPRESS;Initial Catalog=FingerprintApplication;Integrated Security=True";
    con.Open();
    string Username = UsernameTextBox.Text;
    string Password = PasswordTextBox.Text;
    SqlCommand cmd = new SqlCommand("select Username,Password,Role from LoginAccounts where Username='" + UsernameTextBox.Text + "'and Password='" + PasswordTextBox.Text + "'", con);
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    DataTable dt = new DataTable();
    da.Fill(dt);
    if (dt.Rows.Count == 1)
    {
        switch (dt.Rows[0]["Role"] as string)
        {
            case "Admin":
            {
                this.Hide();
                SearchForm ss = new SearchForm();
                ss.Show();
                break;
            }

            case "Exams":
            {
                this.Hide();
                ExamsOfficeReports mf = new ExamsOfficeReports();
                mf.Show();
                break;
            }

            case "Invigilator":
            {
                this.Hide();
                ExamInvigilator mf = new ExamInvigilator();
                mf.Show();
                break;
            }
        }
    }
    else
    {
        MessageBox.Show("Invalid Login! Please check Username and Password");
    }
    con.Close();
}

```

- SearchForm.cs – Once a user enters the student number and presses the ‘Search’ button or presses the ‘Enter’ button on the keyboard, the student number is checked

in the IPSUsers table within the IDCardSystem database. If the student number is found, the student number, first name and last name is retrieved from the database. The student photograph is linked from the link specified in the ‘PhotoUNC’ field in the table and the ‘Enroll Fingerprint’ button is displayed. If no record is found, a message box is displayed indicating the record searched is not found as shown in code below. Full code for this form can be seen in Appendix X.

```

private void SearchButton_Click(object sender, EventArgs e)
{
    SqlConnection connection = new SqlConnection();
    connection.ConnectionString = "Data Source=RUPA-TOSH\\SQLEXPRESS;Initial Catalog=IDCardSystem;Integrated Security=True";
    String sql = "select * from IPSUsers where ADLogonName=' " + SearchBox.Text
+ " '";
    SqlDataAdapter dataAdapter = new SqlDataAdapter(sql, connection); //c.con
is the connection string
    using (SqlCommand cmd = new SqlCommand(sql, connection))

    {
        connection.Open();
        using (SqlDataReader reader = cmd.ExecuteReader())

        {
            if (reader.HasRows)
            {
                while (reader.Read())
                {
                    listBox1.Items.Add(reader["ADLogonName"].ToString() + " "
+ reader["FirstName"].ToString() + " " + reader["LastName"].ToString());
                    Image image =
Image.FromFile(@"C:\Users\Rupa\Documents\Pictures\Rupa\" + SearchBox.Text + ".jpg");
                    this.MemberPhoto.Image = image;
                    MemberPhoto.Visible = true;
                    listBox1.Visible = true;
                    EnrollFingerprint.Visible = true;

                }
                reader.Close();
            }
            else
            {
                MessageBox.Show("No record is found with this number: " + " " +
SearchBox.Text.ToString() + " ");
            }
        }
        connection.Close();
    }
}

```

- ExamsOfficeReports.cs – Once the required report is selected with Exam Code and Date of Exam values populated, the Exam Code and Date of Exam values are compared with fields in the exams database ‘esdev’. This is then compared with student records within the FingerprintApplication and IDCardSystem databases depending on the report required as shown in the code below. Full code is shown in Appendix XI.

```

private void button1_Click(object sender, EventArgs e)
{

```

```

        if (comboBox1.SelectedItem.ToString() == "Absentee Students for Exams
Office")
    {
        var select = "SELECT HostKey AS ID_NUMBER, Description AS FULL_NAME,'" +
+ textBox1.Text + "' AS Module_Code," + "'" + textBox2.Text + "' AS Exam_Date INTO
#TempTableList FROM esdev.rdreader.V_Student WHERE Id IN(SELECT StudentId FROM
esdev.rdreader.V_ExamAllocation E WHERE EXISTS (SELECT F.ExamRequirementId FROM
esdev.rdreader.V_Examination F WHERE E.ExamRequirementId=F.ExamRequirementId AND
F.NAME LIKE '" + textBox1.Text + "%' AND StartDateTime >= '" + textBox2.Text + "'));
SELECT ADLogonName AS ID_Number, FirstName AS First_Name, LastName AS Last_Name INTO
#TempTable FROM IDCardSystem.dbo.IPSUsers WHERE ADLogonName IN(SELECT DISTINCT
MemberNo FROM FingerprintApplication.dbo.FingerprintIdentification WHERE DeviceId
IN(SELECT DeviceID FROM FingerprintApplication.dbo.DeviceList d INNER JOIN
esdev.rdreader.V_Location l ON d.DeviceLocation=l.Description WHERE EXISTS(SELECT
StudentId, ExamRoomLocationId FROM esdev.rdreader.V_ExamAllocation E WHERE EXISTS
(SELECT F.ExamRequirementId FROM esdev.rdreader.V_Examination F WHERE
E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '" + textBox1.Text + "%' AND
StartDateTime >= '" + textBox2.Text + "')));SELECT * FROM #TempTableList WHERE
ID_Number NOT IN(SELECT ID_NUMBER FROM #TempTable);";
        var c = new SqlConnection("Data Source = RUPA-TOSH\SQLEXPRESS;
Initial Catalog = esdev; Integrated Security = True");
        var dataAdapter = new SqlDataAdapter(select, c);
        var commandBuilder = new SqlCommandBuilder(dataAdapter);
        var ds = new DataSet();
        dataAdapter.Fill(ds);
        dataGridView1.ReadOnly = true;
        dataGridView1.DataSource = ds.Tables[0];
        dataGridView1.Visible = true;
        ExportResults.Visible = true;
    }
    if (comboBox1.SelectedItem.ToString() == "Extra Students for Exams
Office")
    {
        var select = "SELECT HostKey AS ID_NUMBER, Description AS FULL_NAME,'" +
+ textBox1.Text + "' AS Module_Code," + "'" + textBox2.Text + "' AS Exam_Date INTO
#TempTableList FROM esdev.rdreader.V_Student WHERE Id IN(SELECT StudentId FROM
esdev.rdreader.V_ExamAllocation E WHERE EXISTS (SELECT F.ExamRequirementId FROM
esdev.rdreader.V_Examination F WHERE E.ExamRequirementId=F.ExamRequirementId AND
F.NAME LIKE '" + textBox1.Text + "%' AND StartDateTime >= '" + textBox2.Text + "'));
SELECT ADLogonName AS ID_Number, FirstName AS First_Name, LastName AS Last_Name,'" +
+ textBox1.Text + "' AS Module_Code," + "'" + textBox2.Text + "' AS Exam_Date INTO
#TempTable FROM IDCardSystem.dbo.IPSUsers WHERE ADLogonName IN(SELECT DISTINCT
MemberNo FROM FingerprintApplication.dbo.FingerprintIdentification WHERE DeviceId
IN(SELECT DeviceID FROM FingerprintApplication.dbo.DeviceList d INNER JOIN
esdev.rdreader.V_Location l ON d.DeviceLocation=l.Description WHERE EXISTS(SELECT
StudentId, ExamRoomLocationId FROM esdev.rdreader.V_ExamAllocation E WHERE EXISTS
(SELECT F.ExamRequirementId FROM esdev.rdreader.V_Examination F WHERE
E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '" + textBox1.Text + "%' AND
StartDateTime >= '" + textBox2.Text + "')));SELECT * FROM #TempTable WHERE ID_Number
NOT IN(SELECT ID_NUMBER FROM #TempTableList);";
        var c = new SqlConnection("Data Source = RUPA-TOSH\SQLEXPRESS;
Initial Catalog = esdev; Integrated Security = True");
        var dataAdapter = new SqlDataAdapter(select, c);
        var commandBuilder = new SqlCommandBuilder(dataAdapter);
        var ds = new DataSet();
        dataAdapter.Fill(ds);
        dataGridView1.ReadOnly = true;
        dataGridView1.DataSource = ds.Tables[0];
        dataGridView1.Visible = true;
        ExportResults.Visible = true;
    }
    if (comboBox1.SelectedItem.ToString() == "Absentee Students for Marker")
    {

```

```

        var select = "SELECT HostKey AS ID_NUMBER, Numeric1 AS
Candidate_Number,'" + textBox1.Text + "' AS Module_Code," + "" + textBox2.Text + "' AS Exam_Date INTO #TempTableList FROM esdev.rdreader.V_Student WHERE Id IN(SELECT
StudentId FROM esdev.rdreader.V_ExamAllocation E WHERE EXISTS (SELECT
F.ExamRequirementId FROM esdev.rdreader.V_Examination F WHERE
E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '" + textBox1.Text + "%' AND
StartTime >= '" + textBox2.Text + "')); SELECT ADLogonName AS ID_Number, FirstName
AS First_Name, LastName AS Last_Name INTO #TempTable FROM IDCardSystem.dbo.IPSUsers
WHERE ADLogonName IN(SELECT DISTINCT MemberNo FROM
FingerprintApplication.dbo.FingerprintIdentification WHERE DeviceId IN(SELECT DeviceID
FROM FingerprintApplication.dbo.DeviceList d INNER JOIN esdev.rdreader.V_Location l ON
d.DeviceLocation=l.Description WHERE EXISTS(SELECT StudentId, ExamRoomLocationId FROM
esdev.rdreader.V_ExamAllocation E WHERE EXISTS (SELECT F.ExamRequirementId FROM
esdev.rdreader.V_Examination F WHERE E.ExamRequirementId=F.ExamRequirementId AND
F.NAME LIKE '" + textBox1.Text + "%' AND StartTime >= '" + textBox2.Text +
"')));SELECT Candidate_Number, Module_Code, Exam_Date FROM #TempTableList WHERE
ID_Number NOT IN(SELECT ID_NUMBER FROM #TempTable);";
        var c = new SqlConnection("Data Source = RUPA-TOSH\SQLEXPRESS;
Initial Catalog = esdev; Integrated Security = True");
        var dataAdapter = new SqlDataAdapter(select, c);
        var commandBuilder = new SqlCommandBuilder(dataAdapter);
        var ds = new DataSet();
        dataAdapter.Fill(ds);
        dataGridView1.ReadOnly = true;
        dataGridView1.DataSource = ds.Tables[0];
        dataGridView1.Visible = true;
        ExportResults.Visible = true;
    }
    if (comboBox1.SelectedItem.ToString() == "Extra Students for Marker")
    {
        var select = "SELECT HostKey AS ID_NUMBER, Description AS FULL_NAME,'" +
+ textBox1.Text + "' AS Module_Code," + "" + textBox2.Text + "' AS Exam_Date INTO
#TempTableList FROM esdev.rdreader.V_Student WHERE Id IN(SELECT StudentId FROM
esdev.rdreader.V_ExamAllocation E WHERE EXISTS (SELECT F.ExamRequirementId FROM
esdev.rdreader.V_Examination F WHERE E.ExamRequirementId=F.ExamRequirementId AND
F.NAME LIKE '" + textBox1.Text + "%' AND StartTime >= '" + textBox2.Text + "'));
SELECT ADLogonName AS ID_Number, FirstName AS First_Name, LastName AS Last_Name,'" +
+ textBox1.Text + "' AS Module_Code," + "" + textBox2.Text + "' AS Exam_Date INTO
#TempTable FROM IDCardSystem.dbo.IPSUsers WHERE ADLogonName IN(SELECT DISTINCT
MemberNo FROM FingerprintApplication.dbo.FingerprintIdentification WHERE DeviceId
IN(SELECT DeviceID FROM FingerprintApplication.dbo.DeviceList d INNER JOIN
esdev.rdreader.V_Location l ON d.DeviceLocation=l.Description WHERE EXISTS(SELECT
StudentId, ExamRoomLocationId FROM esdev.rdreader.V_ExamAllocation E WHERE EXISTS
(SELECT F.ExamRequirementId FROM esdev.rdreader.V_Examination F WHERE
E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '" + textBox1.Text + "%' AND
StartTime >= '" + textBox2.Text + "')));SELECT DISTINCT Numeric1 AS
CANDIDATE_NUMBER,'" + textBox1.Text + "' AS Module_Code," + "" + textBox2.Text + "' AS Exam_Date INTO #TempTableC FROM esdev.dbo.V_Student WHERE HostKey IN(SELECT
ID_Number FROM #TempTable WHERE EXISTS(SELECT DISTINCT Numeric1, HostKey FROM
esdev.dbo.V_Student WHERE HostKey=ID_Number));SELECT * From #TempTableC;";
        var c = new SqlConnection("Data Source = RUPA-TOSH\SQLEXPRESS;
Initial Catalog = esdev; Integrated Security = True");
        var dataAdapter = new SqlDataAdapter(select, c);
        var commandBuilder = new SqlCommandBuilder(dataAdapter);
        var ds = new DataSet();
        dataAdapter.Fill(ds);
        dataGridView1.ReadOnly = true;
        dataGridView1.DataSource = ds.Tables[0];
        dataGridView1.Visible = true;
        ExportResults.Visible = true;
    }
}

```

- ExamInvigilator.cs – Once the invigilator enters the student number and clicks on the ‘Search’ button or presses the ‘Enter’ button the keyboard, the ID number is compared with the ADLogonName in the IDCardSystem database. If a match is found, the student’s ID number, first name and last name is retrieved from the database and the matching photograph is linked with the relevant path. If no record is matched, a message box is displayed as shown in the code below. Full code can be seen in Appendix XII.

```

private void SearchButton_Click(object sender, EventArgs e)
{
    SqlConnection connection = new SqlConnection();
    connection.ConnectionString = "Data Source=RUPA-TOSH\SQLEXPRESS;Initial Catalog=IDCardSystem;Integrated Security=True";
    String sql = "select * from IPSUsers where ADLogonName=' " + SearchBox.Text
+ " '";
    SqlDataAdapter dataAdapter = new SqlDataAdapter(sql, connection); //c.con
is the connection string
    using (SqlCommand cmd = new SqlCommand(sql, connection))

    {
        connection.Open();
        using (SqlDataReader reader = cmd.ExecuteReader())

        {
            if (reader.HasRows)
            {
                while (reader.Read())
                {
                    listBox1.Items.Add(reader["ADLogonName"].ToString() + " "
+ reader["FirstName"].ToString() + " " + reader["LastName"].ToString());
                    Image image =
Image.FromFile(@"C:\Users\Rupa\Documents\Pictures\Rupa\" + SearchBox.Text + ".jpg");
                    this.MemberPhoto.Image = image;
                    MemberPhoto.Visible = true;
                    listBox1.Visible = true;
                    SignIn.Visible = true;
                }
                reader.Close();
            }
            else
            {
                MessageBox.Show("No record is found with this number: " +
+ SearchBox.Text.ToString() + "'");
            }
        }

        connection.Close();
    }
}

```

CHAPTER 5: SYSTEM TESTING

The fingerprint application was first tested without any enhancements carried out to the classes defined by the libraries of the development kit and then tested again once enhanced. A commercial system has been purchased to test fingerprint recognition technology in a real exam situation as part of a Proof of Concept (PoC) pilot project carried out. This chapter highlights the outcomes from the fingerprint enrolment and authentication testing conducted with students and presents full test results obtained from both the developed system and the commercial system.

The developed system refers to the fingerprint application enhanced using the Software Development Kit. The commercial system refers to an out of the box system which was bought to test and compare with the developed system.

5.1 System Deployment via Proof of Concept

Following the receipt of a funding of £6000 from the UH Proof of Concept funding scheme on 20th January 2017, a commercial system along with two fingerprint wall readers and two biometric tablets were purchased from a supplier called Morpho. The funding also covered the costs for recruiting and training additional staff.

The purpose of this phase of the project was to test a commercial fingerprint recognition system in a real University exam setting and share outcomes with the UH teams to then decide whether such a fingerprint-based process is viable at the University of Hertfordshire.

The project commenced on 20th March 2017 with an end date of 30th June 2018. A project plan can be seen in Appendix XIII. As the project start date was after the University's key registration period, the project was split into two phases: Phase One to capture fingerprints and pilot the process during the Semester B exams in May/June 2017 and Phase Two to allow the inclusion of a full academic year for carrying out further tests.

Before carrying out tests in phase one, the following crucial tasks were completed:

- Stakeholders' consultation – Key stakeholders such as suppliers, IT team, Estates team, Senior managers of UH and Deans of Schools were consulted to share project plan and gather feedback for implementation. A focus group was then formed which included some of these stakeholders. Three suppliers were invited to UH to provide a demo of their hardware and software to the focus group. Out of the three, Morpho was selected based on these factors: low costs, multiple biometric modalities in one device making it future proof and the availability of a mobile fingerprint device called Morpho Tablet.
- Consent form – A consent form was created with inputs from the University's Data Protection Officer. Any student taking part in this proof of concept project was asked to sign a consent form. A copy of the consent form has been included in Appendix XIV.
- Server – A password-protected, secure and dedicated server was identified and made available for this project by our IT department. Once this was setup, an SQL database to store fingerprint templates and hold related information such as, ID number, name, finger ID was attached. The commercial system called Morpho Manager was also installed on the server along with the Morpho Manager service which is required to communicate with the wall reader and Morpho Manager application.

- Room survey – All 65 rooms across the University that have held an exam and already have an ID card reader for attendance monitoring, were physically visited to identify how many attendance readers were installed in that room and whether it has a network port and the right power supply required to install a fingerprint wall reader during exams. The plan was to select a room that has two attendance readers, so when one is replaced with a fingerprint wall reader, both the attendance monitoring system and fingerprint system can run in parallel. The former would allow students to carry on swiping their ID card to register their attendance and the latter would allow students to register for their exam. This allows an exam room to still be used for teaching which would require students to register their attendance.
- Staff recruitment and training – Four additional staff were recruited and three more were appointed from within student administration office to assist with fingerprint enrolment sessions. All staff were provided with full training.
- Biometric tablets – The Morpho tablets arrived without any applications installed. These were configured before use.

The PoC project has enabled me to gain valuable experience with fingerprint systems, enhance understanding of the fingerprint technologies and obtain useful insights of the relevant University systems and processes. More significantly, it has enabled a performance comparison between the fingerprint system developed in this research study and a commercial system. The remaining of this chapter details the tests conducted with the developed system and presents the results in comparison to those of the commercial system.

5.2 Enrolment

This subsection details key statistics gathered from enrolment sessions on both the developed system and the commercial system.

5.2.1 Enrolment with the developed system

The first test of the developed fingerprint application (before enhancements) was carried out on 8th March 2017 by approaching students on ‘Data Security and Biometrics’ module within the School of Engineering.

Before asking them to take part in this project, a presentation was given at each session to disseminate information about the project.

Out of the 23 students that attended the class, 13 volunteered to take part in the project. The process involved:

- Reading the participant information sheet as shown in Appendix XV.
- Sign consent form as per Appendix XVI.
- Have one fingerprint per student enrolled.
- Complete feedback form as per Appendix XVII.

Statistics from this session:

- Total time taken to create records, enter personal details (including Student ID number and FingerprintID as defined in section 4.4) and enrol fingerprints of 13 students: 1 minute and 13 seconds per student.
- Average time to enrol fingerprint per student: 3 seconds.

- Average time to identify a user once enrolled: 2 seconds.

Fingerprints were captured using Fingerprint scanner and dongle shown in Figure 67.

USB Fingervein Reader:



License Dongle:



Figure 67 Fingerprint Scanner and License Dongle

The constraint for this session was that the fingerprint enrolment was carried out during the ongoing class. Students were more willing to take part once they saw the format the fingerprints are stored in. Sample fingerprint template:

```
ABAIAMgDAAACEAgABAAAAAAHAAIIeAgABAAAAAgADAULEAgABAAAAAgADAUJEAgA
BAAAAAAAsBAQNEAgABAAAAAIlAQAOEAgABAAAAAIlAQAPEAgABAAAAAAAAAFEEAg
ABAAA AJABAAGEAgABAAA JABAHEAgABAAAAAAA+kMMEAgABAAAAAEAAA wE
AgABAAA AldE1+UKEAgABAAAAAAAEEQgAJAMAAAERCAA EAAAAAQAAAIRCAA
EAAAAAAAAMRCAAEAAAQAQIAAQRCAAEAAAABAAA BARCAA EAAAQAAC
RCAA EAAAQAQAAABMRCAA EAAA kAEAABQRCAA EAAA kAEAABERCAA EAAA
yAAA ABIRCAA EAAA yAAAABURCAA EAAAAAAA BYRCAA EAAAAAAA YRCAA EAAA
wAAA ACMRCAA EAAA kAEA ACQRCAA EAAA kAEA ACERCAA EAAA yAAA ACIRCAA EAAA
AyAAA AASCABQAgAAEBIIAQAAAFAAAAERIIAQAAA QAAA JhIIAQAAA QAAA
IBIIAQAAAAnAAAIRIIANQBAACCACQAIJ5rQA4VAAcFACwAGMVYQA4UAACdAD8AF
VZIQA4YAADaAFEApX YHQA4QAACeAFYAh4SFQA4pAADmAFkAAe+kQA4gAABNAGkAY
xSdPwMgAACCAGsAxDmVQA4QAABoAGwABs2LQA4QAACaAHIAg6aeQAsUAACmAHw
AwsqhQAOQAAChAIcAVXroPw cRAADXAIkAKM4AQA4hAABKAI0ABs2LQAcQAABfAI8Ax
mVQA4RAACjAJEAAe+kQAO RAAC4AKMAUgv1Pw4IAACuAKYAKM4AQA4hAAD/ALEAI/A
ZQAO RAABN ALIAV+nbPwUgAADkAlcAIYEmQA4hAADXAMQAO MksQA4gAABwAMYAp
YHQAO QAADBANIAoMksQA4QAADzANIAHKM/QAO hAAB3ANQAOjggQA4RAACcANQAIY
EmQA4RAABjANUAIYEmQA4RAADXANUAm+tFQA4QAAD1AOQAm+tFQAKgAABkAOYA
nlo5QAgRAACcAOYAnlo5QA4RAACAAOcAHxIzQA4QAAB5APUAHKM/QA4hAAB/AAwBm
XxSQAw hAACB ABgBm+tFQAGhAACpAC0B5MsWPgUhAACoADwBfFnEQA4gAACtAE4B
m+tFQA4hAAA EwgAPAAA ADETCAA EAAA APgAAADM TCAA EAAA yAAA ADITCAA EAA
AAPgAAADUTCAA EAAA AnwICQUATCAA EAAA wAAA ==
```

Over the 13 students who took part, on average it took 1 attempt per student to capture a fingerprint image with high quality. However, it took 4 attempts before one student's fingerprint could be captured with high image quality. This was due to dry fingers. Having learnt from this, moisturising cream was made available for further test sessions.

Enhancements to the project system were implemented after the first round of test. These include:

- Storing fingerprint templates in a database instead of XML file and record ID number against it.
- Retrieve ID number of the student identified by the fingerprint identification process and obtain corresponding first name and last name from the ID card database.
- Insert details of each identified students along with date and time of identification into FingerprintApplication database to allow generation of reports using the exams reporting system.

Once the application was modified, the system was tested on 23rd March 2018, by students registered on the 'Data Security and Biometrics' module. Out of around 20 students who attended the class, 8 volunteered to take part in the project. Results from this second test with the developed system are presented later in comparison to those of the commercial system.

5.2.2 Enrolment with the commercial system

As mentioned, the Morpho Manager commercial system was also tested for enrolment as part of the PoC project. The commercial system captures a fingerprint four times and requires these from at least two fingers as shown in Figure 68. This adds significant amount of time to the enrolment process. Four prints per finger were captured on the commercial system but only one template per finger was generated based on the multiple prints. Matching score was calculated based on comparison against one template.



Figure 68 Enrolment using Commercial System

An optional Duress Finger can also be registered but this was not implemented for the pilot with the project system. To maintain consistency, it was decided that all students would be asked to register their left and right index fingers during all the tests with Morpho Manager. By registering two fingers, the system provides a backup mechanism in case one of the enrolled fingers cannot be used e.g. due to cuts or other unforeseen issues during authentication. If one enrolled finger is rejected but the other enrolled finger is accepted, the commercial system would still accept the user. It does not require both fingers to be matched.

Five fingerprint enrolment sessions were arranged with students from three different Schools: School of Computer Science, School of Engineering and Technology, and School of Law. Two laptops were setup to capture fingerprints of students. Table 7 shows some key information of these sessions.

Table 7 Key Information from Tests using Commercial System

Date	Module	Number of Students Enrolled on System	Constraints/Notes
20/04/2017	Engineering Statistics Students – School of Engineering	9	<ul style="list-style-type: none"> • There was a scheduled test on the day • Split into 2 locations • Network ports weren't available so had to use Wi-Fi
25/04/2017	All final year students – School of Computer Science	39 out of 154	<ul style="list-style-type: none"> • There was an NSS Survey being carried out at the same time • A Careers & Employment stand in the same room which diverted attention of students • Network ports weren't available so had to use Wi-Fi
14/03/2018	LLB Students – School of Law	13 out of 20	<ul style="list-style-type: none"> • A fingerprint capture session was timetabled and lecture theatre booked so students could listen to presentation first and then volunteer to take part
16/03/2018	Programming Students – School of Computer Science	8 out of 12	<ul style="list-style-type: none"> • Students had another lecture to attend at the booked time • The follow-up lecture took place on a Friday afternoon so not many of them turned up for lecture hence we couldn't capture many
26/03/2018	Project Management & Product Development Students – School of Engineering and Technology	18 out of 50+	<ul style="list-style-type: none"> • Poor Wi-Fi connection

Main challenges faced were:

- Short period to run Phase 1 of Proof of Concept project as exams were due to take place within 2 months of funding approval notification received.
- Only had one week window to install readers during which classrooms were available to carry out installation.
- As the tests were being carried out in actual exam environment, the biometric reader had to operate in parallel with the existing attendance monitoring system so students can carry on using existing system to register attendance. For this reason, rooms with two readers had to be identified where one could be swapped using the existing power and network port and the other could remain. There were only 13 rooms with more than one attendance readers installed.
- One exam could take place in 4 different rooms so some students whose fingerprints were collected, didn't have their exam taking place in the rooms where the readers were installed which means tests could not be carried out for these students.
- The Engineering Statistics test (after which fingerprints were to be captured) was split into two rooms which were far from each other. This meant that the probability had already been reduced by 50% as there was only one fingerprint USB scanner to capture fingerprints. Students also had to be elsewhere after their test. Another USB scanner was purchased at a later date for Phase Two PoC testing.
- Due to the temporary setup for fingerprint enrollment, a laptop with Wi-Fi connection was used and there were connection issues which put some students off and those waiting to take part had to leave as they had a lecture to attend. The process of capturing fingerprints was extremely slow as the system failed to connect to the server database when network connection was lost. In addition to Wi-Fi connection, as the database for the commercial system was located on a UH server, the laptops used also needed to be connected to the University's Virtual Private Network (VPN) to gain access to the server.
- The specification of one laptop was also lower than the other making the enrolment process slower on the low specification laptop.
- The final year Computer Science students were asked to complete NSS Survey and there was also a Careers and Employment stand in the Computer Science lab where projects were being handed in and fingerprints were being captured. This also reduced the probability as students couldn't allocate time to everything.

Despite of these issues, students who did not have time constraints were happy to wait as they recognise the benefits of the use of fingerprint technology in the University's exam process.

The average time to create a record, record the required information, student reading and signing a consent form and enrolling two fingerprints was 1 minute and 40 seconds per student. With system issues, mainly losing Wi-Fi connection, the total time rose to just over 3 minutes.

There weren't any students whose fingerprints could not be captured.

5.2.3 Collectability and universality of fingerprints of University student population

This subsection provides statistics from both systems and analyse the collectability and universality of fingerprints with our University student population.

One enhancement to the developed system was to enable the recording of the live quality scores of each of the captured fingerprint image. The scores recorded in the second test with the developed system are as shown in Table 8. All students' fingerprints were captured on the first attempt and all produced good quality scores with an average of 95.

Table 8 Live Quality Scores using Developed System

No.	Live Quality Score
1.	122
2.	106
3.	92
4.	91
5.	132
6.	106
7.	106
8.	98

The live quality image scores recorded using the commercial system are presented in Table 9, with challenging fingers highlighted in bold.

The commercial system requires an image quality score above 49 to be considered as a valid print and activate the subsequent enrolment process. If a score is below 49, the system would prompt the user to either restart the fingerprint capture process or accept the generated template with low quality prints.

Summary of findings from the commercial system tests:

- Proves the theory that burnt fingers can cause issues. The scores recorded in row 6 belong to students whose fingers were burnt and the substantially low quality scores reflect this. This was noticed during enrolment.
- Dry fingers also affect the enrolment and authentication processes as mentioned in the subject review and section 5.2.1. The scores recorded in row 25 confirm this. This particular student's hands were dry hence generating a low average quality score of 24 for the right index finger. The student was advised to apply hand cream on the left hand. Once this was applied, the average score improved by 63% as seen from scores recorded of his/her left index finger.

With general public, the universality and collectability rates of fingerprints are both medium. In contrast, as showed in Table 8 and Table 9, the University student population tend to have fingerprints that can produce print images with high quality and hence high universality. The fingerprint quality scores in Table 8 and Table 9 averaged to 95 and 75, respectively. In addition, with some easy to implement measures i.e. use of hand cream, students with burnt fingers were able to produce good prints, indicating an improved collectability.

Table 9 Live Quality Scores using Commercial System

No.	Right Index Finger					Left Index Finger				
	Score 1	Score 2	Score 3	Score 4	Average Quality	Score 1	Score 2	Score 3	Score 4	Average Quality
1.	85	85	85	85	85	51	64	66	64	64
2.	76	73	74	66	74	52	75	68	38	64
3.	87	88	86	87	80	87	87	41	72	72
4.	87	87	86	87	87	87	88	87	87	87
5.	77	72	66	71	72	72	74	74	75	73
6.	76	73	78	80	75	6	27	33	27	21
7.	82	86	87	86	85	85	85	83	83	85
8.	78	75	71	77	75	74	74	79	78	75
9.	89	90	90	92	89	86	88	89	88	88
10.	69	75	75	76	73	67	74	75	74	73
11.	79	81	79	71	80	80	79	68	63	76
12.	82	85	85	86	85	81	85	80	81	82
13.	75	68	73	67	72	65	69	72	71	68
14.	85	81	85	85	83	69	63	58	60	63
15.	77	78	81	83	79	74	80	81	75	78
16.	76	89	90	90	86	82	91	92	91	89
17.	87	83	87	86	86	83	85	85	82	82
18.	58	78	81	62	73	82	79	76	80	79
19.	68	87	87	87	83	87	87	87	87	87
20.	56	51	58	56	55	61	63	56	66	60
21.	87	87	87	85	87	89	88	88	89	88
22.	73	85	87	86	83	85	86	87	87	86
23.	90	88	92	90	90	89	77	88	86	86
24.	71	74	79	78	75	71	68	74	74	71
25.	26	32	18	14	24	61	68	63	59	64
26.	74	83	86	85	81	85	72	88	66	83
27.	87	83	89	88	87	89	88	87	88	88
28.	69	71	73	67	71	63	71	62	59	65
29.	79	81	83	85	71	86	82	75	79	81
30.	90	90	91	90	90	90	90	90	89	90
31.	55	73	67	68	64	74	77	68	76	74
32.	58	60	66	43	61	67	69	78	73	72
33.	86	77	74	82	79	85	77	71	86	77
34.	60	71	72	85	67	56	71	85	78	71
35.	87	90	90	87	89	88	88	90	90	89
36.	63	74	66	76	68	48	74	72	79	64
37.	57	52	52	47	53	58	51	51	49	53
38.	71	85	85	86	80	78	79	66	85	75

5.3 Authentication

With both systems, fingerprint identification was performed to recognise the identity of an exam candidate in the place of an ID card.

As mentioned in the Chapter 3, although verification would allow a faster authentication process, it can pose security risk. To facilitate verification, the fingerprint template of a particular individual had to be distinguished from templates of the other individuals based on some personal information rather than fingerprint. This can be through manually entering a user ID or retrieving the user ID from an ID card. The time required by the manual approach would be longer than the 2 seconds required by the fingerprint identification process of the developed system. It would require additional actions involving the students and is prone to human errors such as a wrong ID being cited by the student or entered by the invigilator.

User ID can be retrieved from an ID card. However, this would require an additional ID card reader and the issue with a lost/forgotten ID card would occur.

Another approach to facilitate verification is to have the fingerprint template(s) stored on one's ID card so a one-to-one match could be carried out. However, if the template is stored on an ID card and the card is lost or stolen, the security of the data stored could be compromised. Following conference calls with our ID card manufacturers, they also advised against storing the template on ID cards. One of the other issue with this would be space. It can take up space on the ID chip so the University would have to invest in ID cards with chips that would have larger storage capacity to store fingerprint templates.

For identification, a threshold needs to be set. The threshold should be adjusted depending on the environment it is used in. For example, in a work environment where the purpose of using fingerprint technology is to record time a member of staff started work, a low threshold is acceptable so users can swipe in quickly. Where accuracy plays a vital role as opposed to quick access i.e. at airports, the threshold can be increased to ensure the system's false acceptance rate decreases.

The threshold should also be adjusted according to the number of users. For the system which has been developed for this project, the default threshold value was '3500'. This was reduced to '2000' for test purposes. The commercial system's default threshold value of '4000' was not changed during pilot tests.

A USB fingerprint/fingervein scanner was used to identify students when testing the developed system. A fingerprint wall reader which was purchased via Proof of Concept funding was used to carry out tests with the commercial system. The fingerprint wall reader as shown in Figure 69 was fitted for three weeks in a selected room where exam was due to take place. An attendance reader was taken off so existing power and network could be used for the fingerprint reader. The attendance system had to be configured to ensure the reader that was taken off temporarily wouldn't show errors.

Once the exams period ended, the fingerprint reader was taken off and replaced with the attendance reader. The attendance reader had to be activated on the attendance system and the fingerprint reader had to be deactivated on the commercial fingerprint system, Morpho Manager.

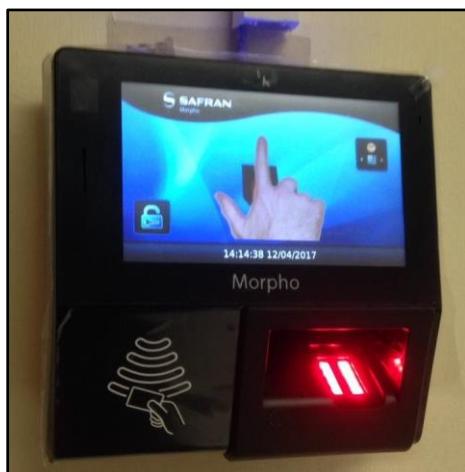


Figure 69 Fingerprint Wall Reader

It took 4 seconds to identify a fingerprint against a registered population of 48 with the commercial system using a fingerprint wall reader during a real exam. It took 2 seconds for the developed system against a registered population of 13.

The commercial system generates reports as seen in Figure 70.

The screenshot displays two main sections of a biometric system's user interface. The top section is a table titled 'Access Logs' with columns: Access Time, Access, User Name, Biometric Device, Key, and Presentation Type. The table lists several access entries for a user named 'ALEX JEFFRIES'. The bottom section shows a detailed view for 'ALEX JEFFRIES' with fields for Date of Birth (01/01/0001), Selected Authentication (Biometric (1:Many)), User Policy (All Access), and Disabled status (No). It also includes a photo placeholder, two sample fingerprints, and buttons for 'Export Photo' and 'Add Photo'.

Access Time	Access	User Name	Biometric Device	Key	Presentation Type
12/04/2017 15:45:53	Granted	ALEX JEFFRIES	F325	Exam IN	Biometric Identification
12/04/2017 15:09:06	Granted	ALEX JEFFRIES	F325	IN	Biometric Identification
12/04/2017 14:17:37	Granted	ALEX JEFFRIES	F325	OUT	Biometric Identification
12/04/2017 14:17:25	Denied	(unknown user)	F325	No key	Biometric Identification
12/04/2017 14:16:06	Granted	(unknown user)	F325	IN	Biometric Identification
12/04/2017 14:13:45	Granted	ALEX JEFFRIES	F325	OUT	Biometric Identification
12/04/2017 14:13:24	Granted	ALEX JEFFRIES	F325	IN	Biometric Identification

Figure 70 Commercial System Reports

The developed system records the matching score and identification result into the developed Fingerprint Application database which can then be used to generate the exam office reports.

The final section of this chapter highlights the results obtained from the authentication tests with both systems.

5.3.1 Test Results – Performance

Before commencing tests using the commercial system, a class list for each of the sessions mentioned in Table 7 were obtained from the three Schools. These lists were then used to produce the exam class list for each testing session. Table 10 shows the results from the tests with the commercial system and Table 11 presents the results with the developed system.

Table 10 Matching Scores from Tests of Commercial System

No.	Matching Score
2.	7400
3.	7209
5.	7819
6.	10598
9.	10398
10.	9896
11.	9677
13.	10363
17.	10557
18.	7636
19.	11418
20.	6686
22.	10456
24.	10385
26.	6723
27.	11278
31.	10126
35.	9930
36.	6495
37.	9805
38.	9769

Matching scores for 17 students were not recorded.

Table 11 Matching Scores from Test 1 of Developed System Before Enhancements

No.	Matching Score
1.	13873
2.	9297
3.	11918
4.	11691
5.	18014
6.	11907
7.	14774
8.	15043
9.	12936
10.	10527
11.	10771
12.	15863
13.	9702

The developed system was also tested using a verification process to know the outcome of an imposter. With student number 8, his/her template was used to match against two members of staff and the matching score was '1000' for both staff. When the template of student number 9 was loaded and used to match against a member of staff, the matching score returned was '1023'. The same tests were repeated with the templates of student numbers 12 and 13 which returned a matching score of '1263' and '1040' respectively. All these tests results confirmed that the imposters were rejected on the system and this portrays characteristics of a good fingerprint system.

The system was tested again, once all developments were completed. Table 12 shows results from the final test carried out with the developed system.

Table 12 Matching Scores from Test 2 of Developed System after Enhancements

No.	Matching Score
1.	13447
2.	14096
3.	9457
4.	14989
5.	15409
6.	13937
7.	13363
8.	13438

The average matching scores between prints of the same finger for the two systems are 9268 for the commercial system and 13069 for the one developed for this project (across both tests). The variance in the scores is 2641431.46 (17.5%) for the commercial system and 5299530.73 (17.6%) for the developed system. The nearly identical score variance values generated by the two systems indicate that the developed system is able to capture the same amount of diversity in fingerprints as the commercial system.

Before the tests were carried out, it was anticipated that the commercial system would have better matching rates as it captures one fingerprint four times but when compared to the system which has been developed, this is not true. As seen in

Matching scores for 17 students were not recorded.

Table 11, the highest recorded matching score for the developed system is 18014 for test 1 and 15409 for test 2 with a live quality score of 132. The highest recorded matching score for the commercial system is 11418 with a live quality score of 92. When the threshold is reduced, a lower matching score would be accepted and fewer genuine attempts would be rejected, whilst a higher threshold requires a higher matching score so more imposters would be rejected. At the same time, genuine attempts may also be falsely classified as non-matches [58] by a high threshold.

Further offline experiment was carried out to analyse the performance of the two systems against imposters. The matching scores shown in Table 13 were generated by verifying an imposter against each of the registered templates captured from the second test with the developed system.

Table 13 Matching Scores of Imposter from Developed System

No.	Matching Score against Threshold= 2000 (Value of Developed System)	Matching Score against Threshold= 3500 (Default Value of Developed System)	Matching Score against Threshold= 4000 (Default Value of Commercial System)
1.	1271	2197	1216
2.	961	686	736
3.	706	1189	525
4.	774	1521	538
5.	905	1513	944
6.	912	1096	974
7.	1407	1359	1411
8.	1753	1420	1113

The average matching scores between prints of different fingers for the developed and commercial systems are 1086 and 932 respectively. The variance in the scores is 128937.26 and 100580.98 respectively.

The quoted False Acceptance Rate (FAR) of the commercial system according to suppliers is 10^{-5} where Threshold=4000 so FAR probability < 0.001% making it a highly secured application.

The FAR of the developed system according to suppliers is 10^{-1} where Threshold=2000 so FAR probability < 2%. The threshold of 2000 should only be used for testing purposes as used in this project.

A False Acceptance Rate (FAR) is calculated as:

$$\text{FAR} = \frac{\text{Number of False Acceptance}}{\text{Total Number of Acceptance}} \quad (15)$$

A False Reject Rate is calculated as:

$$\text{FRR} = \frac{\text{Number of False Rejections}}{\text{Total Number of Rejections}} \quad (16)$$

This equation was applied to calculate FAR and FRR from statistics obtained from the tests carried out within this research work. The $\text{FAR} = 0$ for both systems with the threshold values used.

As there weren't any false rejections, FRR for both systems is 0. In fact, any threshold value within the range of [1753, 9297] would produce the same results for the developed system. This makes the developed system highly reliable.

Low threshold is preferred where convenience to users is at a higher priority level but this produces higher FAR hence affecting the security of a system. If imposter score > threshold, the FRR increases but decreases the FAR rate.

A threshold of 4000 would be most suitable for the exam registration process as accuracy plays a vital role in this process. The vast gap between an imposter's matching score and genuine user's matching score as shown across Tables 10 to 13 also confirms the high level of uniqueness and reliability of the fingerprint recognition system.

5.4 User Feedback – Acceptability

This subsection highlights feedback received from users after using both the commercial system and the developed system.

Following each of the test sessions, students provided feedback on the commercial system.

As shown in Figure 71, feedback from users of the developed system indicates that 100% of users strongly agreed that the system was easy to use based on the first test and 99% of users stated the same in the second test once the developed system was enhanced as seen in Figure 72. The system was tested by students taking the 6ENT1031 module before (Figure 71) and after making enhancements (Figure 72). Before the system was enhanced, fingerprint templates were being inserted into an XML file but following enhancements, these were stored into a database, hence increasing the enrolment time by a fraction of a second. This explains the drop of 1% in the feedback received for 'ease of use' question in the feedback received following second test.

In comparison, feedback received from users of commercial system as shown in Figure 73 indicates that only 76% of the users found it easy to use. This is because the commercial system captures one fingerprint four times and they have to repeat this process once more for the other index finger.

Some written feedback was also received in the 'Additional Comments' section for second test session of the developed system, for example:

- "The speed of enrolment is a little bit slow, but this issue can be solved by upgrading application. Anyway, I like this system".
- "The form should be redesigned (sign up) as it is a bit confusing and should be made more clear". Student was referring to the consent form as shown in Appendix XVI.

By comparing feedback received following all tests carried out on both the commercial system and the developed system, the feedback confirms that the developed system is better as all of the users indicated that they 'strongly agree' or 'agree' with the statement: "I feel comfortable using a fingerprint system". They also prefer using fingerprint recognition system instead of an ID card for exams and attendance monitoring processes.

All feedback and results will be shared with the project focus group for decision making.

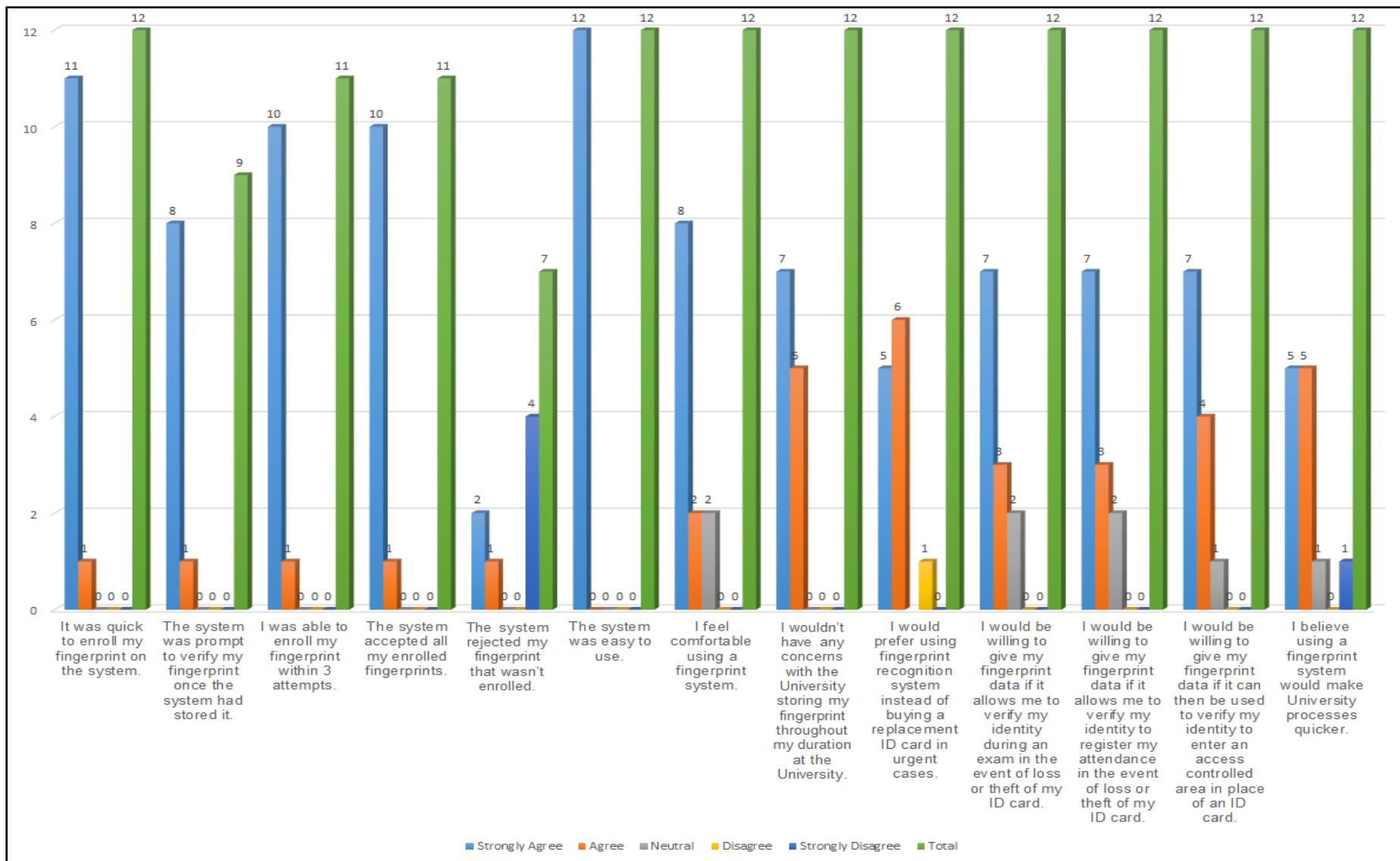


Figure 71 Feedback of the Developed System from Test 1

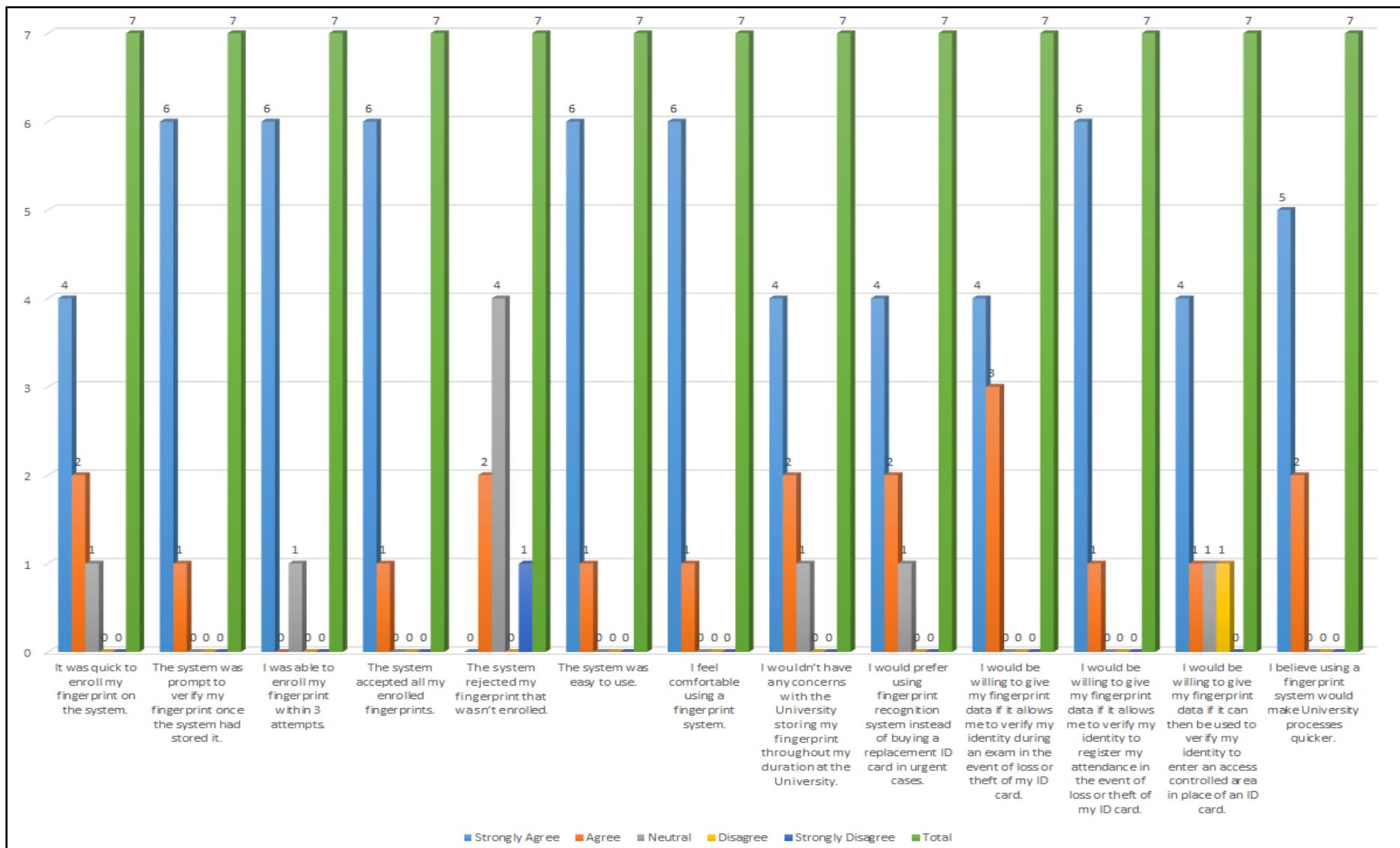


Figure 72 Feedback of the Developed System from Test 2

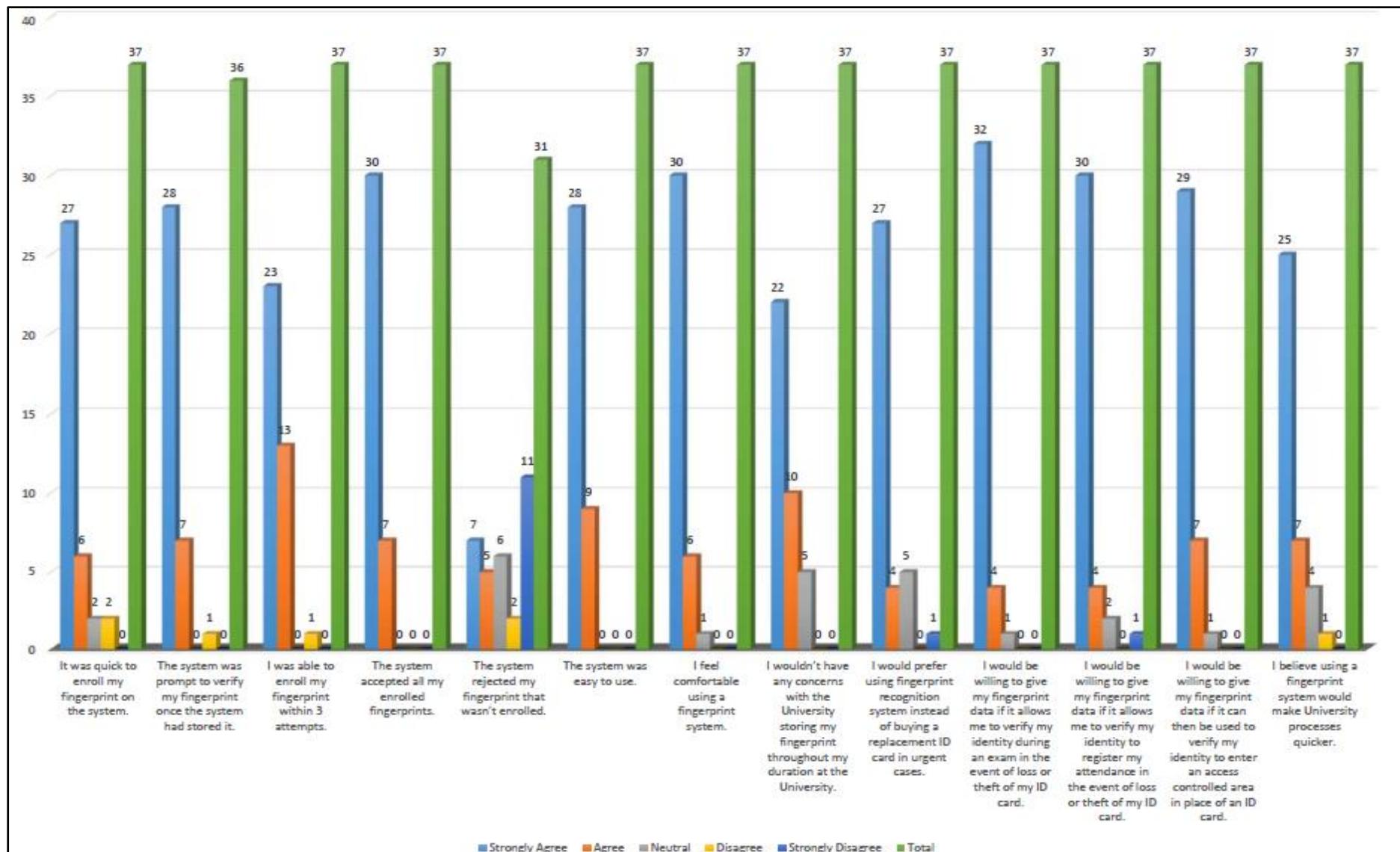


Figure 73 Feedback from System Users of Commercial System

CHAPTER 6: RECOMMENDATION AND CONCLUSION

Based on findings from this project and analysing feedback gathered from various tests, the following recommendations are proposed for deployment in a University environment:

- Enrolment process should be integrated in the registration process. Fingerprints should be captured when students arrive to register for their course in their first year. Trying to get students to enrol fingerprints after they have registered will not get much uptake as students may not have the time to provide these between lectures, labs and tutorials as identified in this research project.
- For applications with an enrolment or authentication process involving multiple users, touchless fingerprint technology should be tested and considered. This is likely to increase the acceptance rate amongst users as the hygiene issue would be addressed. If using the contact fingerprint readers, the devices should be cleaned as part of the daily/routine cleaning cycle. Before purchasing the fingerprint readers, advice on cleaning process should be sought from vendors.
- Fingerprints from at least two fingers should be enrolled, one from each hand as this would provide backup in case of any issues during authentication stage.
- For an organisation with a large group of users like UH, attempts should be made to avoid identification against the entire population.
- Explicit consent should be obtained from all users who should sign a consent form. The consent form should clearly state:
 - Purpose of capturing fingerprints.
 - How and when these will be collected.
 - Who else will have access to this information and why.
 - What the access rules are for other authorities such as law enforcement agencies.
 - How fingerprints will be stored i.e. raw images or templates, whether encrypted or not, stored offsite or elsewhere etc.
 - How long these fingerprints will be stored for.
 - When these will be deleted. For example, when they leave the organisation or within a certain period afterwards.
 - How users will be informed in case of privacy breach, solutions put in place for these breaches and liabilities the system owners would have.
 - Contact details should the user wish to request immediate deletion of their fingerprint(s) or have any queries.
 - If fingerprints of minors are required, what regulations have been put in place to protect them.
- If fingerprints are being collected, the company policy should clearly state this and provide reasons and guidelines as above.
- Fingerprints should be deleted when no longer required for the purpose stated.

- Anti-spoof fingerprint system should be implemented to prevent unauthorised access.
- Fingerprints/templates should NEVER leave the country where they were captured. If these need to be shared for whatever reason, the privacy laws of relevant countries should be abided.

In this fast-paced daily routine where most of us are in a rush, whether it is to get to work, catch a plane, attend a meeting, pick up kids from school or getting to an exam, we often forget to take something which is important to us; be it keys, books, USB or even important identity documents. This thesis has exhibited the use of a biometric feature which a user has with them at all times whilst not having to remember to carry it with them – their fingers!

A novel approach of using a fingerprint recognition system instead of ID cards in exams processes has been demonstrated in this research project. Developing a reliable fingerprint recognition system was not the only objective of this research work. Deploying such a system within a University environment, merging it with the relevant existing University systems and making University processes (in particular the exam registration process) more efficient and more reliable were also the core objectives of this project.

The objectives of this research work: To provide “*A Biometric Approach to Prevent False Use of IDs*” and testing this novel approach during exams, have both been met.

Thanks to funding received from University of Hertfordshire, a Software Development Kit (SDK) was bought to gain an understanding of fingerprint technology whilst enhancing it to achieve the above mentioned project objectives. An industry standard commercial system was also purchased for deployment and performance comparison.

Survey results highlighted two areas of concerns for users of fingerprint recognition system:

- Hygiene – Even though door handles that we touch everyday have germs on them, target users are not keen on touching fingerprint scanners in the fear of catching unwanted germs.
- Privacy – Unfortunately, some target users still associate fingerprint technology with crime investigation.

The privacy issue was addressed by not storing raw fingerprint images but rather storing templates on a password-protected secure server. Recommendations on the maintenance of fingerprint devices have been given in Chapter 6, which have been closely followed in this project to address the hygiene issue mentioned above.

System test results revealed:

- The commercial system takes longer to enrol as it requires two fingers and captures the print of each finger four times. Average time taken by the commercial system just to enrol fingerprint is 40 seconds. In comparison, the developed system only takes an average of 3 seconds to enrol.
- The developed system is better, evidenced by the higher matching scores generated. A bigger gap between matching scores of the same finger and imposter finger allows more flexibility with the set threshold and the ability to achieve a more secured system with a high threshold value.

- Feedback received from users reveal that the developed system is user-friendly with 100% stating the system is easy to use

Like most systems and processes, there are limitations. Due to budget constraints, a touchless fingerprint reader could not be purchased within the scope of this project but it is in our interest to further explore touchless fingerprint system before deploying the fingerprint recognition technology within our University. Fingerprint enrolment with a larger group of students can be carried out during the University's registration sessions and further testing can be carried out during the exam period before decision-making.

REFERENCES

- [1] A. K. Jain, D. Maio, D. Maltoni, and S. Prabhakar, "Handbook of Fingerprint Recognition," 2003.
- [2] (2014, 28/05/2018). *Biometrics in Schools* Available:
https://www.bigbrotherwatch.org.uk/files/reports/Biometrics_final.pdf
- [3] "Advantages and Disadvantages of technologies," Available:
<http://biometrics.pbworks.com/w/page/14811349/Advantages%20and%20disadvantages%20of%20technologies>
- [4] I. Bouchrika, "A Survey of Using Biometrics for Smart Visual Surveillance: Gait Recognition," in *Surveillance in Action*: Springer International Publishing, 2018, pp. 3-23.
- [5] N. S. a. T. C. (NSTC), "Iris Recognition," pp. 114-118Accessed on: 22/05/2018Available:
https://www.fbi.gov/file-repository/about-us-cjis-fingerprints_biotometrics-biometric-center-of-excellences-iris-recognition.pdf/view
- [6] P. Khaw, "Iris Recognition Technology for Improved Authentication," Available:
https://www.researchgate.net/publication/242754664_Iris_Recognition_Technology_for_Improved_Authentication
- [7] "25 Advantages and Disadvantages of Iris Recognition," Available:
<https://biometrictoday.com/25-advantages-disadvantages-iris-recognition/>
- [8] "Pros and Cons of Facial Recognition Technology For Your Business," Accessed on: 29/05/2018Available: <https://www.upwork.com/hiring/for-clients/pros-cons-facial-recognition-technology-business/>
- [9] P. Verma, Y. Bahendwar, A. Sahu, and M. Dubey, "Feature Extraction Algorithm of Fingerprint Recognition," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 10.
- [10] M. S. Al-Ani, "A Novel Thinning Algorithm for Fingerprint Recognition," *International Journal of Engineering Sciences*, 2013.
- [11] Unknown, "BIOMETRIC TECHNOLOGIES," Available: <https://www.fingerprints.com/>
- [12] K. UCHIDA, "Detection and Recognition Technologies Fingerprint Identification," *NEC Journal of Advanced Technology*, vol. 2, no. 1, 2005.
- [13] S. S. Ponnarasi and M. Rajaram, "Impact of Algorithms for the Extraction of Minutiae Points in Fingerprint Biometrics," *Journal of Computer Sciences*, vol. 8, 2012.
- [14] U. Rajanna, A. Erol, and G. Bebis, "A comparative study on feature extraction for fingerprint classification and performance improvements using rank-level fusion," *Pattern Anal Applic*, 2008.
- [15] A. M. Bazen, "Fingerprint Identification - Feature Extraction, Matching, and Database Search," 2002.
- [16] T.Trimpe. (2009). *Fingerprint Basics*. Available:
<http://sciencespot.net/Media/FrnsScience/fingerprintbasicscard.pdf>
- [17] A. Patel, V. Agrawal, and V. H. Shah, "Improve Fingerprint and Recognition Using Both Minutiae Based and Pattern Based Method," *ORIENTAL JOURNAL OF COMPUTER SCIENCE & TECHNOLOGY*, 2014.
- [18] A. Jain, "50 Years of Biometric Research: Almost Solved, The Unsolved, and The Unexplored," ed: Michigan State University, 2013, p. 44.
- [19] L. Qiu, "FINGERPRINT SENSOR TECHNOLOGY " presented at the IEEE 9th Conference on Industrial Electronics and Applications (ICIEA), 2014.
- [20] D. S. Hsu, "Fingerprint Sensor Technology And Security Requirements," Available:
<http://semiengineering.com/fingerprint-senor-technology-and-security-requirements/>
- [21] P. Gupta and P. Gupta, "An accurate slap fingerprint based verification system," *Neurocomputing*, vol. 188, pp. 178-189, 2016.
- [22] E. C. Lee, H. Jung, and D. Kim, "New Finger Biometric Method Using Near Infrared Imaging," *sensors*, pp. 2319-2333, 2011.

- [23] J. Shah and U. Poshiya, "TOUCHLESS FINGERPRINT RECOGNITION," *Asian Journal of Computer Science And Information Technology*, vol. 3, no. 5, pp. 73-76, 2013.
- [24] T. Meister, "Touchless 3D - A New Dimension in Fingerprint Technology," pp. 1-18 Available: <https://www.slideshare.net/securitysession/touchless-fingerprint-recognition>
- [25] A. COSTE, "University of Utah: CS6640 Image Processing Report, Project 1: Histograms," Available: http://www.sci.utah.edu/~acoste/uou/Image/project1/Arthur_COSTE_Project_1_report.html
- [26] N. H. Barnouti, "Fingerprint Recognition Improvement Using Histogram Equalization and Compression Methods," *International Journal of Engineering Research and General Science*, vol. 4, no. 2, pp. 685-692, 2016.
- [27] R. Vignesh, "What is the meaning of Fourier transform of an image ? Why is it important in image processing?," Accessed on: 02/06/2018 Available: <https://www.quora.com/What-is-the-meaning-of-Fourier-transform-of-an-image-Why-is-it-important-in-image-processing>
- [28] N. Singh and S. Rani, "Fingerprint Recognition and Identification by Minutiae Detection Phase Spectrum Analysis Algorith," *International Journal of Software and Web Sciences (IJSWS)*, 2013.
- [29] S. Bharadwaj, M. Vatsa, and R. Singh, "Biometric quality: a review of fingerprint, iris, and face," *EURASIP Journal on Image and Video Processing*, 2014.
- [30] R. Bansal, P. Sehgal, and P. Bedi, "Minutiae Extraction from Fingerprint Images - a Review," *International Journal of Computer Science Issues*, vol. 8, no. 5, 2011.
- [31] "Fingerprint Identification," Available: http://www.realtimenorthamerica.com/download/Fingerprint_Identification.pdf
- [32] H. H. Ahmed, K. Hamdy N, M. S. Tolba, and M. A. ELRashidy, "Comparitive Study of Various Techniques of Fingerprint Recognition Systems," *International Journal of Computing and Network Technology*, vol. 3, no. 3, 2015.
- [33] A. J. Bertino, Forensic Science: Fundamentals and Investigations 2012 Update, 2012. [Online]. Available: http://ngl.cengage.com/search/productOverview.do?N=201+4294918536&Ntk=NGL%7CP_EPI&Ntt=1367707560133261614320027647301534107526&Ntx=mode%2Bmatchallpartial.
- [34] M. JAMPOUR, M. YAGHOobi, and M. ASHOURZADEH, "A new fast technique for fingerprint identification with fractal and chaos game theory," *Fractals*, vol. 18, no. 3, pp. 293-300, 2009.
- [35] D. Kocharyan and H. Sarukhanyan, "Feature Extraction Techniques and Minutiae-Based Fingerprint Recognition Process," Available: <https://archive.org/stream/FeatureExtractionTechniquesAndMinutiae-basedFingerprintRecognitionProcess/34-39#page/n0/mode/1up>
- [36] N. Zaeri, "Minutiae-based Fingerprint Extraction and Recognition," *InTech*, 2011.
- [37] S. Z. Li and A. Jain, Encyclopedia of Biometrics Springer, 2009. [Online]. Available: http://link.springer.com/referenceworkentry/10.1007%2F978-0-387-73003-5_394.
- [38] D. Maio and D. Maltoni, "Direct Gray-Scale Minutiae Detection In Fingerprints," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 19, no. 1, 1997.
- [39] L. Ji and Z. Yi, "Fingerprint orientation field estimation using ridge projection," *PATTERN RECOGNITION*, vol. 41, pp. 1491-1503, 2008.
- [40] J. Zhou and J. Gu, "A Model-Based Method for the Computation of Fingerprints' Orientation Field," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 13, no. 6, pp. 821-835, 2004.
- [41] S. Sharma and S. M. Ratna, "Fingerprint Matching," Available: <http://slideplayer.com/slide/13006989/>
- [42] I. G. Babatunde, Akinyokun, O. Charles, and A. B. Kayode, "A Modified Approach to Crossing Number and Post-processing Algorithms for Fingerprint Minutiae Extraction and Validation," *IMS Manthan (The Journal of Mgt., Comp. Science & Journalism)* vol. 6, no. 1, 2011.
- [43] P. A. Kumari and G. JayaSuma, *A Comparative Study of Various Minutiae Extraction Methods for Fingerprint Recognition Based on Score Level Fusion*. Springer, Singapore, 2016.

- [44] A. K. Jain, J. Feng, and K. Nandakumar, "Fingerprint Matching," *IEEE Computer Society*, vol. 43, no. 2, pp. 36-44, 2010.
- [45] D. Thakkar, "Minutiae Based Extraction in Fingerprint Recognition," Available: <https://www.bayometric.com/minutiae-based-extraction-fingerprint-recognition/>
- [46] S. C. A. S. P. I. T. Force, "Smart Cards and Biometrics in Privacy-Sensitive Secure Personal Identification Systems," vol. 10, no. 5, Available: https://www.securetechalliance.org/newsletter/may_2005/feature_0505.html
- [47] S. Chikkerur, "Online Fingerprint Verification," Available: <http://slideplayer.com/slide/3851592/>
- [48] S. Narwal and D. Kaur, "Comparison between Minutiae Based and Pattern Based Algorithm of Fingerprint Image" *I.J. Information Engineering and Electronic Busines*, vol. 2, pp. 23-29, 2016.
- [49] N. Celik, N. Manivannan, W. Balachandran, and S. Kosunalp, "Multimodal Biometrics for Robust Fusion Systems using Logic Gates," *Journal of Biometrics & Biostatistics*, vol. 6, no. 1, pp. 1-6, 2015.
- [50] A. K. Jain and K. Nandakumar, "Local Correlation-based Fingerprint Matching," vol. ICVGIP, 2004.
- [51] "Fingerprint Verification," Available: <http://www.dds.co.jp/en/fv/algorith.html>
- [52] NIST, "NIST Special Database 4," *NIST 8-Bit Gray Scale Images of Fingerprint Image Groups(FIGS)*,
- [53] N. L. o. P. R., "Center for Biometrics and Security Research. CASIA-FingerprintV5," Available: <http://biometrics.idealtest.org/dbDetailForUser.do?id=7>
- [54] F. F.-L. a. D. B. L. Bong, "FingerDOS: A fingerprint database based on optical sensor," *WSEAS Transactions on Information Science and Applications*, vol. 12, no. 29, pp. 297-304, 2015.
- [55] A. T. A. Taneja, A. Malhotra, A. Sankaran, M. Vatsa, and R. Singh, "Fingerphoto Spoofing in Mobile Devices: A Preliminary Study," presented at the International Conference on Biometrics: Theory, Applications and Systems, 2016.
- [56] A. M. A. Sankaran, A. Mittal, M. Vatsa, and R. Singh, "On smartphone camera based fingerphoto authentication," *International Conference on Biometrics: Theory, Applications and Systems*, pp. 1-7, 2015.
- [57] (29/05/2018). IDEMIA Morpho. Available: <https://usa.morpho.com/biometric-terminals/software-applications-and-development-kits>
- [58] "UNDERSTANDING BIOMETRIC PERFORMANCE EVALUATION," Available: <https://precisebiometrics.com/wp-content/uploads/2014/11/White-Paper-Understanding-Biometric-Performance-Evaluation.pdf>

APPENDIX I: SAMPLE FORM TO RECORD ABSENT AND EXTRA STUDENTS



Invigilator: place this copy in the Exams Folder

ABSENT & EXTRA CANDIDATES FORM

Room:	Day:	Date:	Scheduled Start Time:

Invigilator: Use a separate form for every exam

Exam Code:	Exam Title:

*to be supplied by Exams & Awards Office

ABSENTEES		EXTRAS		
	Candidate Name	Exam Number	Candidate Name	Student Number
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				

Page ____ of ____ (Invigilator to complete)

APPENDIX II: UH SCREENS BROADCAST REQUEST



UHScreens Broadcast Request

We will try to put your designs up as and when specified, and will let you know if we can't for any reason.

To: [REDACTED]

From: Rupa Patel

Tel: [REDACTED]

Email: [REDACTED]

Date: 04/12/2016

School/SBU:

Timings

Date to start (please give 48 hours' notice)	09/12/2016
Date to finish (maximum 60 days, unless otherwise agreed)	22/12/2016
Audience (Students, staff and/or public)	Students

Checklist for successful TVTools production

Please ensure you have checked the following, otherwise your design will not be broadcast.

- Powerpoint files should be supplied at a custom size of 33.86 cm x 14.66 cm, or use the template provided on StaffNet
- Video files should be supplied in WMV or MP4, only one video can be played at a time. Alternatively videos can be streamed from YouTube. Sound will not be used.
- Images should be provided as a JPEG or PNG (96 dpi, 1280 pixels by 554 pixels)
- Animation can be in: Quicktime (MOV), AVI, TGA
- Text on a slide needs to be easy to read from a distance (minimum font size 20 point in Powerpoint).
- Text – needs to be clear, direct and free of jargon.
- No University logo
- Times need to be displayed in 24 hour clock format

Please keep a copy of your information for reference and future use as we cannot guarantee to file your material.

APPENDIX III: FINGERPRINT SURVEY METHODS

Survey Link on StudyNet – Reminder:

My StudyNet ▾ My Course Online Library Help & Support

University News

 Attention All Students - Final Chance!
Rupa_4 Patel on the 20/12/2016 17:25:17

Fingerprint Recognition System

As part of a research project, we would like to gather your thoughts on using fingerprint recognition system to improve processes at UH.

It will take less than 5 minutes to complete. Your feedback is greatly appreciated, thank you.

Survey closes on Thursday, 22nd December 2016 so please complete the survey if you haven't done so already.

Share your thoughts now!

This research project has received Ethics approval from the Science and Technology ECDA committee. Protocol number: ENT/PGR/UH/02043.

Wishing you all Happy Holidays

Survey Promoted on UH Facebook Page:

University of Hertfordshire 
17 December 2016 · 

We would like to gather your thoughts on using fingerprint recognition system to improve processes at the University. The survey consists of 7 questions and will take less than 5 minutes to complete. Your feedback is greatly appreciated. Thank you!
<http://hnts.me/SBdnA>
(This research project has received Ethics approval from the Science and Technology ECDA committee. Protocol number: ENT/PGR/UH/02043)



34 Likes 4 Comments 6 Shares

Survey Promoted Using Table Signs Placed in Social Areas and Student Centre:

**Please share your
thoughts on Fingerprint
Technology**

Via:

StudyNet News Feed Post, Titled:
'Attention All Students'

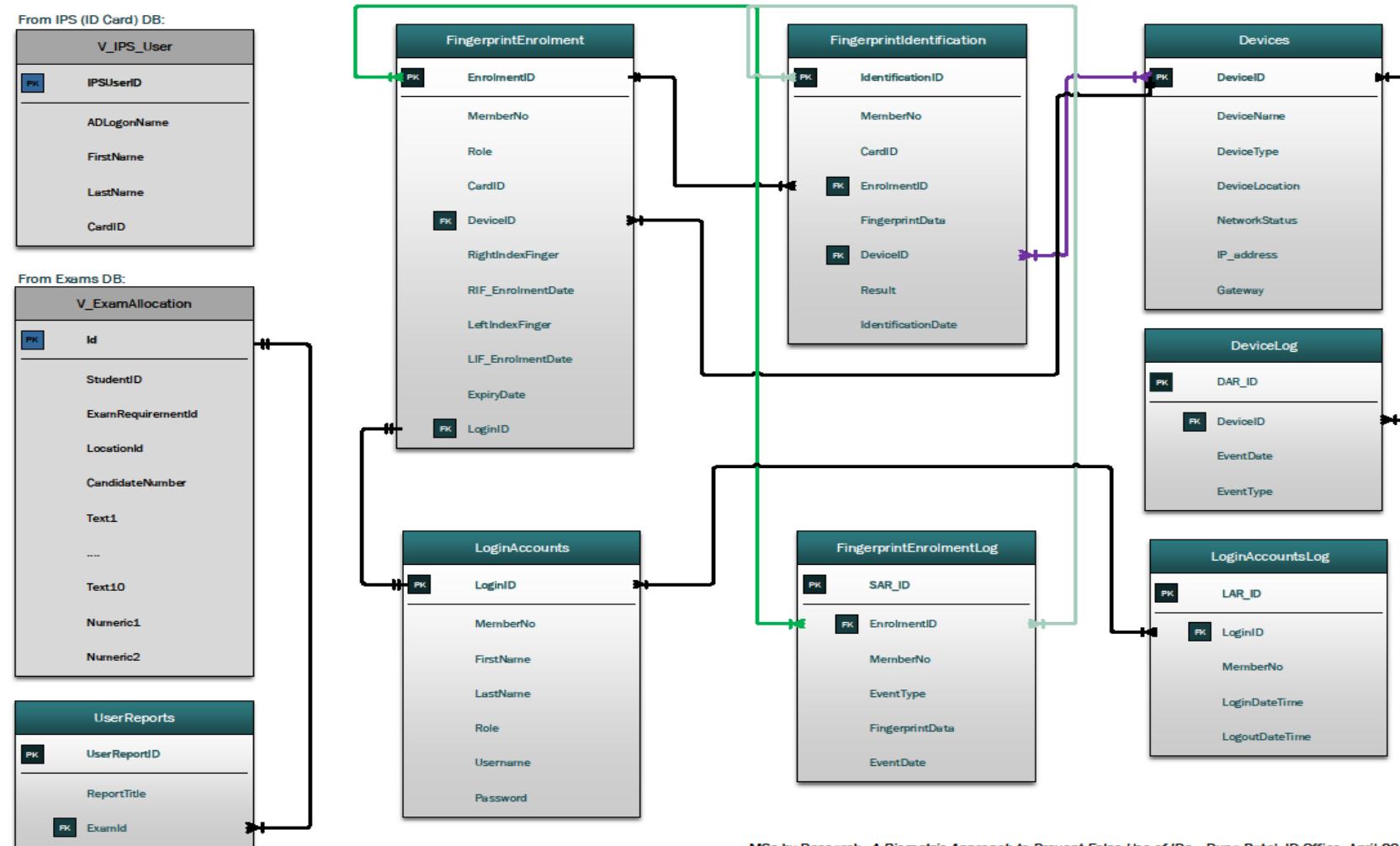
posted on 05/12/2016

Or

<https://www.surveymonkey.co.uk/r/FingerprintTechnology>

Thank you!

APPENDIX IV: ENTITY RELATIONSHIP DIAGRAM (ERD) V1



MSc by Research, A Biometric Approach to Prevent False Use of IDs. Rupa Patel, ID Office. April 2016

APPENDIX V: SQL STATEMENTS

For Exam Marker:

1. Query to retrieve list of students scheduled to take exam for a particular module:

```
SELECT HostKey AS ID_NUMBER, Numeric1 AS Candidate_Number  
INTO #TempTableList  
FROM esdev.rdreader.V_Student  
WHERE Id IN(  
    SELECT StudentId  
    FROM esdev.rdreader.V_ExamAllocation E  
    WHERE EXISTS  
        (SELECT F.ExamRequirementId  
        FROM esdev.rdreader.V_Examination F  
        WHERE E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '5BUS1094-0906%' AND  
        StartDateTime >= '2016-06-22'))
```

2. Query to retrieve list of students who have taken the relevant exam (i.e. '5BUS1094-0906') in the relevant location (i.e. 'Club DH'):

```
SELECT ADLogonName AS ID_Number, FirstName AS First_Name, LastName AS Last_Name  
INTO #TempTable  
FROM IDCardsystem.dbo.IPSUsers  
WHERE ADLogonName IN(  
    SELECT DISTINCT MemberNo  
    FROM FingerprintApplication.dbo.FingerprintIdentification  
    WHERE DeviceId IN(  
        SELECT DeviceID  
        FROM FingerprintApplication.dbo.DeviceList d INNER JOIN esdev.rdreader.V_Location  
        l  
        ON d.DeviceLocation=l.Description  
        WHERE EXISTS(  
            SELECT StudentId, ExamRoomLocationId  
            FROM esdev.rdreader.V_ExamAllocation E  
            WHERE EXISTS  
                (SELECT F.ExamRequirementId  
                FROM esdev.rdreader.V_Examination F  
                WHERE E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '5BUS1094-0906%' AND  
                StartDateTime >= '2016-06-22'))))
```

3. Query to retrieve list of absent students – those who didn't turn up for the exam.

```
SELECT Candidate_Number FROM #TempTableList  
WHERE ID_Number NOT IN(SELECT ID_NUMBER FROM #TempTable)
```

4. Final query to retrieve list of extra students – those who have not been scheduled to take the exam in the queried location but did end up taking the exam in that room/location.

```
SELECT DISTINCT Numeric1 AS CANDIDATE_NUMBER  
INTO #TempTableC  
FROM esdev.dbo.V_Student  
WHERE HostKey IN(  
    SELECT ID_Number  
    FROM #TempTable  
    WHERE EXISTS(  
        SELECT DISTINCT Numeric1, HostKey  
        FROM esdev.dbo.V_Student  
        WHERE HostKey=ID_Number)))
```

For Exams Office:

1. Query to retrieve list of students scheduled to take exam for a particular module (e.g. '5BUS1094-0906') during a particular exam period (e.g. referred/deferred exams on or after '2016-06-22').

```
SELECT HostKey AS ID_NUMBER, Description AS FULL_NAME  
INTO #TempTableList  
FROM esdev.rdreader.V_Student  
WHERE Id IN(  
    SELECT StudentId  
    FROM esdev.rdreader.V_ExamAllocation E  
    WHERE EXISTS  
        (SELECT F.ExamRequirementId  
        FROM esdev.rdreader.V_Examination F  
        WHERE E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '5BUS1094-0906%' AND  
        StartDateTime >= '2016-06-22'))
```

2. Query to retrieve list of students who have taken the relevant exam (i.e. '5BUS1094-0906') in the relevant location (i.e. 'Club DH').

```
SELECT ADLogonName AS ID_Number, FirstName AS First_Name, LastName AS Last_Name  
INTO #TempTable  
FROM IDCardSystem.dbo.IPSUsers  
WHERE ADLogonName IN(  
    SELECT DISTINCT MemberNo  
    FROM FingerprintApplication.dbo.FingerprintIdentification  
    WHERE DeviceId IN(  
        SELECT DeviceID  
        FROM FingerprintApplication.dbo.DeviceList d INNER JOIN esdev.rdreader.V_Location  
        l  
        ON d.DeviceLocation=l.Description  
        WHERE EXISTS(  
            SELECT StudentId, ExamRoomLocationId  
            FROM esdev.rdreader.V_ExamAllocation E  
            WHERE EXISTS  
                (SELECT F.ExamRequirementId  
                FROM esdev.rdreader.V_Examination F  
                WHERE E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '5BUS1094-0906%' AND  
                StartDateTime >= '2016-06-22'))))
```

3. Query to retrieve list of absent students – those who didn't turn up for the exam.

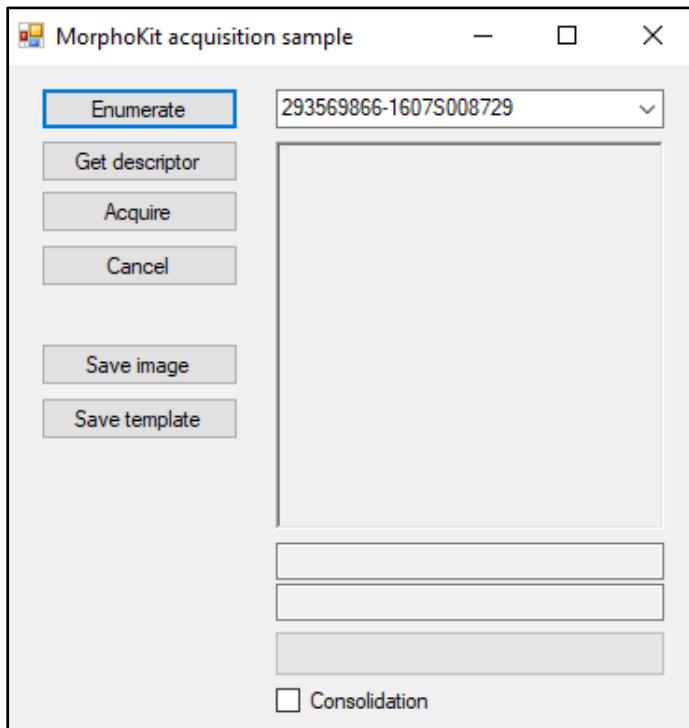
```
SELECT * FROM #TempTableList WHERE  
ID_Number NOT IN(SELECT ID_NUMBER FROM #TempTable)
```

4. Final query to retrieve list of extra students – those who weren't scheduled to take the exam in the queried location but did end up taking the exam in that room/location.

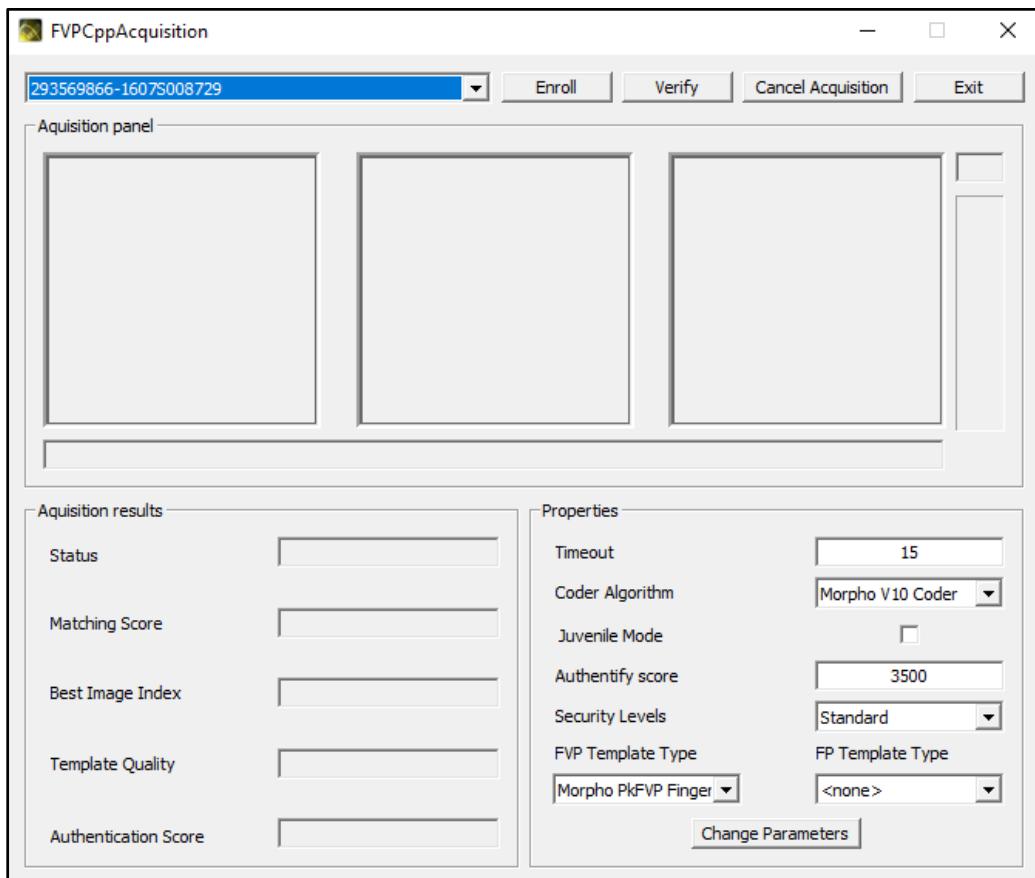
```
SELECT * FROM #TempTable  
WHERE ID_NUMBER NOT IN(SELECT ID_NUMBER FROM #TempTableList)
```

APPENDIX VI: SAMPLE SDK FINGERPRINT APPLICATIONS

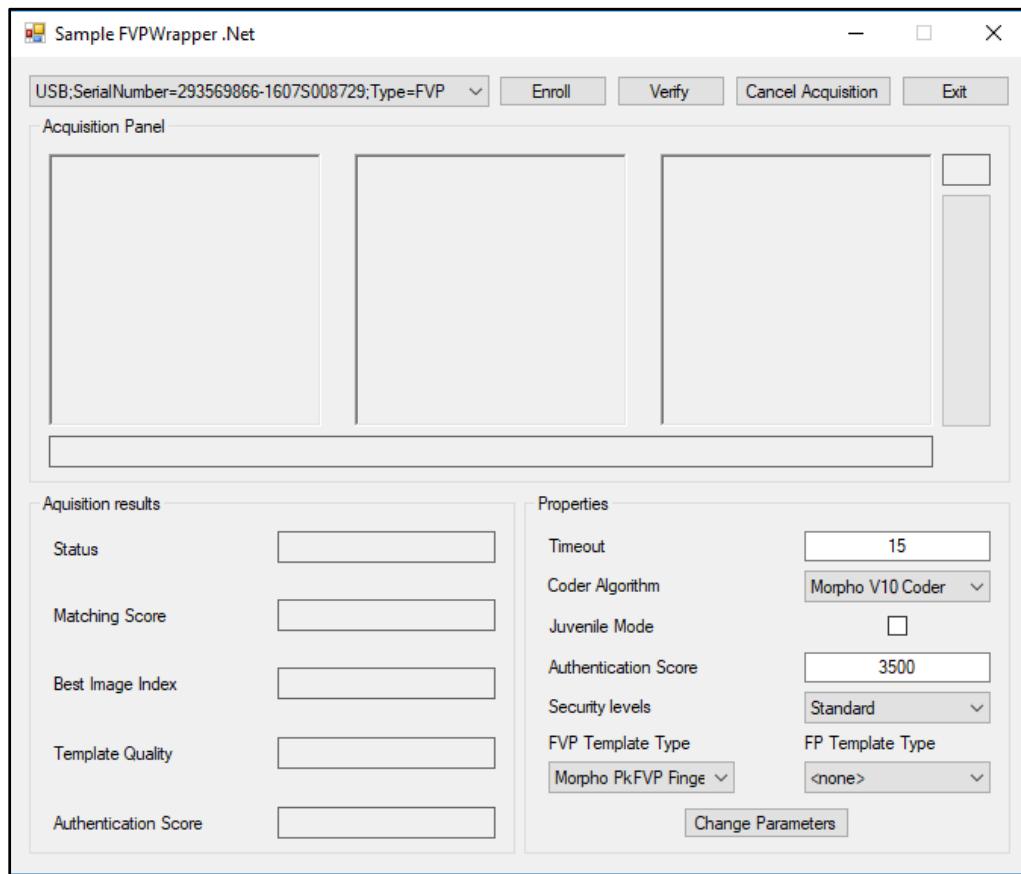
Application Name – Acquisition:



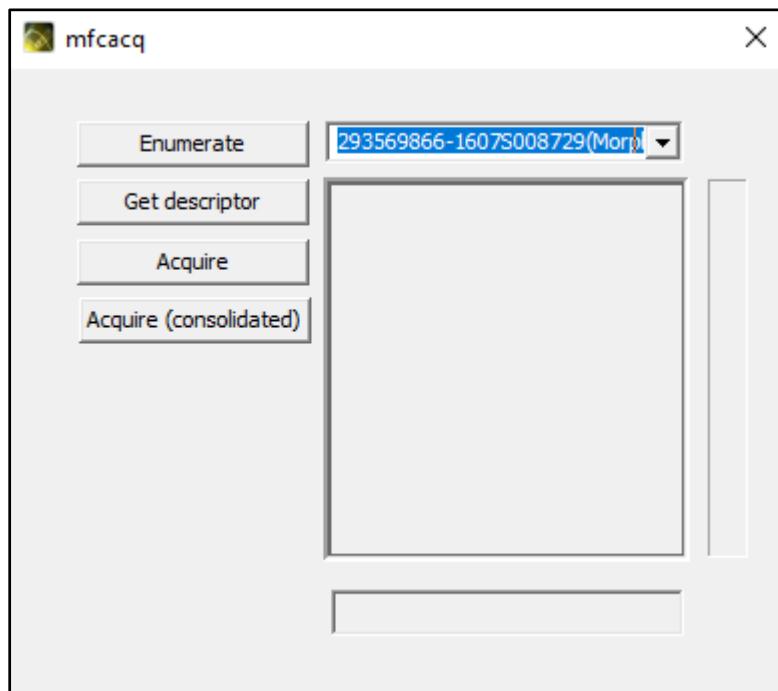
Application Name – FVPCppAcquisition:



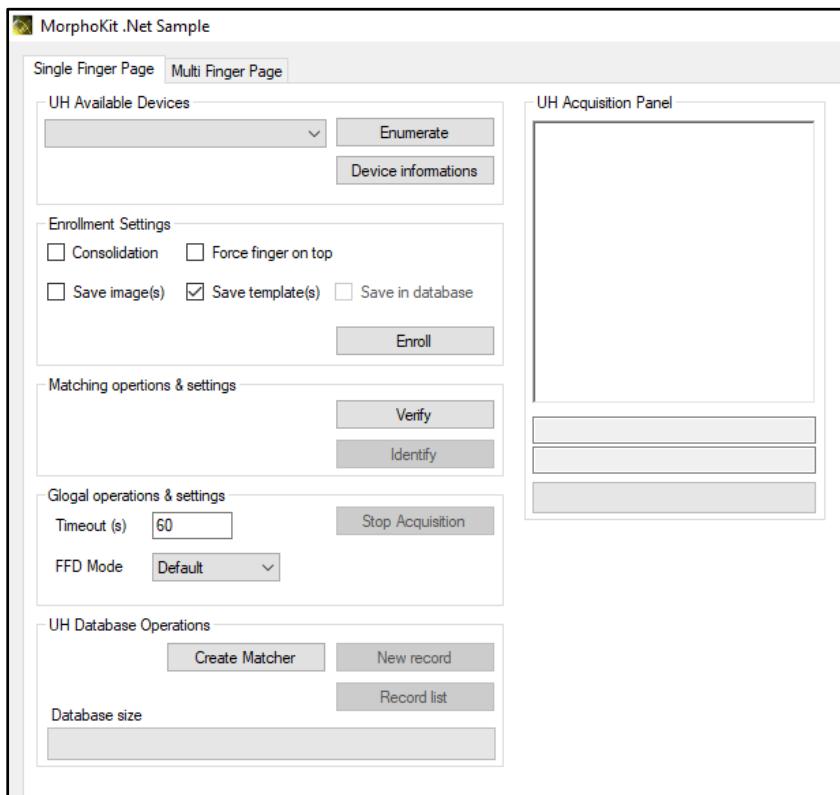
Application Name – FVPNetAcquisition:



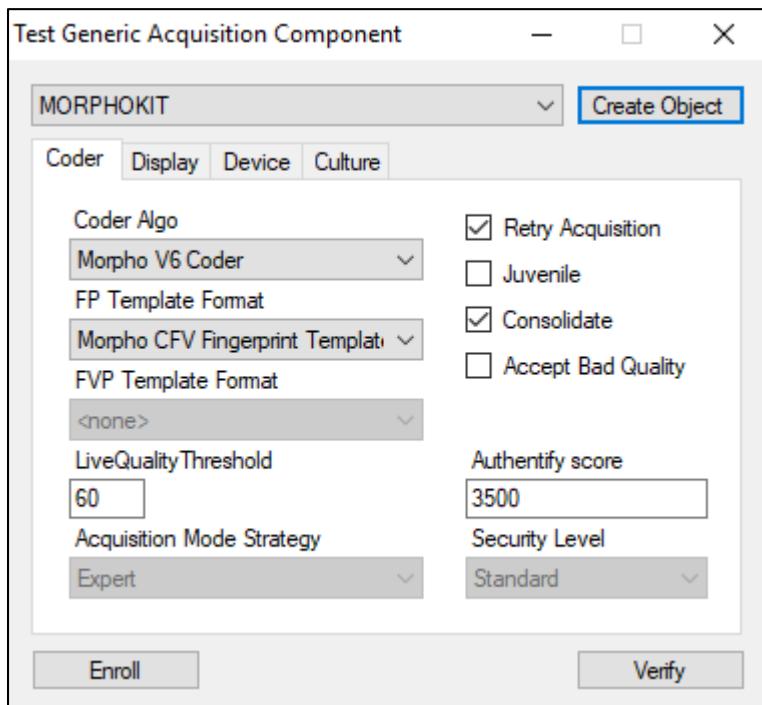
Application Name – mfcacq:



Application Name – MorphoKitDotNETSample:



Application Name – Sample_GenericAcquisitionComponent:



APPENDIX VII: FINGERPRINT APPLICATION RECORDLIST.CS CODE

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using MorphoKitDotNETSample.Core;
using Sagem.MorphoKit;
using System.IO;
using System.Xml.Serialization;
using System.Data.SqlClient;
using System.Data.OleDb;
using System.Data.SqlClient;

namespace MorphoKitDotNETSample.Controls
{
    public partial class RecordList : Form
    {
        private MatchingContext _matchingContext;

        public RecordList()
        {
            InitializeComponent();
            _matchingContext = null;
        }

        public MatchingContext CurrentMatchingContext
        {
            set
            {
                _matchingContext = value;
                if (_matchingContext != null)
                {
                    btn_refresh.PerformClick();
                }
            }
        }

        private void btn_cancel_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btn_refresh_Click(object sender, EventArgs e)
        {
            //lv_templates.Items.Clear();
            string[] ids = _matchingContext.GetRecordIds();
            foreach (string id in ids)
            {
                ListViewItem item = lv_templates.FindItemWithText(id);
                if (item == null)
                {
                    IRecord record = _matchingContext.FindRecord(id);
                    ListViewItem new_item = new ListViewItem(id);
                    new_item.SubItems.Add(Encoding.ASCII.GetString(record.Payload));
                    new_item.SubItems.Add(record.NumberOfTemplates.ToString());
                    lv_templates.Items.Add(new_item);
                }
            }
        }
    }
}
```

```

        }
    }

    private void btn_remove_Click(object sender, EventArgs e)
    {
        try
        {
            foreach (ListViewItem eachItem in lv_templates.SelectedItems)
            {
                _matchingContext.RemoveRecord(eachItem.Text);
                lv_templates.Items.Remove(eachItem);
            }
            btn_refresh.PerformClick();
        }
        catch (Exception exc)
        {
            MessageBox.Show(exc.Message, "Remove record", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }

    private void RemoveRecord_Load(object sender, EventArgs e)
    {
        btn_refresh.PerformClick();
    }

    private void btn_db_export_Click(object sender, EventArgs e)
    {
        SaveFileDialog fileDlg = new SaveFileDialog();
        fileDlg.AddExtension = true;

        fileDlg.FileName = "matcher_records";
        fileDlg.Title = "Export database records";
        fileDlg.DefaultExt = "xml";
        fileDlg.Filter = "XML file (*.xml)|*.xml|All Files (*.*)|*.*";
        if (fileDlg.ShowDialog() == DialogResult.OK)
        {
            try
            {
                XmlSerializer serializer = new XmlSerializer(typeof(DatabaseXml));
                TextWriter writer = new StreamWriter(fileDlg.FileName);

                DatabaseXml db = new DatabaseXml();
                List<RecordXml> records = new List<RecordXml>();
                string[] ids = _matchingContext.GetRecordIds();
                foreach (string id in ids)
                {
                    RecordXml record = new RecordXml();
                    IRecord irecord = _matchingContext.FindRecord(id);
                    record.Id = irecord.Id;
                    record.Payload = Encoding.ASCII.GetString(irecord.Payload);
                    List<TemplateXml> templates = new
List<TemplateXml>(irecord.NumberOfTemplates);
                    for (int i = 0; i < irecord.NumberOfTemplates; ++i)
                    {
                        TemplateXml template = new TemplateXml();
                        IFingerTemplate fingertemplate = irecord.GetTemplate(i);
                        template.FingerId = fingertemplate.Id;
                        template.Data = fingertemplate.Buffer;
                        templates.Add(template);
                    }
                    record.TemplateItems = templates.ToArray();
                    records.Add(record);
                }
            }
        }
    }
}

```

```

        }
        db.RecordList = records.ToArray();
        serializer.Serialize(writer, db);
        writer.Close();
    }

    catch (Exception exc)
    {
        MessageBox.Show(exc.Message, "Export Database",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
SqlConnection sqlConnection1 = new SqlConnection("Data Source=RUPA-
TOSH\\SQLEXPRESS;Initial Catalog=FingerprintApplication;Integrated Security=True");
SqlCommand cmd = new SqlCommand();
Int32 rowsAffected;

cmd.CommandText = "SP_ImportFingerprintData_XML";
cmd.CommandType = CommandType.StoredProcedure;
cmd.Connection = sqlConnection1;

sqlConnection1.Open();

rowsAffected = cmd.ExecuteNonQuery();

sqlConnection1.Close();
}

private void btn_db_import_Click(object sender, EventArgs e)
{
    OpenFileDialog fileDlg = new OpenFileDialog();
    fileDlg.AddExtension = true;
    fileDlg.Title = "Import database records";
    fileDlg.DefaultExt = "xml";
    fileDlg.Filter = "XML file (*.xml)|*.xml|All Files (*.*)|*.*";
    if (fileDlg.ShowDialog() == DialogResult.OK)
    {
        try
        {
            XmlSerializer serializer = new XmlSerializer(typeof(DatabaseXml));
            FileStream fs = new FileStream(fileDlg.FileName, FileMode.Open);
            DatabaseXml db = (DatabaseXml)serializer.Deserialize(fs);
            if (db.RecordList.Length == 0)
                return;

            foreach (RecordXml record in db.RecordList)
            {
                Record irecord = new Record();
                irecord.Id = record.Id;
                irecord.Payload = Encoding.ASCII.GetBytes(record.Payload);
                foreach (TemplateXml item in record.TemplateItems)
                {
                    IFingerTemplate template = new FingerTemplate();
                    template.Id = item.FingerId;
                    template.Buffer = item.Data;
                    irecord.AddTemplate(template);
                }
                try
                {
                    _matchingContext.AddRecord(irecord);
                }
                catch (Exception exc)
                {

```

```

        MessageBox.Show(exc.Message, String.Format("Add record
{0}", record.Id), MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
btn_refresh.PerformClick();
}
catch (Exception exc)
{
    MessageBox.Show(exc.Message, "Import Database",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
}

[XmlAttribute("MorphoKitDatabase", Namespace = "http://www.morpho.com",
IsNullable = false)]
public class DatabaseXml
{
    [XmlArrayAttribute("RecordList")]
    public RecordXml[] RecordList;
}

[XmlType("Record")]
public class RecordXml
{
    [XmlAttribute]
    public string Id;
    public string Payload;
    [XmlArrayAttribute("TemplateList")]
    public TemplateXml[] TemplateItems;
}

[XmlType("Template")]
public class TemplateXml
{
    [XmlAttribute]
    public byte FingerId;
    public byte[] Data;
}
}

```

APPENDIX VIII: FINGERPRINT APPLICATION MSOACQUISITIONSTRATEGY.CS CODE

```
using System;
using System.Collections.Generic;
using System.Text;
using Sagem.MorphoKit;
using System.Collections;
using System.Drawing;
using System.Drawing.Imaging;
using System.Windows.Forms;
using System.IO;
using System.Data;
using System.Data.SqlClient;
using System.Data.OleDb;
using System.Data.SqlClient;

namespace MorphoKitDotNETSample.Core
{
    public class MsoAcquisitionStrategy : IAcquisitionStrategy
    {
        private void DisplayStatusMessage(int status, string caption)
        {
            string error_msg = "";
            switch (status)
            {
                case -1:
                    error_msg = "Unknown acquisition status";
                    break;
                case -19:
                    error_msg = "The specified time was reached before the acquisition completes";
                    break;
                case -26:
                    error_msg = "Acquisition had been cancelled by user";
                    break;
                case -46:
                    error_msg = "The acquisition device had detected a false finger (only for acquisition devices with FFD feature)";
                    break;
                case -47:
                    error_msg = "The acquired finger can be too moist or the acquisition device is wet";
                    break;
                default:
                    error_msg = "The acquisition failed due to an internal error";
                    break;
            }
            MessageBox.Show(error_msg, caption, MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }

        #region IAcquisitionStrategy Members

        public void Enroll(MatchingContext matching_context)
        {
            try
            {
                _device.AcquisitionMode = AcquisitionMode.ENROLL;
                _device.TimeOut = _parameters.Timeout;
                _device.FakeFingerMode = _parameters.FakeFingerMode;
            }
        }
    }
}
```

```

        _device.ForceFingerOnTop = (_parameters as
MsoAcquisitionParameters).ForceFingerOnTop;
        if ((_parameters as MsoAcquisitionParameters).Consolidation)
{
    IConsolidatedAcquisitionResult consoAcquisRes =
_device.AcquireConsolidated(_currentDevice);
    if (consoAcquisRes.Status == 0)
{
    ICoder coder = new Coder();
    IConsolidationResult enrollConsoResult =
coder.EnrollConsolidated(
        consoAcquisRes.ImageBuffer1,
        consoAcquisRes.ImageBuffer2,
        consoAcquisRes.ImageBuffer3,
        consoAcquisRes.Width, consoAcquisRes.Height, 2);

    SaveImage(consoAcquisRes.ImageBuffer1 as byte[]);
    SaveTemplate(enrollConsoResult.Template as byte[]);
    if (matching_context != null)
{
        Record record = new Record();
        record.Id =
DateTime.Now.ToString();//Guid.NewGuid().ToString();
        record.Payload =
Encoding.ASCII.GetBytes(String.Format("Template quality {0}",
enrollConsoResult.Quality));
        IFingerTemplate template = new FingerTemplate();
        template.Id = 1;
        template.Buffer = enrollConsoResult.Template as byte[];
        record.AddTemplate(template);
        matching_context.AddRecord(record);
}
}
else
{
    DisplayStatusMessage(consoAcquisRes.Status, "Enrollment");
}
}
else
{
    IAquisitionResult acqresult = _device.Acquire(_currentDevice);
    if (acqresult.Status == 0)
{
    ICoder coder = new Coder();
    ICoderResult enrollResult = coder.Enroll(
        acqresult.ImageBuffer,
        acqresult.Width, acqresult.Height, 2);

    SaveImage(acqresult.ImageBuffer as byte[]);
    SaveTemplate(enrollResult.Template as byte[]);
    if (matching_context != null)
{
        Record record = new Record();
        record.Id =
DateTime.Now.ToString();//Guid.NewGuid().ToString();
        record.Payload =
Encoding.ASCII.GetBytes(String.Format("Template quality {0}", enrollResult.Quality));
        IFingerTemplate template = new FingerTemplate();
        template.Id = 1;
        template.Buffer = enrollResult.Template as byte[];
        record.AddTemplate(template);
        matching_context.AddRecord(record);
}
}
}
}

```

```

        else
        {
            DisplayStatusMessage(acqresult.Status, "Enrollment");
        }
    }
}
catch (Exception exc)
{
    MessageBox.Show(exc.Message, "Enrollment", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
}
}

public void Verfiy(IRecord record)
{
    try
    {
        _device.AcquisitionMode = AcquisitionMode.VERIFY;
        _device.TimeOut = _parameters.Timeout;
        _device.FakeFingerMode = _parameters.FakeFingerMode;
        _device.ForceFingerOnTop = (_parameters as
MsoAcquisitionParameters).ForceFingerOnTop;
        IAcquisitionResult acqresult = _device.Acquire(_currentDevice);
        if (acqresult.Status == 0)
        {
            ICoder coder = new Coder();
            ICoderResult enrollResult = coder.Enroll(
                acqresult.ImageBuffer,
                acqresult.Width, acqresult.Height, 2);
            IAuthenticator authenticator = new Authenticator();
            int score =
authenticator.Authenticate(record.GetTemplate(0).Buffer, enrollResult.Template);
            MessageBox.Show(String.Format("The resulted matching score : {0}",
score), "Verify",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            DisplayStatusMessage(acqresult.Status, "Verify");
        }
    }
    catch (Exception exc)
    {
        MessageBox.Show(exc.Message, "Verify", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
    }
}

public void Identify(MatchingContext matching_context)
{
    try
    {
        _device.AcquisitionMode = AcquisitionMode.VERIFY;
        _device.TimeOut = _parameters.Timeout;
        _device.FakeFingerMode = _parameters.FakeFingerMode;
        _device.ForceFingerOnTop = (_parameters as
MsoAcquisitionParameters).ForceFingerOnTop;
        IAcquisitionResult acqresult = _device.Acquire(_currentDevice);
        if (acqresult.Status == 0)
        {
            ICoder coder = new Coder();
            ICoderResult enrollResult = coder.Enroll(
                acqresult.ImageBuffer,

```

```

        acqresult.Width, acqresult.Height, 2);
IFingerTemplate template = new FingerTemplate();
template.Id = 0;
template.Buffer = enrollResult.Template;
Record record = new Record();
record.Id = "Temp";
record.AddTemplate(template);
ICandidate candidate = matching_context.Identify(record);
if (candidate.Score > 2000)
{
    System.Windows.Forms.Form f =
System.Windows.Forms.Application.OpenForms[" MainForm"];
    SqlConnection con = new SqlConnection();
    SqlConnection con1 = new SqlConnection();
    con.ConnectionString = "Data Source=RUPA-
TOSH\\SQLEXPRESS;Initial Catalog=FingerprintApplication;Integrated Security=True";
    con1.ConnectionString = "Data Source=RUPA-
TOSH\\SQLEXPRESS;Initial Catalog=IDCardSystem;Integrated Security=True";
    con.Open();
    con1.Open();
    SqlCommand cmd = new SqlCommand("Select MemberNo from
FingerprintTemplates where RecordId=' " + candidate.Id + "' ", con);
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    DataTable dt = new DataTable();
    da.Fill(dt);
    SqlCommand cmd1 = new SqlCommand("Select FirstName, LastName
from IPSUsers where ADLogonName=' " + dt.Rows[0][0].ToString() + "' ", con1);
    SqlDataAdapter da1 = new SqlDataAdapter(cmd1);
    DataTable dt1 = new DataTable();
    da1.Fill(dt1);
    SqlCommand cmd2 = new SqlCommand("Select LastName from
IPSUsers where ADLogonName=' " + dt.Rows[0][0].ToString() + "' ", con1);
    SqlDataAdapter da2 = new SqlDataAdapter(cmd2);
    DataTable dt2 = new DataTable();
    da2.Fill(dt2);
    if (dt.Rows.Count == 1)
        if (dt1.Rows.Count == 1)
            if (dt2.Rows.Count == 1)
                MessageBox.Show(String.Format("Candidate id :
{0}\nMatching Score : {1}\nMember Number : {2}\nFirst Name : {3}\nLast Name : {4}",
candidate.Id, candidate.Score, dt.Rows[0][0].ToString(),
dt1.Rows[0][0].ToString(), dt2.Rows[0][0].ToString()));

                //SqlCommand cmd3 = new SqlCommand("Insert INTO
dbo.FingerprintIdentification(MemberNo, EnrolmentID, MatchingScore VALUES(" +
dt.Rows[0][0].ToString() + "," + candidate.Id + "," + candidate.Score + ")",
con);
                //SqlDataAdapter da3 = new SqlDataAdapter(cmd3);
                //DataTable dt3 = new DataTable();

                SqlDataAdapter da3 = new SqlDataAdapter();
                da3.InsertCommand = new SqlCommand("INSERT INTO
FingerprintIdentification
(MemberNo,EnrolmentID,LiveQuality,DeviceID,MatchingScore,IdentificationDate)
VALUES(@MemberNo,@EnrolmentID,@LiveQuality,'293569866-
1607S008729',@MatchingScore,@IdentificationDate)", con);
                da3.InsertCommand.Parameters.Add("@MemberNo",
SqlDbType.VarChar).Value = dt.Rows[0][0].ToString();
                da3.InsertCommand.Parameters.Add("@EnrolmentID",
SqlDbType.VarChar).Value = candidate.Id;
                da3.InsertCommand.Parameters.Add("@LiveQuality",
SqlDbType.VarChar).Value = enrollResult.Quality;
                //da3.InsertCommand.Parameters.Add("@DeviceID",
SqlDbType.VarChar).Value = _device.Acquire(deviceSerialNumber);

```

```

                da3.InsertCommand.Parameters.Add("@MatchingScore",
SqlDbType.Int).Value = candidate.Score;
                da3.InsertCommand.Parameters.Add("@IdentificationDate",
SqlDbType.DateTime).Value = DateTime.Now;

                da3.InsertCommand.ExecuteNonQuery();

                con.Close();
                con1.Close();
            }
            else MessageBox.Show("User not found");
        }
        else
        {
            DisplayStatusMessage(acqresult.Status, "Identify");
            //MessageBox.Show("User not found");
        }
    }
    catch (Exception exc)
    {
        MessageBox.Show(exc.Message, "Verify", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
}

public void CancelAcquisition()
{
    _device.CancelAcquisition();
}

public void SetacquisitionParameters(IAcquisitionParameters parameters)
{
    if (AcquisitionDeviceEnum.SingleFingerDevice !=
parameters.AcquisitionDeviceType)
    {
        _parameters = new MsoAcquisitionParameters();
        _parameters = parameters;
    }
    else
    {
        _parameters = parameters;
    }
}

public string GetDeviceInformation(String value)
{
    IAcquisitionDeviceDescriptor descriptor = _device.GetDescriptor(value);
    StringBuilder sb = new StringBuilder();
    sb.AppendFormat("{0}\n{1}\n{2}\n", descriptor.ProductDescriptor,
descriptor.SensorDescriptor,
descriptor.SoftwareDescriptor);
    return sb.ToString();
}

public IList<String> DevicesList
{
    get
    {
        IList<string> list = new List<string>();
        int numberOfDevices = _device.GetNumberOfDevices();
        if (numberOfDevices == 0)
        {
            //MessageBox.Show("No devices were found", "Enumeration result",

```

```

        //    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        return list;
    }
IAcquisitionDeviceInfo[] deviceInfos = _device.EnumerateDevices();

foreach (IAcquisitionDeviceInfo info in deviceInfos)
{
    list.Add(info.SerialNumber);
}
return list;
}
}

public string DeviceToUse
{
    get
    {
        return _currentDevice;
    }
    set
    {
        if (String.IsNullOrEmpty(value))
            throw new ArgumentNullException("Device serial number shold not be
null or empty.");
        _currentDevice = value;
    }
}

public AcquisitionDeviceEnum AcquisitionDeviceType
{
    get { return AcquisitionDeviceEnum.SingleFingerDevice; }
}

#endregion

#region Implementation

public MsoAcquisitionStrategy()
{
    _acquisitionDeviceType = AcquisitionDeviceEnum.SingleFingerDevice;
    _device = new AcquisitionDevice();
    _parameters = new MsoAcquisitionParameters();
}

public void SetFingerEvent(FingerEventHandler ev)
{
    _device.FingerEvent += ev;
}

public void SetEnrolmentEvent(EnrolmentEventHandler ev)
{
    _device.EnrolmentEvent += ev;
}

public void SetQualityEvent(QualityEventHandler ev)
{
    _device.QualityEvent += ev;
}

public void SetImageEvent(ImageEventHandler ev)
{
    _device.ImageEvent += ev;
}

```

```

public void SetImageHandle(IntPtr handle)
{
    _device.Display = handle;
}

#endregion

private void SaveImage(byte[] data)
{
    if (_parameters.SaveImages)
    {
        SaveFileDialog fileDlg = new SaveFileDialog();
        fileDlg.AddExtension = true;

        fileDlg.FileName = "fingerprint";
        fileDlg.Title = "Save picture";
        fileDlg.DefaultExt = "raw";
        fileDlg.Filter = "RAW Image (*.raw)|*.raw|All Files (*.*)|*.*";
        if (fileDlg.ShowDialog() == DialogResult.OK)
        {
            FileStream fs = new FileStream(fileDlg.FileName, FileMode.Create);
            fs.Write(data, 0, data.Length);
            fs.Close();
        }
    }
}

private void SaveTemplate(byte[] data)
{
    if (_parameters.SaveTemplates)
    {
        SaveFileDialog fileDlg = new SaveFileDialog();
        fileDlg.AddExtension = true;

        fileDlg.FileName = "fingerprint";
        fileDlg.Title = "Save fingerprint template";
        fileDlg.DefaultExt = "cfv";
        fileDlg.Filter = "Morpho CFV Fingerprint Template (*.cfv)|*.cfv|All
Files (*.*)|*.*";
        if (fileDlg.ShowDialog() == DialogResult.OK)
        {
            FileStream fs = new FileStream(fileDlg.FileName, FileMode.Create);
            fs.Write(data, 0, data.Length);
            fs.Close();
        }
    }
}

private IAcquisitionDevice _device;
private string _currentDevice;
private AcquisitionDeviceEnum _acquisitionDeviceType;
private IAcquisitionParameters _parameters;
}

public class MsoAcquisitionParameters : IAcquisitionParameters
{
    #region IAcquisitionParameters Members

    public int Timeout
    {
        get
        {
            return _timeout;
        }
    }
}

```

```

        }
    set
    {
        _timeout = value;
    }
}

public FFDMode FakeFingerMode
{
    get
    {
        return _ffdmode;
    }
    set
    {
        _ffdmode = value;
    }
}

public Boolean SaveImages
{
    get
    {
        return _saveImages;
    }
    set
    {
        _saveImages = value;
    }
}

public Boolean SaveTemplates
{
    get
    {
        return _saveTemplates;
    }
    set
    {
        _saveTemplates = value;
    }
}

public AcquisitionDeviceEnum AcquisitionDeviceType
{
    get { return _acquisitionDeviceType; }
}

#endregion

public MsoAcquisitionParameters()
{
    _acquisitionDeviceType = AcquisitionDeviceEnum.SingleFingerDevice;
    _timeout = 60;
    _forceFingerOnTop = false;
    _consolidation = false;
    _saveImages = false;
    _saveTemplates = false;
}

public Boolean ForceFingerOnTop
{
    get { return _forceFingerOnTop; }
    set { _forceFingerOnTop = value; }
}

```

```
}

public Boolean Consolidation
{
    get { return _consolidation; }
    set { _consolidation = value; }
}

private bool _forceFingerOnTop;
private bool _consolidation;
private bool _saveTemplates;
private bool _saveImages;
private int _timeout;
private FFDMode _ffdmode;
private AcquisitionDeviceEnum _acquisitionDeviceType;
}

}
```

APPENDIX IX: EXAMS REPORT SYSTEM: CODE FOR LOGINFORM.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Data.OleDb;
using System.Data.SqlClient;

namespace LoginForm
{
    public partial class LoginForm : Form
    {
        SqlConnection con = new SqlConnection();
        public LoginForm()
        {
            SqlConnection con = new SqlConnection();
            con.ConnectionString = "Data Source=RUPA-TOSH\\SQLEXPRESS;Initial Catalog=FingerprintApplication;Integrated Security=True";

            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection("Data Source=RUPA-TOSH\\SQLEXPRESS;Initial Catalog=FingerprintApplication;Integrated Security=True");
            con.Open();

        }

        private void Login_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection();
            con.ConnectionString = "Data Source=RUPA-TOSH\\SQLEXPRESS;Initial Catalog=FingerprintApplication;Integrated Security=True";
            con.Open();
            string Username = UsernameTextBox.Text;
            string Password = PasswordTextBox.Text;
            SqlCommand cmd = new SqlCommand("select Username,Password,Role from LoginAccounts where Username='" + UsernameTextBox.Text + "and Password='" + PasswordTextBox.Text + "'", con);
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            da.Fill(dt);
            if (dt.Rows.Count == 1)
            {
                switch (dt.Rows[0]["Role"] as string)
                {
                    case "Admin":
                        {
                            this.Hide();
                            SearchForm ss = new SearchForm();
                            ss.Show();
                            break;
                        }
                }
            }
        }
}
```

```

        case "Exams":
        {
            this.Hide();
            ExamsOfficeReports mf = new ExamsOfficeReports();
            mf.Show();
            break;
        }

        case "Invigilator":
        {
            this.Hide();
            ExamInvigilator mf = new ExamInvigilator();
            mf.Show();
            break;
        }
    }

    else
    {
        MessageBox.Show("Invalid Login! Please check Username and Password");
    }
    con.Close();
}

private void button1_Click(object sender, EventArgs e)
{
    Application.Exit();
}
}
}

```

APPENDIX X: EXAMS REPORT SYSTEM – CODE FOR SEARCHFORM.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Data.OleDb;
using System.Data.SqlClient;

namespace LoginForm
{
    public partial class SearchForm : Form
    {
        public SearchForm()
        {

            InitializeComponent();
            EnrollFingerprint.Visible = false;
            //GridView1.Visible = false;
            listBox1.Visible = false;

        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            listBox1.Items.Clear();
            EnrollFingerprint.Visible = false;
            MemberPhoto.Visible = false;
        }

        private void SearchButton_Click(object sender, EventArgs e)
        {
            SqlConnection connection = new SqlConnection();
            connection.ConnectionString = "Data Source=RUPA-TOSH\SQLEXPRESS;Initial Catalog=IDCardSystem;Integrated Security=True";
            String sql = "select * from IPSUsers where ADLogonName=' " + SearchBox.Text
            + " '";
            SqlDataAdapter dataAdapter = new SqlDataAdapter(sql, connection); //c.con is the connection string
            using (SqlCommand cmd = new SqlCommand(sql, connection))

            {
                connection.Open();
                using (SqlDataReader reader = cmd.ExecuteReader())

                {
                    if (reader.HasRows)
                    {
                        while (reader.Read())
                        {
                            listBox1.Items.Add(reader["ADLogonName"].ToString() + " "
+ reader["FirstName"].ToString() + " " + reader["LastName"].ToString());
                            Image image =
Image.FromFile(@"C:\Users\Rupa\Documents\Pictures\Rupa\" + SearchBox.Text + ".jpg");
                            this.MemberPhoto.Image = image;
                            MemberPhoto.Visible = true;
                            listBox1.Visible = true;
                        }
                    }
                }
            }
        }
    }
}
```

```

        EnrollFingerprint.Visible = true;

    }
    reader.Close();
}
else
{
    MessageBox.Show("No record is found with this number: " + "" +
SearchBox.Text.ToString() + "'");
}

connection.Close();
}
}

private void EnrollFingerprint_Click(object sender, EventArgs e)
{
System.Diagnostics.Process.Start("C:/Morpho/MorphoKit/Samples/vs100/win32/Debug/Morpho
KitDotNETSample.exe");
}

private void ExitButton_Click(object sender, EventArgs e)
{
    Application.Exit();
}
}
}

```

APPENDIX XI: EXAMS REPORT SYSTEM – CODE FOR EXAMSOFFICEREPORTS.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Data.SqlClient;
using System.Data.OleDb;
using System.Data.SqlClient;
using Excel = Microsoft.Office.Interop.Excel;
using Microsoft.Office.Interop.Excel;

namespace LoginForm
{
    public partial class ExamsOfficeReports : Form
    {
        public ExamsOfficeReports()
        {
            InitializeComponent();
            comboBox1.Items.Add("Absentee Students for Exams Office");
            comboBox1.Items.Add("Extra Students for Exams Office");
            comboBox1.Items.Add("Absentee Students for Marker");
            comboBox1.Items.Add("Extra Students for Marker");
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (comboBox1.SelectedItem.ToString() == "Absentee Students for Exams Office")
            {
                var select = "SELECT HostKey AS ID_NUMBER, Description AS FULL_NAME, '" + textBox1.Text + "' AS Module_Code," + "'" + textBox2.Text + "' AS Exam_Date INTO #TempTableList FROM esdev.rdreader.V_Student WHERE Id IN(SELECT StudentId FROM esdev.rdreader.V_ExamAllocation E WHERE EXISTS (SELECT F.ExamRequirementId FROM esdev.rdreader.V_Examination F WHERE E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '" + textBox1.Text + "%' AND StartDateTime >= '" + textBox2.Text + "'))";
                SELECT ADLogonName AS ID_Number, FirstName AS First_Name, LastName AS Last_Name INTO #TempTable FROM IDCardsystem.dbo.IPSUsers WHERE ADLogonName IN(SELECT DISTINCT MemberNo FROM FingerprintApplication.dbo.FingerprintIdentification WHERE DeviceId IN(SELECT DeviceID FROM FingerprintApplication.dbo.DeviceList d INNER JOIN esdev.rdreader.V_Location l ON d.DeviceLocation=l.Description WHERE EXISTS(SELECT StudentId, ExamRoomLocationId FROM esdev.rdreader.V_ExamAllocation E WHERE EXISTS (SELECT F.ExamRequirementId FROM esdev.rdreader.V_Examination F WHERE E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '" + textBox1.Text + "%' AND StartDateTime >= '" + textBox2.Text + "')));SELECT * FROM #TempTableList WHERE ID_Number NOT IN(SELECT ID_NUMBER FROM #TempTable);";
                var c = new SqlConnection("Data Source = RUPA-TOSH\\SQLEXPRESS;
Initial Catalog = esdev; Integrated Security = True");
                var dataAdapter = new SqlDataAdapter(select, c);
                var commandBuilder = new SqlCommandBuilder(dataAdapter);
                var ds = new DataSet();
                dataAdapter.Fill(ds);
                dataGridView1.ReadOnly = true;
                dataGridView1.DataSource = ds.Tables[0];
                dataGridView1.Visible = true;
                ExportResults.Visible = true;
```

```

}

//Start of Code Set one which works
//{
//SqlConnection con;
//SqlCommand cmd;
//SqlDataAdapter da;
//DataSet ds;

//con = new SqlConnection("Data Source=RUPA-TOSH\\SQLEXPRESS;Initial Catalog=esdev;Integrated Security=True");
//cmd = new SqlCommand("Select Id from V_Location where Description='"
+ textBox1.Text + "'", con);
//con.Open();

//da = new SqlDataAdapter(cmd);
//ds = new DataSet();

//da.Fill(ds);
// Prepare a dummy string, this would appear in the dialog
//string dummyFileName = "Absentee Students Report";

//SaveFileDialog sf = new SaveFileDialog();
//sf.Filter = "txt files (*.txt)|*.txt|Microsoft Excel Files (*.xlsx)|*.xlsx|All files (*.*)|*.*";
//sf.FilterIndex = 2;
//sf.RestoreDirectory = true;
// Feed the dummy name to the save dialog
//sf.FileName = dummyFileName;

//if (sf.ShowDialog() == DialogResult.OK)
//{
// Save Folder
//string savePath = Path.GetDirectoryName(sf.FileName);
// Get Data
//ds.WriteXml(sf.FileName);

//System.Windows.Forms.MessageBox.Show("Results exported");
//}
//End of First Set of Code

//Start of Second Set of Code which also works but does not give user
option to save file where they want
//ds.WriteXml(@"C:\Users\Rupa\Documents\EngD\Thesis\LocationId.xlsx");



//string query = ("Select Id from V_Location where Description='"
+ textBox1.Text + "'");
//string connectionSql = "Data Source=RUPA-TOSH\\SQLEXPRESS;Initial Catalog=esdev;Integrated Security=True";
//StreamWriter myFile = new StreamWriter(@"C:\Users\Rupa\Documents\EngD\Thesis\LocationId.txt");

//using (SqlConnection connection = new SqlConnection(connectionSql))
//{
// SqlCommand command = new SqlCommand(query, connection);
//connection.Open();
//SqlDataReader reader = command.ExecuteReader();
//try
//{
// while (reader.Read())
//{
// myFile.WriteLine(String.Format("{0}",
// //reader["Id"])));
}

```

```

//System.Windows.Forms.MessageBox.Show("Results exported");
//}
//}
//catch (Exception exc)
//{
//    MessageBox.Show(exc.Message, "Error", MessageBoxButtons.OK,
//MessageBoxIcon.Error);
//}
//finally
//{
//    reader.Close();
//myFile.Close();
//}
//End of Second Set of Code
if (comboBox1.SelectedItem.ToString() == "Extra Students for Exams
Office")
{
    var select = "SELECT HostKey AS ID_NUMBER, Description AS FULL_NAME, ''"
+ textBox1.Text + "' AS Module_Code," + "" + textBox2.Text + "' AS Exam_Date INTO
#TempTableList FROM esdev.rdreader.V_Student WHERE Id IN(SELECT StudentId FROM
esdev.rdreader.V_ExamAllocation E WHERE EXISTS (SELECT F.ExamRequirementId FROM
esdev.rdreader.V_Examination F WHERE E.ExamRequirementId=F.ExamRequirementId AND
F.NAME LIKE '" + textBox1.Text + "%' AND StartDateTime >= '" + textBox2.Text + "'));
SELECT ADLogonName AS ID_Number, FirstName AS First_Name, LastName AS Last_Name,'" +
textBox1.Text + "' AS Module_Code," + "" + textBox2.Text + "' AS Exam_Date INTO
#TempTable FROM IDCardSystem.dbo.IPSUsers WHERE ADLogonName IN(SELECT DISTINCT
MemberNo FROM FingerprintApplication.dbo.FingerprintIdentification WHERE DeviceId
IN(SELECT DeviceID FROM FingerprintApplication.dbo.DeviceList d INNER JOIN
esdev.rdreader.V_Location l ON d.DeviceLocation=l.Description WHERE EXISTS(SELECT
StudentId, ExamRoomLocationId FROM esdev.rdreader.V_ExamAllocation E WHERE EXISTS
(SELECT F.ExamRequirementId FROM esdev.rdreader.V_Examination F WHERE
E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '" + textBox1.Text + "%' AND
StartDateTime >= '" + textBox2.Text + "')));SELECT * FROM #TempTable WHERE ID_Number
NOT IN(SELECT ID_NUMBER FROM #TempTableList);";
    var c = new SqlConnection("Data Source = RUPA-TOSH\SQLEXPRESS;
Initial Catalog = esdev; Integrated Security = True");
    var dataAdapter = new SqlDataAdapter(select, c);
    var commandBuilder = new SqlCommandBuilder(dataAdapter);
    var ds = new DataSet();
    dataAdapter.Fill(ds);
    dataGridView1.ReadOnly = true;
    dataGridView1.DataSource = ds.Tables[0];
    dataGridView1.Visible = true;
    ExportResults.Visible = true;

}
if (comboBox1.SelectedItem.ToString() == "Absentee Students for Marker")
{
    var select = "SELECT HostKey AS ID_NUMBER, Numeric1 AS
Candidate_Number,'" + textBox1.Text + "' AS Module_Code," + "" + textBox2.Text + "'"
AS Exam_Date INTO #TempTableList FROM esdev.rdreader.V_Student WHERE Id IN(SELECT
StudentId FROM esdev.rdreader.V_ExamAllocation E WHERE EXISTS (SELECT
F.ExamRequirementId FROM esdev.rdreader.V_Examination F WHERE
E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '" + textBox1.Text + "%' AND
StartDateTime >= '" + textBox2.Text + "')); SELECT ADLogonName AS ID_Number, FirstName
AS First_Name, LastName AS Last_Name INTO #TempTable FROM IDCardSystem.dbo.IPSUsers
WHERE ADLogonName IN(SELECT DISTINCT MemberNo FROM
FingerprintApplication.dbo.FingerprintIdentification WHERE DeviceId IN(SELECT DeviceID
FROM FingerprintApplication.dbo.DeviceList d INNER JOIN esdev.rdreader.V_Location l ON
d.DeviceLocation=l.Description WHERE EXISTS(SELECT StudentId, ExamRoomLocationId FROM
esdev.rdreader.V_ExamAllocation E WHERE EXISTS (SELECT F.ExamRequirementId FROM
esdev.rdreader.V_Examination F WHERE E.ExamRequirementId=F.ExamRequirementId AND
F.NAME LIKE '" + textBox1.Text + "%' AND StartDateTime >= '" + textBox2.Text + "
");
}

```

```

"')));SELECT Candidate_Number, Module_Code, Exam_Date FROM #TempTableList WHERE
ID_Number NOT IN(SELECT ID_NUMBER FROM #TempTable);";
        var c = new SqlConnection("Data Source = RUPA-TOSH\SQLEXPRESS;
Initial Catalog = esdev; Integrated Security = True");
        var dataAdapter = new SqlDataAdapter(select, c);
        var commandBuilder = new SqlCommandBuilder(dataAdapter);
        var ds = new DataSet();
        dataAdapter.Fill(ds);
        dataGridView1.ReadOnly = true;
        dataGridView1.DataSource = ds.Tables[0];
        dataGridView1.Visible = true;
        ExportResults.Visible = true;

    }
    if (comboBox1.SelectedItem.ToString() == "Extra Students for Marker")
    {
        var select = "SELECT HostKey AS ID_NUMBER, Description AS FULL_NAME,'"
+ textBox1.Text + "' AS Module_Code," + "" + textBox2.Text + "' AS Exam_Date INTO
#TempTableList FROM esdev.rdreader.V_Student WHERE Id IN(SELECT StudentId FROM
esdev.rdreader.V_ExamAllocation E WHERE EXISTS (SELECT F.ExamRequirementId FROM
esdev.rdreader.V_Examination F WHERE E.ExamRequirementId=F.ExamRequirementId AND
F.NAME LIKE '" + textBox1.Text + "%' AND StartDateTime >= '" + textBox2.Text + "'));
SELECT ADLogonName AS ID_Number, FirstName AS First_Name, LastName AS Last_Name,'" +
textBox1.Text + "' AS Module_Code," + "" + textBox2.Text + "' AS Exam_Date INTO
#TempTable FROM IDCardSystem.dbo.IPSUsers WHERE ADLogonName IN(SELECT DISTINCT
MemberNo FROM FingerprintApplication.dbo.FingerprintIdentification WHERE DeviceId
IN(SELECT DeviceID FROM FingerprintApplication.dbo.DeviceList d INNER JOIN
esdev.rdreader.V_Location l ON d.DeviceLocation=l.Description WHERE EXISTS(SELECT
StudentId, ExamRoomLocationId FROM esdev.rdreader.V_ExamAllocation E WHERE EXISTS
(SELECT F.ExamRequirementId FROM esdev.rdreader.V_Examination F WHERE
E.ExamRequirementId=F.ExamRequirementId AND F.NAME LIKE '" + textBox1.Text + "%' AND
StartDateTime >= '" + textBox2.Text + "')));SELECT DISTINCT Numeric1 AS
CANDIDATE_NUMBER,'" + textBox1.Text + "' AS Module_Code," + "" + textBox2.Text + "' 
AS Exam_Date INTO #TempTableC FROM esdev.dbo.V_Student WHERE HostKey IN(SELECT
ID_Number FROM #TempTable WHERE EXISTS(SELECT DISTINCT Numeric1, HostKey FROM
esdev.dbo.V_Student WHERE HostKey=ID_Number));SELECT * From #TempTableC;";
        var c = new SqlConnection("Data Source = RUPA-TOSH\SQLEXPRESS;
Initial Catalog = esdev; Integrated Security = True");
        var dataAdapter = new SqlDataAdapter(select, c);
        var commandBuilder = new SqlCommandBuilder(dataAdapter);
        var ds = new DataSet();
        dataAdapter.Fill(ds);
        dataGridView1.ReadOnly = true;
        dataGridView1.DataSource = ds.Tables[0];
        dataGridView1.Visible = true;
        ExportResults.Visible = true;

    }
}

private void ExportResults_Click(object sender, EventArgs e)
{
    string dummyFileName = "Exams Office Report";
    SaveFileDialog sfd = new SaveFileDialog();
    //DialogResult drSaveFile = sfd.ShowDialog();
    sfd.Filter = "Microsoft Excel Files (*.xlsx)|*.xlsx";
    sfd.FilterIndex = 2;
    sfd.RestoreDirectory = true;
    // Feed the dummy name to the save dialog
    sfd.FileName = dummyFileName;

    try
    {
        if (sfd.ShowDialog() == DialogResult.OK)

```

```
{  
    string savePath = Path.GetDirectoryName(sfd.FileName);  
    ApplicationClass ExcelApp = new ApplicationClass();  
    ExcelApp.Application.Workbooks.Add(Type.Missing);  
  
    //ExcelApp.ActiveWorkbook.FileFormat = XlFileFormat.xlsExcel8;  
    // Change properties of the Workbook  
    ExcelApp.Columns.ColumnWidth = 20;  
  
    // Storing header part in Excel  
    for (int i = 1; i < dataGridView1.Columns.Count + 1; i++)  
    {  
        ExcelApp.Cells[1, i] = dataGridView1.Columns[i -  
1].HeaderText;  
    }  
  
    // Storing Each row and column value to excel sheet  
    for (int i = 0; i < dataGridView1.Rows.Count; i++)  
    {  
        for (int j = 0; j < dataGridView1.Columns.Count; j++)  
        {  
            if (j == 2 || j == 5)  
            {  
                ExcelApp.Cells[i + 2, j + 1] = " " +  
dataGridView1.Rows[i].Cells[j].Value.ToString();  
            }  
            else  
            {  
                ExcelApp.Cells[i + 2, j + 1] =  
dataGridView1.Rows[i].Cells[j].Value.ToString();  
            }  
        }  
    }  
  
    //Dictate file Path  
    //ExcelApp.ActiveWorkbook.SaveCopyAs("C:\\\" + FileName);  
  
    //OR use SaveFileDialog  
    ExcelApp.ActiveWorkbook.SaveCopyAs(sfd.FileName);  
  
    //OR use SaveAs function  
    //ExcelApp.ActiveWorkbook.SaveAs(sfd.FileName,  
XlFileFormat.xlsExcel8, null, null, null,  
                                // null, XlSaveAsAccessMode.xlShared, null, null, null,  
null);  
    ExcelApp.ActiveWorkbook.Saved = true;  
    ExcelApp.Quit();  
    System.Windows.Forms.MessageBox.Show("Results exported");  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}  
}
```

APPENDIX XII: EXAMS REPORT SYSTEM – CODE FOR EXAMINVIGILATOR.CS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Data.OleDb;
using System.Data.SqlClient;
using System.Diagnostics;

namespace LoginForm
{
    public partial class ExamInvigilator : Form
    {
        public ExamInvigilator()
        {
            InitializeComponent();
            listBox1.Visible = false;
            SignIn.Visible = false;
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            listBox1.Items.Clear();
            MemberPhoto.Visible = false;
        }

        private void SearchButton_Click(object sender, EventArgs e)
        {
            SqlConnection connection = new SqlConnection();
            connection.ConnectionString = "Data Source=RUPA-TOSH\\SQLEXPRESS;Initial Catalog=IDCardSystem;Integrated Security=True";
            String sql = "select * from IPSUsers where ADLogonName=' " + SearchBox.Text + "'";
            SqlDataAdapter dataAdapter = new SqlDataAdapter(sql, connection); //c.con is the connection string
            using (SqlCommand cmd = new SqlCommand(sql, connection))

            {
                connection.Open();
                using (SqlDataReader reader = cmd.ExecuteReader())

                {
                    if (reader.HasRows)
                    {
                        while (reader.Read())
                        {
                            listBox1.Items.Add(reader["ADLogonName"].ToString() + " " +
+ reader["FirstName"].ToString() + " " + reader["LastName"].ToString());
                            Image image =
Image.FromFile(@"C:\Users\Rupa\Documents\Pictures\Rupa\" + SearchBox.Text + ".jpg");
                            this.MemberPhoto.Image = image;
                            MemberPhoto.Visible = true;
                            listBox1.Visible = true;
                            SignIn.Visible = true;
                        }
                    }
                    reader.Close();
                }
            }
        }
    }
}
```

```

        }
        else
        {
            MessageBox.Show("No record is found with this number: " + """
+ SearchBox.Text.ToString() + '''");
            }
        }

        connection.Close();
    }
}

private void SignIn_Click(object sender, EventArgs e)
{
    string str =
@"C:\Morpho\MorphoKit\Samples\vs100\win32\Debug\MorphoKitDotNETSample.exe";
    Process process = new Process();
    process.StartInfo.FileName = str;
    process.Start();
}
}
}

```

APPENDIX XIII: PROJECT PLAN

	Task Mo	Task Name	Duration	Start	Finish
1		Proof of Concept - Biometrics Pilot	335 days	Mon 20/03/17 08:00	Fri 29/06/18 17:00
2		Phase 1	132 days	Mon 20/03/17 08:00	Tue 19/09/17 17:00
3	✓	Initiating	20 days	Mon 20/03/17 08:00	Fri 14/04/17 17:00
4	✓	Transfer of funds	10 days	Mon 20/03/17 08:00	Fri 31/03/17 17:00
5	✓	Stock order	2 days	Mon 03/04/17 08:00	Tue 04/04/17 17:00
6	✓	Liaising with stakeholders	6 days	Mon 03/04/17 08:00	Mon 10/04/17 17:00
7	✓	Implementation	14 days	Tue 11/04/17 08:00	Fri 28/04/17 17:00
8	✓	Installation and Training	1 day	Wed 12/04/17 08:00	Wed 12/04/17 17:00
9	✓	Disseminating information to students and capturing fingerprint data	9 days	Tue 18/04/17 08:00	Fri 28/04/17 17:00
10		Testing (Identification)	14 days	Tue 02/05/17 08:00	Fri 19/05/17 17:00
11	✓	Exams processes using wall reader	14 days	Tue 02/05/17 08:00	Fri 19/05/17 17:00
12		Demo and System Feedback	1 day	Wed 24/05/17 08:00	Wed 24/05/17 17:00
13	✓	Share results with focus group	1 day	Wed 24/05/17 08:00	Wed 24/05/17 17:00
14		Decommission	1 day	Fri 26/05/17 08:00	Fri 26/05/17 17:00
15	✓	Reinstall the TDS readers	1 day	Fri 26/05/17 08:00	Fri 26/05/17 17:00
16		Conclude	82 days	Mon 29/05/17 08:00	Tue 19/09/17 17:00
17	✓	Integrate/Enhance android	12 days	Mon 29/05/17 08:00	Tue 13/06/17 17:00
18	✓	Analyse results and prepare report	6 days	Tue 13/06/17 08:00	Tue 20/06/17 17:00
19	✓	Arrange visit to Beaumont School, liaise with stakeholders to prepare for Phase 2 for Academic year	52 days	Wed 21/06/17 08:00	Thu 31/08/17 17:00
20		Test system during Phase 2	7 days	Mon 04/09/17 08:00	Tue 12/09/17 17:00
21		Phase 2	216 days	Fri 01/09/17 08:00	Fri 29/06/18 17:00
22		Fingerprint Enrollment	26 days	Fri 01/09/17 08:00	Fri 06/10/17 17:00
23		Recruit and train casual staff	6 days	Fri 01/09/17 08:00	Fri 08/09/17 17:00
24		Setup for Registration	5 days	Mon 11/09/17 08:00	Fri 15/09/17 17:00
25		Capture fingerprints for students on nominated cohort(s) of at least 200	15 days	Mon 18/09/17 08:00	Fri 06/10/17 17:00
26		Implementation and Testing	45 days	Mon 26/03/18 08:00	Fri 25/05/18 17:00
27		TDS-->Morpho reader transition	10 days	Mon 26/03/18 08:00	Fri 06/04/18 17:00
28		Exams processes using wall reader	15 days	Mon 30/04/18 08:00	Fri 18/05/18 17:00
29		Morpho-->TDS reader transition	5 days	Mon 21/05/18 08:00	Fri 25/05/18 17:00
30		Feedback, Consultation and Review	5 days	Mon 18/06/18 00:00	Fri 22/06/18 17:00
31		Share results with focus group/interested parties and make decision on live system	5 days	Mon 18/06/18 08:00	Fri 22/06/18 17:00
32		Approval	5 days	Mon 25/06/18 08:00	Fri 29/06/18 17:00
33		Submit proposal	5 days	Mon 25/06/18 08:00	Fri 29/06/18 17:00

APPENDIX XIV: CONSENT FORM FOR PROOF OF CONCEPT



Fingerprint Recognition System - Consent Form

As part of a research project, we would like to test a fingerprint recognition system for exams and attendance instead of using ID cards.

For this purpose and to test the speed and accuracy of the system, we would like to collect your fingerprint template so we can then test it by asking you to register for your exams using your fingerprint.

The related research project has received Ethics Approval. Protocol Numbers: aENT/PGR/UH/02043(2).

Key Facts:

- The image of your fingerprint will NEVER be stored.
- The fingerprint will be converted to an encrypted template consisting of numbers and letters only.
- The fingerprint template, your UH ID number and name will be stored on a secure server.
- The data will be removed after end of year: By 30th June 2018.
- You can withdraw and ask for your fingerprint template to be deleted from our system at any time after you have given your consent. Details of this can be found at the bottom of this form.

Your Personal Details

Student Number

First Name

Last Name

E-mail Address

By providing the above information and ticking this box, you are giving your consent for your fingerprint to be enrolled on our test system and then to be verified as stated above. Please tick this box to submit your approval.

Signature:

Date:

For further information or to request your fingerprint to be deleted from the system, please E-mail:
Rupa Patel, [REDACTED] If your request is to delete your fingerprint record, you will receive a confirmation email from us within 5-10 working days.

APPENDIX XV: PARTICIPANT INFORMATION SHEET

UNIVERSITY OF HERTFORDSHIRE

ETHICS COMMITTEE FOR STUDIES INVOLVING THE USE OF HUMAN PARTICIPANTS
(‘ETHICS COMMITTEE’)

FORM EC6: PARTICIPANT INFORMATION SHEET

Title of study

A Biometric Approach to Preventing False Use of IDs

Introduction

You are being invited to take part in a study. Before you decide whether to do so, it is important that you understand the research that is being done and what your involvement will include. Please take the time to read the following information carefully and discuss it with others if you wish. Do not hesitate to ask us anything that is not clear or for any further information you would like to help you make your decision. Please do take your time to decide whether or not you wish to take part. **The University’s regulations governing the conduct of studies involving human participants can be accessed via this link:**
<http://sitem.herts.ac.uk/secreg/upr/RE01.htm>

Thank you for reading this.

What is the purpose of this study?

The purpose of this study is to test the performance of our fingerprint recognition system and gather feedback from users.

Do I have to take part?

It is completely up to you whether or not you decide to take part in this study. If you do decide to take part you will be given this information sheet to keep and be asked to sign a consent form. Agreeing to join the study does not mean that you have to complete it. You are free to withdraw at any stage without giving a reason. A decision to withdraw at any time, or a decision not to take part at all, will not affect any treatment/care that you may receive (should this be relevant).

Are there any age or other restrictions that may prevent me from participating?

We would require consent of parent/guardian if you are under 18

How long will my part in the study take?

If you decide to take part in this study, you will be involved

- for 2-5 minutes for the testing our system
- for 5-10 minutes for filling in the feedback form

What will happen to me if I take part?

- Enrollment

The first thing to happen will be that you will be required to present your finger on a fingerprint scanner and our fingerprint system will register your presented finger into our database.

- Testing

Once enrolled, you will be required to present your registered finger on the scanner again to test the speed and accuracy of our fingerprint recognition system.

What are the possible disadvantages, risks or side effects of taking part?

None

What are the possible benefits of taking part?

You will contribute the development of the UH strategic plan of world-leading research.

How will my taking part in this study be kept confidential?

We will not request you to give your name or any similar personal details in the whole process. Only images of your fingerprints will be taken and kept in our system.

What will happen to the data collected within this study?

- The collected fingerprint images as well as any data generated from these images will be used for research purposes only.
- None of the collected fingerprint images as well as any data generated from these images will be distributed to a third party.
- None of the collected fingerprint images as well as any data generated from these images will be displayed publicly to the media.
- If you want, you can give us consent to allow your fingerprint images as well as the data generated from them to appear in publications and to be used in research events such as conferences, workshops or seminars.

Who has reviewed this study?

This study has been reviewed by: The University of Hertfordshire Science and Technology Ethics Committee with Delegated Authority

The UH protocol number is ENT/PGR/UH/02043

Who can I contact if I have any questions?

If you would like further information or would like to discuss any details personally, please get in touch with Rupa Patel, by email: [REDACTED]

Although we hope it is not the case, if you have any complaints or concerns about any aspect of the way you have been approached or treated during the course of this study, please write to the University's Secretary and Registrar.

Thank you very much for reading this information and giving consideration to taking part in this study.

APPENDIX XVI: CONSENT FORM FOR RESEARCH PROJECT

UNIVERSITY OF HERTFORDSHIRE

**ETHICS COMMITTEE FOR STUDIES INVOLVING THE USE OF HUMAN PARTICIPANTS
(‘ETHICS COMMITTEE’)**

FORM EC4

CONSENT FORM FOR STUDIES INVOLVING HUMAN PARTICIPANTS FOR USE WHERE THE PROPOSED PARTICIPANTS ARE MINORS, OR ARE OTHERWISE UNABLE TO GIVE INFORMED CONSENT ON THEIR OWN BEHALF

I, the undersigned [*please give your name here, in BLOCK CAPITALS*]

.....
of [*please give contact details here, sufficient to enable the investigator to get in touch with you, such as a postal or email address*]

.....
hereby freely give approval for [*please give name of participant here, in BLOCK CAPITALS*]

.....
to take part in the study entitled *A Biometric Approach to Prevent False Use of IDs*

.....
1 I confirm that I have been given a Participant Information Sheet (a copy of which is attached to this form) giving particulars of the study, including its aim(s), methods and design, the names and contact details of key people and, as appropriate, the risks and potential benefits, and any plans for follow-up studies that might involve further approaches to participants. I have been given details of his/her involvement in the study. I have been told that in the event of any significant change to the aim(s) or design of the study I will be informed, and asked to renew my consent for him/her to participate in it.

2 I have been assured that he/she may withdraw from the study, and that I may withdraw my permission for him/her to continue to be involved in the study, at any time without disadvantage to him/her or to myself, or having to give a reason.

3 I have been told how information relating to him/her (data obtained in the course of the study, and data provided by me, or by him/her, about him/herself) will be handled: how it will be kept secure, who will have access to it, and how it will or may be used.

4 I have been told that I may at some time in the future be contacted again in connection with this or another study.

5 I declare that I am an appropriate person to give consent on his/her behalf, and that I am aware of my responsibility for protecting his/her interests.

Signature of person giving consent

.....Date.....

Relationship to participant

Signature of (principal) investigator

.....Date.....

APPENDIX XVII: SYSTEM FEEDBACK FORM

FINGERPRINT RECOGNITION SYSTEM FEEDBACK FORM

Thank you for agreeing to test our Fingerprint recognition system. We would appreciate your feedback!

Speed	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
It was quick to enroll my fingerprint on the system.					
The system was prompt to verify my fingerprint once the system had stored it.					
Accuracy					
I was able to enroll my fingerprint within 3 attempts.					
The system accepted all my enrolled fingerprints.					
The system rejected my fingerprint that wasn't enrolled.					
System					
The system was easy to use.					
I feel comfortable using a fingerprint system.					
Privacy					
I wouldn't have any concerns with the University storing my fingerprint throughout my duration at the University.					
Statement					
I would prefer using fingerprint recognition system instead of buying a replacement ID card in urgent cases.					
I would be willing to give my fingerprint data if it allows me to verify my identity during an exam in the event of loss or theft of my ID card.					
I would be willing to give my fingerprint data if it allows me to verify my identity to register my attendance in the event of loss or theft of my ID card.					
I would be willing to give my fingerprint data if it can then be used to verify my identity to enter an access controlled area in place of an ID card.					
I believe using a fingerprint system would make University processes quicker.					
Additional Comments (if any):					