University of Tartu
Institute of Computer Science

Roshan Lamichhane

Mobile Biometric Challenge Recording Interface

Master's Thesis

Supervisor: Professor **Eero Vainikko**

Author: ..... ,....., June 2012

Supervisor: ..... ,....., June 2012

Approval for defence

Professor: ..... ,....., June 2012

Tartu, 2012

Table of Contents

List of Figures

Chapter 1

1.1 Introduction

In today's world, information plays vital roles in multiple facets. Information is at the core of not only businesses but also the society as a whole. Among all the technologies that make use of and transmit information, Mobile Communications is of paramount importance as it provides the foremost services powered by a worldwide network and conveys steadfast communications. Nowadays, it is commonly known as the technology leader and solutions provider over urban areas as well as in remote locations around the world. Universal mobile connections reached 6 billion at the end of 2011 (Anon., 2012.) which is comparable to 87 percent of the world population? However this is enormous increase from 5.4 billion and 4.7 billion mobile subscriptions in 2010 and 2009 respectively. (Anon., 2012.) A decade ago nobody would have predicted such a growth of mobile phone uses and the tremendous scope in the applications that it is compatible with today.

Most of the adults dislike the mobile invention because they cannot live without one; this clearly specifies how the mobile permeates the lives of people and how it has become an integral part of today's society. Since the development of smart phones, everybody has been eager to get newer and newer applications and games more than ever before. Among the many different and diverse companies researching in this field, Google has found possibly the best answer to cash in this people's interest, as they developed open and complete platform for mobile devices called Android. This is a common platform includes an OS, middleware, user-interface and applications**.** It is compatible with many cell phones. Google's Android (Developer, 2012) and Microsoft's "Windows Mobile" are the two main competitors in the domain of cell phone OS.

1.2 Motivation

The fastest communication was slogan of the past. Nowadays due to the increased concern for security and the versatility in the field of mobile communications, the slogan has changed to the safest communication. Even in the recent past, the concept that biometric recognition can be applied in phones was considered much farfetched. But the growth of the smart phone numbers demonstrates this statement right. Last year BullGuard (Anon., 2006) recognized an alarming number of 2,500 different types of mobile malware that have intruded on the scene. They affect the normal functionalities – applications like web browsers do not consent to see apps and web pages the way they were designed to be seen. That makes the phone more unsafe as IBM X-Force (IBM, 2010) predicted that "targeting vulnerabilities that affect mobile operating systems will more than double from 2010" (IBM, 2010). So now most of the smart phones have security software installed in the cell phone itself which is a crucial step to increase the overall security in mobile communications. However, the emergence of ubiquitous environments and thus the development of pervasive security aware applications will be the main feature on smart phones which can make the difference in the future. So the applications which enable user interaction by combining a variety of security measures i.e. sensors, Biometry apps for safer user interaction will be an example for a successful future of the cell phone. Nowadays more people in Asia are connected to the internet via their mobile phone than there are PCs with online connections. To capitalize on this huge market, companies like Biometry are focusing on exceptional secure mobile application development to get an enormous potential market around the globe.

1.3 General problem statement

The Mobile industry has revolutionized our world. The use of smart phones in IT infrastructures in day to day business as well as in communication, transportation, banking, and nearly every sector shows our increasing dependency on the mobile communication systems. With all these increasing dependencies the mobile network connected with enormous apps can be likened to the whole world under a single roof. However, much of the data are highly important, private and extremely valuable. The safety of this data is of high concern to anyone who owns the data as there is a growing incidence of cyber-attacks intended to steal the data. This has resulted in launch of a variety of security tools such as Biometry identities, antivirus software firewalls, etc., which help to detect the threats and counter measure these wherever possible. Many studies have been done to discover the origins of these attacks. It is stated that in 2011 approximately half of respondents experienced at least one security incident last year, fully 45.6% of them reported they'd been the subjects of at least one targeted attack, so there is growing security and identity theft concerns that need to be addressed to. So, if the same rate of incidents continues, the resulting losses can be devastating.

1.4 Goal of the Thesis

The thesis focuses on developing a User Interface that incorporates Biometry, User Interface solutions and improves a secure User Interface to carry out analysis of Biometry IDs. The design and implementation of UI is based on Android platform and there will be a fusion Biometry interface for possible sets of voice, face or trained data in the given system specification.

The User Interface is raised to a higher level, the most important factor being to raise the quality of the samples so that the authentication phases are fool proof.

My thesis core will be to study different biometric authentication techniques, derive requirements from these for a mobile interface, and implement and refine an interface prototype.

1.5 Structures

The thesis is structured as below:

Chapter 2: gives the background theory on which the development of our tool is based on. In this chapter we will see theory about abstract system, system components, modeling languages and demonstrating tools and analysis.

Chapter 3 covers the details of analysis and designs patterns of UI .In this chapter we will see the steps and considerations taken for the implementation of the gizmo. We will discuss the principle on which we based our UI and also present different analysis that our system can do. Similarly, we will exhibit how our system is designed to incorporate the future extensions and enhancements of the User Interface.

Chapter 4 provides some detail on the implementation of the theory discussed in earlier chapters. In this chapter we will discuss the Biometry User Interface we developed which can be used for authentication and recognition in the system model. The chapter includes the detailed functionalities of the User Interface and explains the different external libraries used in our system.

In Chapter 5 we will evaluate the functionalities of our Biometry UI .We will set some basic tasks to be performed by the system and check whether the tool outputs are expected.

Chapter 6 provides conclusion of the work. Also, functionalities that we have thought of but haven't implemented yet due to time limitations are briefly described here.

Chapter 2

2.1 Background

This chapter presents some background information required to understand the works done in this thesis project. It introduces the protocols and technologies that are related to Android mobile domain and it's User Interface. It starts by introducing the concept of developing applications on the Android platform. Then, it presents protocols related to Android (Developer, 2012) OS, UI, and framework. Finally it discusses important features integrated in Android.

2.2 Android

Android is Linux-based software for mobile devices that includes an operating system, applications and middleware (Fergytech, 2010). It is unique because Google is actively developing the platform but giving it away for free to hardware manufacturers and phone carriers who want to use Android on their devices. The Android SDK (Developer, 2012)provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

The fig 1.0 shows an Android is open software stack for mobile devices, which includes an operating System, a middleware and some key application. (Developer, 2012)

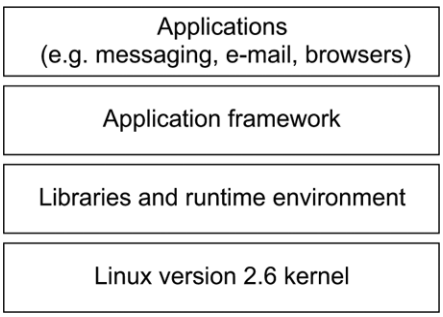


Figure 1 Android software stack layers (Emeraldinsight, 2010)

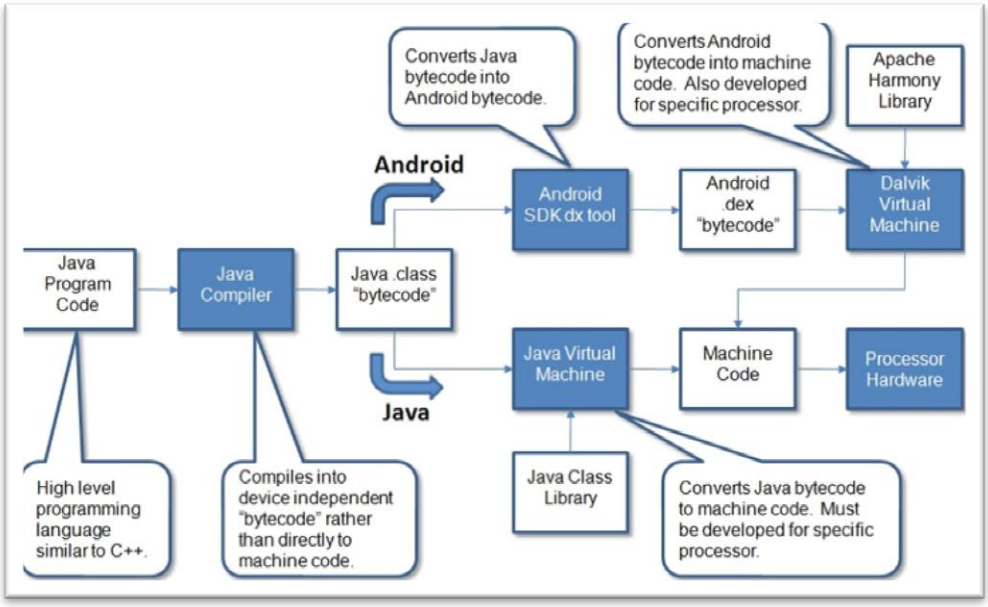


Figure 2 The Android and Java flow chart integrated with hardware device **(Anon., n.d.)**

The Figure 1.1 is intended to show how the various pieces of Java and Android can work together to produce a useable program on a given hardware device. Two paths are shown, a path through the Android OS (Android, 2011) and a more conventional pure Java path. Both produce machine code that can run on a given processor, but take different routes to get there. (Hibben, 2010)

2.3 Android Concepts

Android activity lifecycle (android-apps, 2012) and various event handlers associated with each stage are the key concepts to explore how an application behaves in Android. Therefore as developers, we should have a thorough knowledge on Android activity life cycle before start developing. An Activity in Android has essentially four states, as follows: (android-apps, 2012)

2.3.1 Running state

The activity is in the foreground of the screen. It is completely visible and active to the user. (android-apps, 2012)

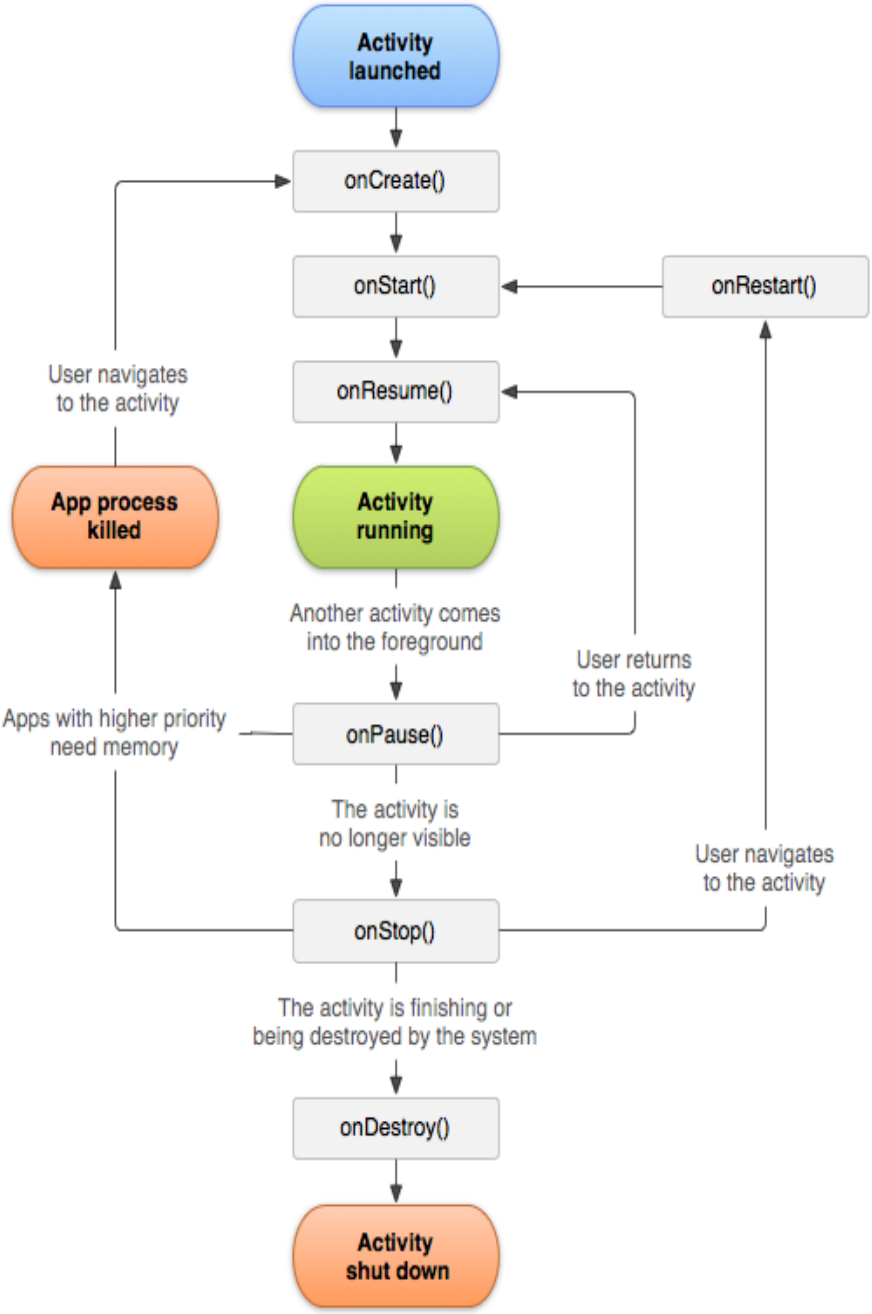


Figure 3 Android Activity Lifecycle flow diagram **(developer, 2012)**

2.3.2 Paused state

In paused state, the activity has lost focus but is still visible. This occurs when some another Activity is on top of this one which doesn't cover the entire screen or having some transparency so that the underlying Activity is partially visible. A paused activity is completely alive, meaning that it can completely maintain state and member information and remains attached to the window manager that controls the windows in Android.

2.3.3 Stopped state (Developer, 2012)

This is when the Activity is no longer visible to the user. If an activity becomes completely obscured by another activity, it is stopped. In this state also the activity is alive and preserves its state, but more likely to be killed by the Android system when memory is needed elsewhere.

2.3.4 Destroyed/Dead state

An Activity is said to be dead or destroyed when it no longer exists in the memory .When it displays that activity to the user, it must resume by restarting and restoring to its previous state. (android-apps, 2012)

2.4 The State Diagram

The figure 3 shows the important state paths of an Activity Lifecycle (android-apps, 2012) flow chart. Each of the rectangular boxes represents methods that are called on your application during the state flow. Likewise, each of the ovals boxes represent major states the Activity can be in. (Developer, 2012)

2.5 Android Lifecycle stages

By analysing the Android Activity lifecycle diagram we can see there are three lifecycle loops exist for every Activity and are defined by those call-back methods. (Developer, 2012)

The Three Lives of Android (developer, 2012)

2.5.1The Entire Lifetime

This is the period between the first call to onCreate() method and the final call to onDestroy () method. This is the time to create global resources as screen layout, global variable etc. for the app in onCreate() method and the release of all resources associated with the app in onDestroy ( ) method.

2.5.2The Visible Lifetime

This is the period between a call to onStart() method until a corresponding call to onStop() method. In this the activity is visible to the user and maintains its state intact.

2.5.3The Foreground Lifetime

The period starts with onResume() method until a corresponding call to onPause() method. During this lifetime the activity is completely visible to the user and it is on the top of all other activities and interacting with the user.

2.6 Android Platform

The development of software is tied to the use of the Delvik Virtual Machine that enables the use of Java as programming language as well as C/C++ libraries.

The application framework consists of predetermined services for managing devices resources as hardware, software and the integration with external sources. Within the framework of java and Android SDK platform the project incorporates set of libraries as OpenG, Media framework etc., that provide real time performance in the field of biometric authentication.

Linux kernel provides an adequate driver for security models, memory management, and networking (sites, 2008).A hardware abstraction layer (HAL) (sites, 2008) is implemented in software, between the physical hardware of a computer and the software that runs on that computer. Its function is to hide differences in hardware from most of the operating system kernel, so that most of the kernel-mode code does not need to be changed to run on systems with different hardware.

There are four major types of component classes in any Android application: (Reed, n.d.)

2.6.1 Activities

Much like a Form for a web page, activities display a user interface for the purpose of performing a single task. An example of an Activity class would be one which displays a Login Screen to the user.

2.6.2 Services

These differ from Activities in that they have no user interface. Services run in the background to perform some sort of task. An example of a Service class would be one which fetches your email from a web server.

2.6.3 Broadcast Receivers

The sole purpose of components of this type is to receive and react to broadcast announcements which are either initiated by system code or other applications. If you've ever

done any work with Java Swing, you can think of these like Event Handlers. For example, a broadcast announcement may be made to signal that a Wi-Fi connection has been established. A Broadcast Receiver for an email application listening for that broadcast may then trigger a Service to fetch your email.

2.6.4 Content Providers

Components of this type function to provide data from their application to other applications. Components of this type would allow an email application to use the phone's existing contact list application for looking up and retrieving an email address.

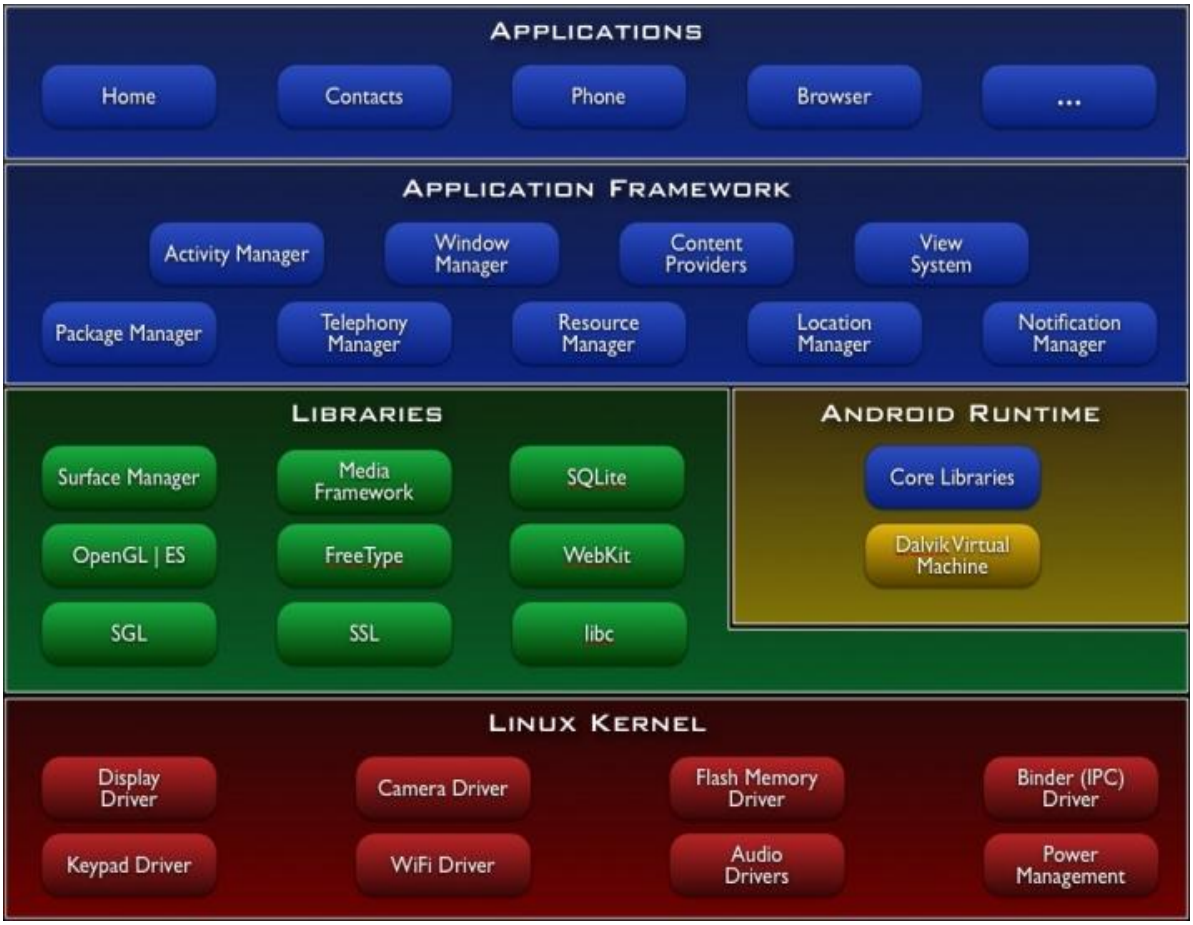


Figure 4 presents the major components of the Google Android operating system architecture **(android, 2012)**

.

Chapter 3

3.1 Problem Statements

In this chapter, we will cover analysis and design patterns to deal with the face and voice detection problem. The aim of this chapter is to provide early developed Biometric framework based on the theory and Implementation of Face and voice recognition and authentication technology.

3.2 Early developed Facial recognition Technology

In Biometry, Facial matching algorithm compares the digital photographs of the face stored in a database. It authenticates the identity of individuals by comparing the stored digital images in the system with a live photo of the individual upon an access attempt in a process called "matching" (Technologies, 2011). There are various incidences of false rejections and acceptances of matching algorithms .These false incidences of matching algorithms can be controlled if the quality of an image is improved. So, Biometry is looking for new standards in User Interface for quality of facial images.

3.3 How is facial recognition technology being used?

There were many different biometric systems being used. The facial recognition was one of the important systems that were used for general investigation, usually in combination with different types of camera such as web camera and public video cameras. In early years, US implemented face recognition technology by using public video camera in three different airports. The Airports that have announced adoption of the technology include Logan Airport in Boston, T.F. Green Airport in Providence, R.I., and San Francisco International Airport and the Fresno Airport in California (Technologies, 2011)

In 2001, Super Bowl in Tampa (Technologies, 2011) used similar technology, where pictures were taken of every person as they entered the stadium through the gates and compared with database of some secret kind.

Likewise, with the hopes of spotting criminals Tampa, Ybor City (Technologies, 2011)has also deployed trained camera on busy public streets. But they were not successful in catching any criminal and it is still unclear what criteria were being used for including photos in the database system.

3.4 Facial Recognition: How it Works

Every face has many distinguishable landmarks including the distance between the eyes, depth of the eye sockets, the areas surrounding the cheekbones, the length of the jaw line and the location of the nose and eyes - to perform verification and identification.

Earlier, 2D images were compared with the 2D (Technologies, 2011) images of database for facial recognition. To be effective and accurate, the image captured needed to be of a face that was looking almost directly at the camera. Even the smallest changes in light or orientation could reduce the effectiveness of the system, so the result was a high rate of failure owing to these circumstances.

3.5 Image Quality

The quality of image is one of the most important factors to be considered in biometric verification. The verification process itself can be marred severely by low quality images that are used as the database in the process, as there are many false instances of rejection and verification. However, different security systems need to deploy different image quality, e.g. in mass surveillance, it would not be viable to scan every face in high resolution as this tends to increase the time for the process and delay the effectiveness required for real time security provisioning. In facial recognition systems processing from far away, there is a substantial trade-off between camera quality and system capabilities. But in private systems like the one this thesis entails, the quality of the image is of paramount importance.

3.6 Initial steps about the facial expression recognition

In his doctor thesis Philippe C.Cattin (Cattin, 2002) discusses the facial expression recognition (Cattin, 2002). It was developed with a wide variety of image processing techniques to meet the system requirements. But, there were still many challenges and problems to solve in such systems, especially in the area of their performance and applicability improvement.

In his future work, he would like to focus on time efficiency (Subesi, 2011) of the system as well as geometrical features (Cattin, 2002) and appearance features for improving the recognition rate of the system.

3.7 Initial steps about the Eigen face detection on Android

In the research paper of Emir Kremić and Abdulhamit Subaşi (Subesi, 2011) discussed about the Implementation of Face Security for Authentication on Mobile Phone. The evaluation presented in the paper focused on Face detection and its authentication. They presented the model which has been implemented called Eigen face. The tests were conducted using MATLAB and droid Emulator for the Android mobile phones. These were based on face recognition as a biometric approach for authentication on mobile phones. Due to the vulnerability of using the PIN, they presented the approach which enabled a new level of mobile phone user's security.

They tested their solution with the database which consists of many images of facial expression (Biometry.com, 2010). Limited hardware capabilities compelled the substitution between accuracy and computation complexity on the application. Proliferation of application and data in diverse fields has led to the increasing need of the user to protect the data which exist in mobile devices.

They also made several suggestions to improve the authentication process by adding the level of intelligence as video and speech recognition in mobile phone security as part of mobile authentication.

3.8 A face recognition system based on local feature analysis

In their paper Stefano Arca, Paola Campadelli, Ra aella Lazaretto (Subesi, 2011) discussed a completely automatic face recognition system. They implemented the system inspired by the elastic bunch graph method, but the dual point localization is completely different (Subesi, 2011) and does not require any operator intervention. Each dual point is characterized with applying a bank of alters which extracts the peculiar texture around it (Stefano Arca, n.d.). The performances of the steerable Gaussian derivatives alters are compared to the ones of the Gabor wavelet transform, showing similar results when images of faces in approximately the same pose are compared.

3.9 Opencv for Face Detection

Opencv (d'Aix-en-Provence, n.d.) is an open source library which is based on BSD-License for the research and commercial use. It is originally developed by Intel (d'Aix-en-Provence, n.d.). The cross platform library mainly focuses on the real time image processing. Nowadays, Opencv (d'Aix-en-Provence, n.d.) is moving forward to Android platform and its libraries supports as real-time image capture import the video file and face detection technology. The API is C and C++ based and the library that includes several hundreds of computer visions algorithm.

3.10 Biometry steps for authentication

Biometry (Biometry.com, 2010) is a Swiss hi-tech company that provides and distributes software for biometric authentication procedure. Their innovative solutions and services protect and secure personal identities and assets. The ComBiom® and BiTCo® (Biometry.com, 2010), are the trusted provider for biometric authentication technology. They have varieties of secure software for biometric authentication such as face recognition, lip movement, voice, word recognition with SSO (Single Sign On), OTP (One Time Password), or RCR (Random Challenge Response) (Biometry.com, 2010). Their main aim is to provide maximum security with low cost. However their software as very expensive to develop because of dependency on other companies software and hardware products.

So biometric is moving forward to open platform and more secure authentication techniques .Recently Biometry has started working on the Android platform and they have developed voice recognition integrated with randomly generated 4 digit number system password challenge response as well as face and lip movement authentication system (Biometry.com, 2010) . Regarding the choice of Biometry authentication, we decided to work on the Android platform. It is driven with very rapid growth and is forecasted to be the number one mobile in near feature .The main motivation for choosing the Android client for the research was because it is open software with a complete stack and it is integrated with java platform and Android SDK .

Authentication is a very important aspect in the computer world .This thesis is implementing the face detection and verification using the Android device as client .Beside that major changes are made to the existing Biometry User Interface with the integration of face detection apps.

There is a variety of solutions widely discussed about the authentication such as finger print verification, voice matching and face- detection. It is difficult to choose one single solution but the research has shown that face recognition is the most consistent way to overcome the problems of Biometry verification**.**

In a face detection mechanism there are basically three steps: sampling, matching and validating. A radically different design is needed to overcome the limitations of the current

technology. Recently many researchers have made significant progress in overcoming the problems of authentication and a lot of work is being done in a variety of levels and fronts by scholars.

The flow process in face recognition consists of four phases. Capture of sample, Feature extraction, template comparison and matching.

3.11 Biometric Steps for Replacing Traditional authentication Methods

Biometric Authentication currently gains momentum by replacing and amending traditional and often complicated token and password based authentication methods. It offers the perspective to simplify and secure authentication significantly. Biometry holds the patents to ComBiom – Communication Biometrics and BiTCo – Biometric Transaction Confirmation (Biometry, 2011) which resembles a revolutionary multi-modal and simultaneous biometric authentication technology. Combiom® combines Face Recognition, Voice, Lip Movement and Speech Recognition simultaneously with SSO (Single Sign On), OTP (One Time Password), or RCR (Random Challenge Response) (Biometry, 2011). They are currently advancing algorithms and adapting these algorithms to the cloud infrastructure.

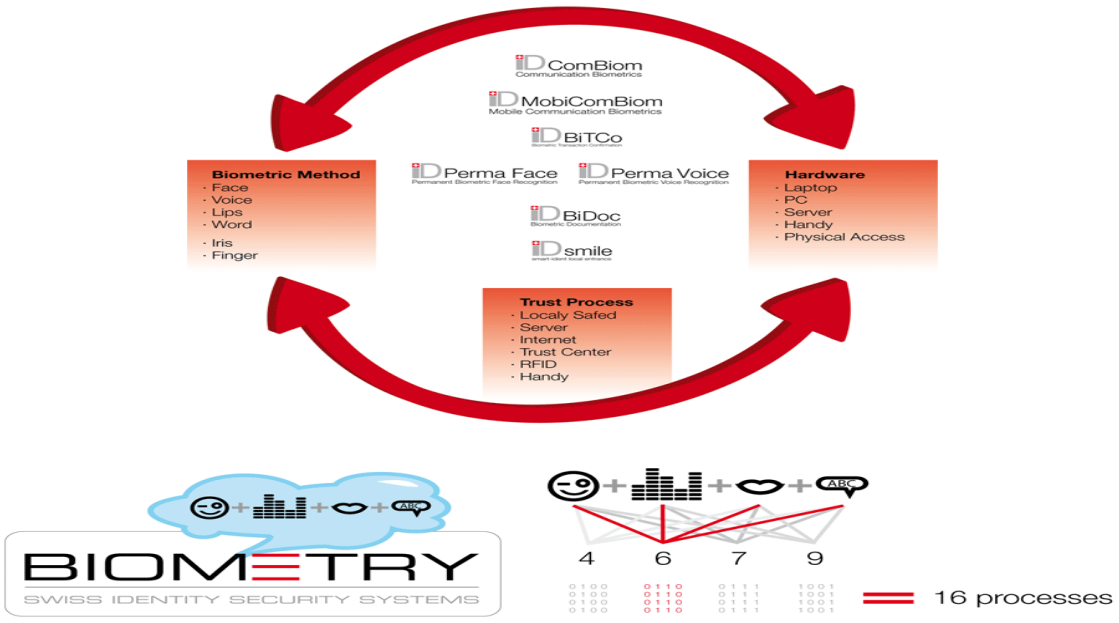


Figure 5 One Software - many applications with adaptive security **(Biometry.com, 2010)**

3.12 Summary

In this chapter, we presented discussion on how Biometry has proceeded to develop an authentication system model. We discussed briefly about the earlier implementations of face detection technology and authentication performance and an analysis of the real world system.

We also provided details on the steps taken on the basic design principals and development of the Biometry evolution and their problem statements.

Chapter 4

4.1 Mobile Biometric Challenge Recording Interface

We discussed several issues in the third chapter regarding the implementation of face detection technology in different sectors and the problems and challenges of Biometry face detection. Furthermore, we also discussed about the problem statements of other related technologies and their limitations.

In this chapter we are going to discuss about the steps of Biometry to overcome the limitations of authentication and verification process in the Android mobile platform.

4.2 Implementation

In this chapter we discuss implementation details of the project. The development of the mobile Biometric challenge recording interface is based upon the Android platform.

Furthermore, we are developing mobile front-ends user interface for Biometry.com based on Android. This research paper is concerned about “Mobile Biometric Challenge Recording Interface".

4.3 Capture of sample with face detector Android class

Face Detector is an useful Android class for detecting faces in a bitmap object .This class identifies face objects in the given bitmap with an array of **Face** objects which contains information about each detected face such as the distance between the eyes, mid-point between eyes and the position of face with respect to the rotation of x,y and z axis.
At first the FaceDetector class initializes by passing parameter i.e., the height and width of the bitmap and maximum number of faces to detect. FaceDetector object is created to use the findfaces method for face detection. After the findface method accepts a bitmap and array of face objects, we can use the findfaces method to start face detection.

4.4 Android Speech Recognition

Basically we have implemented Android voice recognition (google, 2011) engine, this is for the random challenge in the application. We implemented Listview which holds one digit random number challenge in the camera preview overly. The ACTION_RECOGNIZE_SPEECH (android, 2012) handle intends to check any package installed with the help of manager queries. After the speech is recognised the speech will be translated and then the camera proceeds to detect the face.

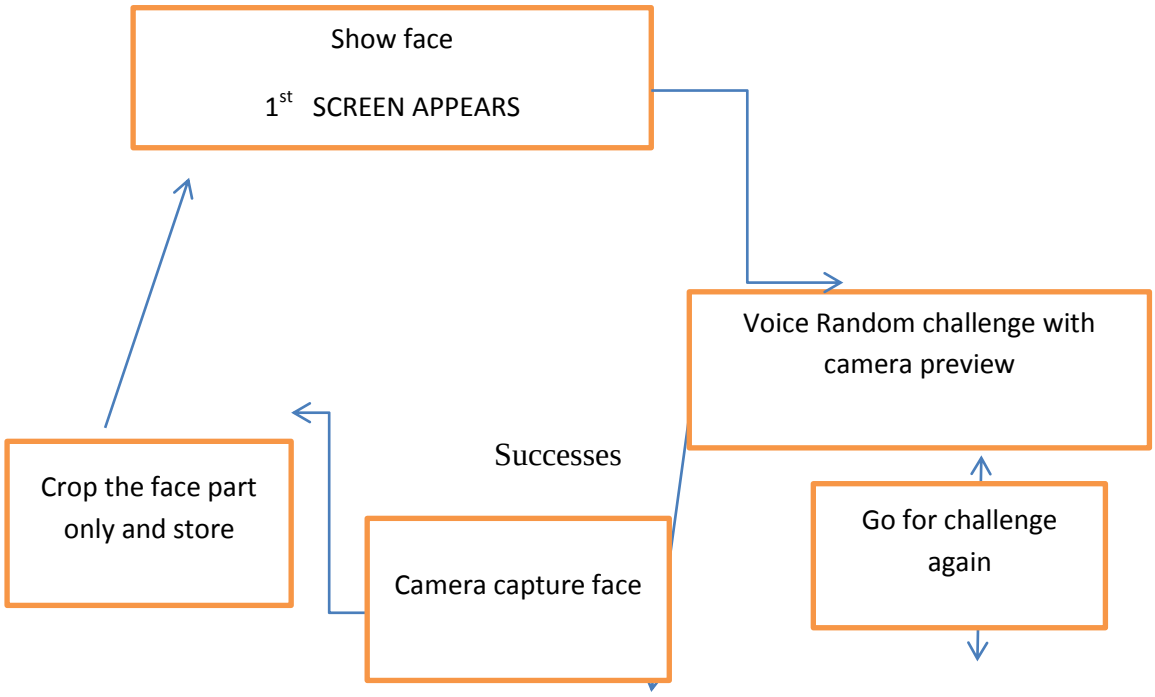


Figure 6 User Interface of ShowFace

ShowFace is an Android application that is used to determine a user's voice random challenge using an Android speech recognition and Android face detection. At first application starts with the main menu where it shows the random challenge with camera preview .The Android speech recognition is implemented for the random voice challenge. Likewise the application runs and shows the two results. If no face has been detected over a certain overlay dotted circle, it assumes the user has no clear face or the face is not real. However, there is refresh button if the user wants another challenge. If the user maintains the requirements of the apps it detects the face and crops it.

4.5 The screen-shots of the ShowFace Application flows

The research is based upon the Android SDK - which makes it easy to implement speech recognition as well as the face detection. The Android SDK is a decent solution to enhance Android application with great successes.

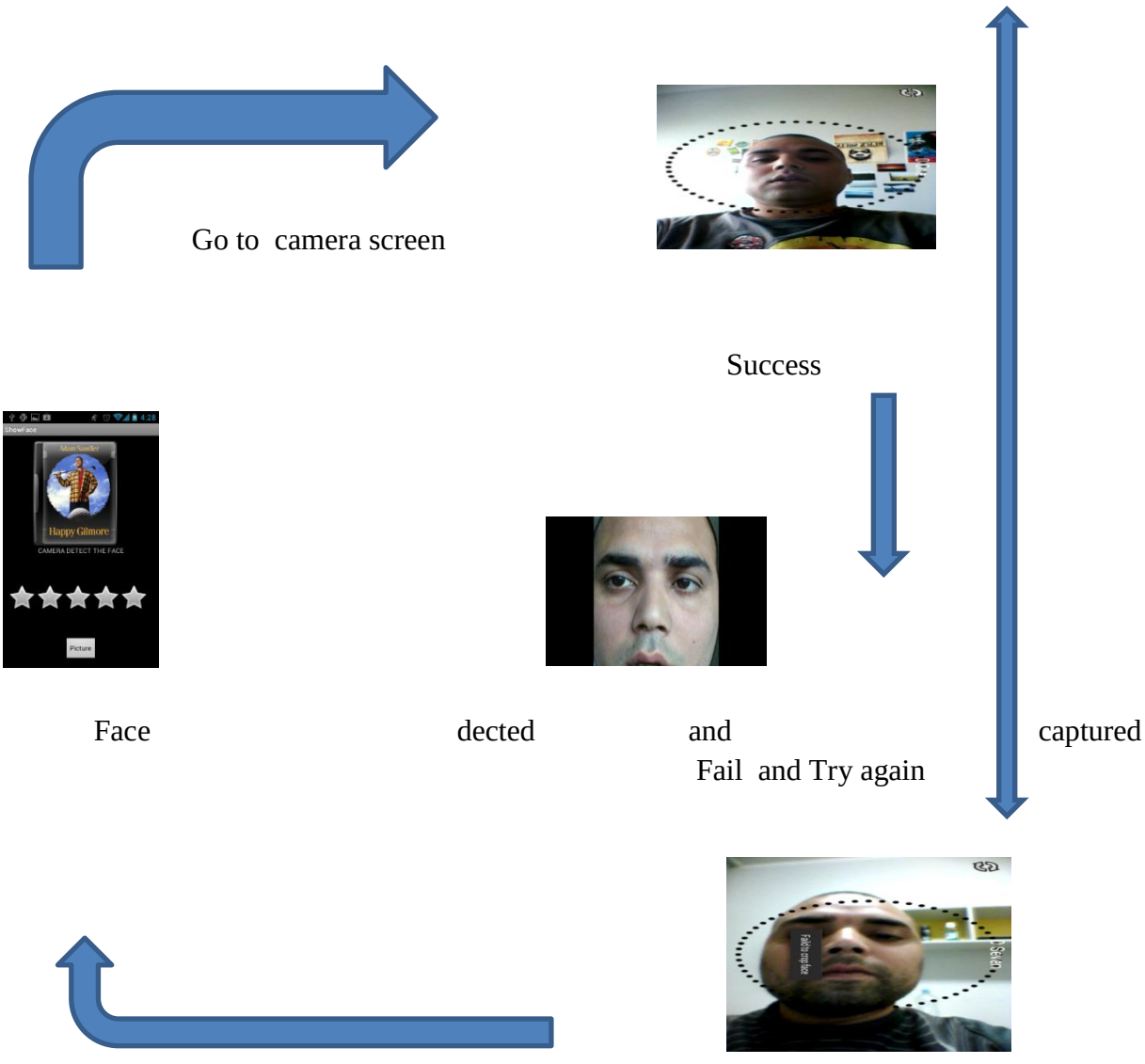


Figure 7 User Interface of the ShowFace

4.6 User Interface and Authentication process of the Biometry

Biometric Authentication currently gains momentum by replacing and amending traditional and often complicated token and password based authentication methods. It offers the perspective to simplify and secure authentication significantly.

When a biometric sample or trait is acquired, such as a video or a voice sample, the captured data goes through some amount of processing to prepare it for the extraction of a relatively small set of numbers, which represent the most unique aspects of the data. This extracted set of data in record is called a recording template. However, to prepare this recording optimally for biometric algorithms several measures already have to be taken on the phone.

For example it has to be checked, if the users really speak, really look at the phone, look straight at the phone, have not closed their eyes, and do not smile too much. Also the background noise has to be monitored and filtered out. The respective data then has to be pre-processed and compressed before it enters the actual biometric authentication mechanism either locally or remotely.

There are two basic usage scenarios for biometrics, known as verification and identification.

Verification is the scenario where a newly created template is compared with only one other template in a database, which is described as a one-to-one comparison. The second biometric usage scenario is called identification, which is used when it is necessary to compare a newly created template to many enrolled templates, which is described as a one-to-many comparison. Both scenarios have to be supported by such a recording interface.

Face captured

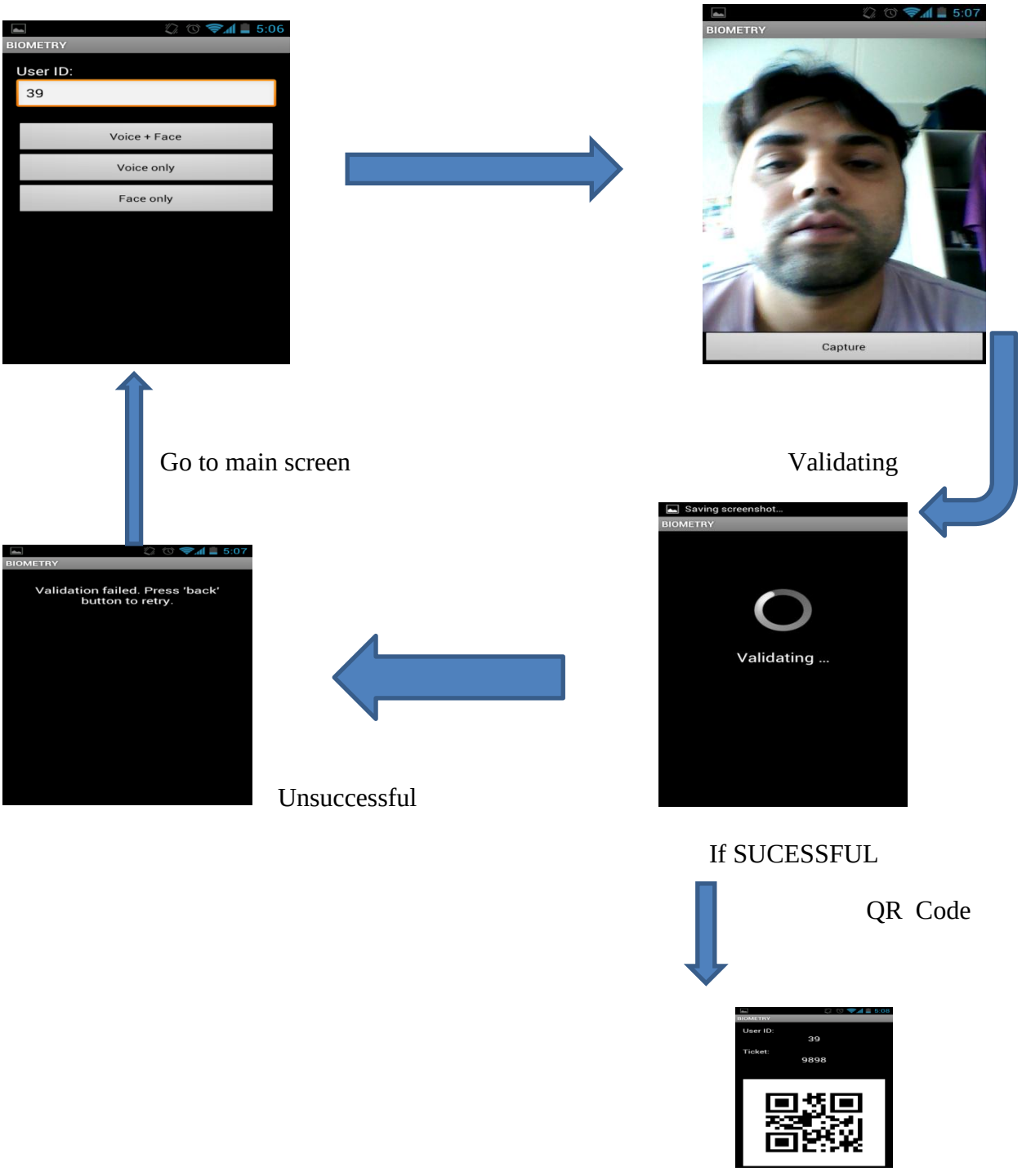


Figure 8 User Inter face and authentication process of the Biometry **(Biometry.com, 2010)**

4.7 UML activity diagram for enrolling application by the user

Lifecycle of ShowFace activity has four states, when the application is run by the user then Android speech recognition responds to the speaker speech which is overly on the camera preview. If the attempt is not successful, the user can try again with new random challenge. If the challenge is a success then camera starts with the face detection class. If the user manages the face to fit into the overly dotted circle then it detects the face otherwise it gives the message "failed to detect face".

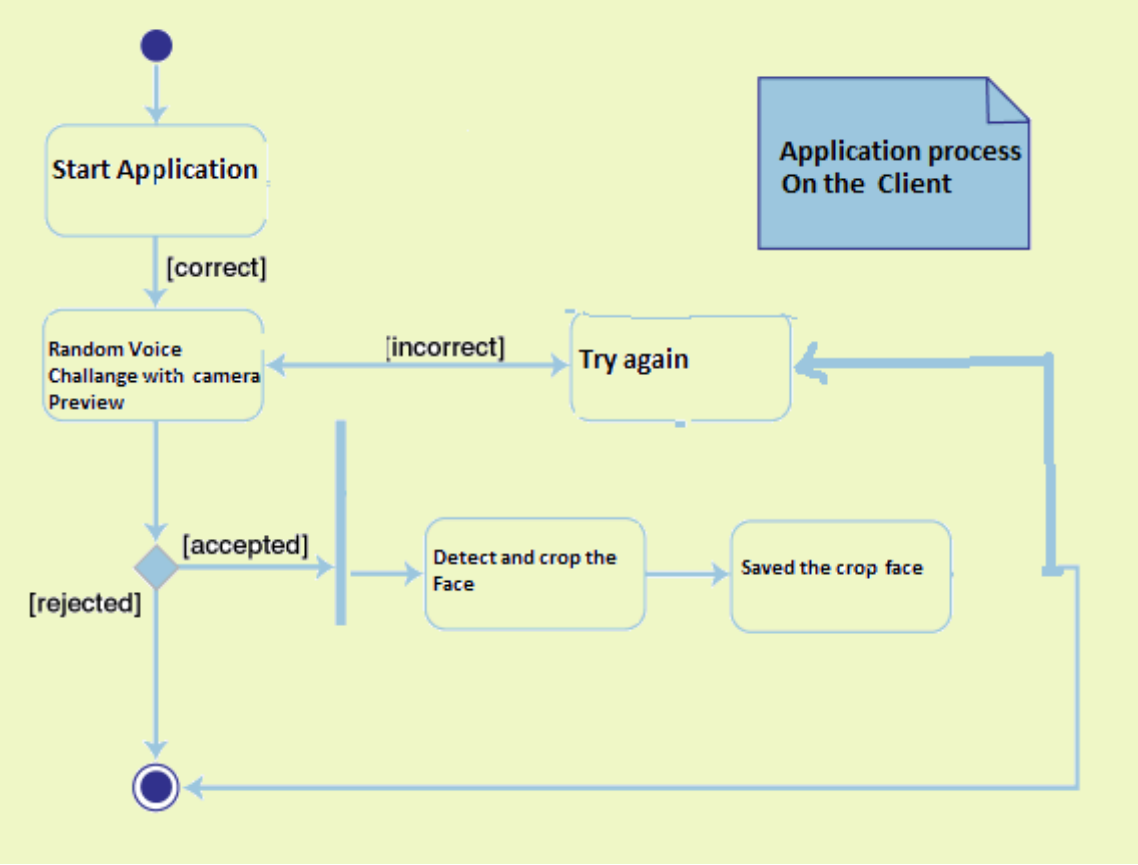
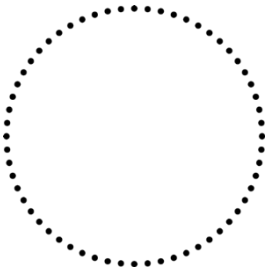


Figure 9 A UML activity diagram for enrolling application by the user

4.8 Some Images Used in the ShowFace



Circlule.png

Relaod.png

icon.png

Figure 10 Some Images Used in the ShowFace

4.9 Summary

In this chapter, we presented the implementation details of our ShowFace application. In doing so we presented details of applications workings, user interface, snap shots and Biometry working user interface snapshots. We also provided insight of xml files and some overview of important methods and class of Android SDK which are important components of this application.

Chapter 5

5.1 Case Studies

In the previous chapter, we analysed the developed process, and discussed the Biometry system functions. Here, we discuss about the issues of different implementation methods and work-flows. This chapter also illustrates the detailed evaluation of Showface application.

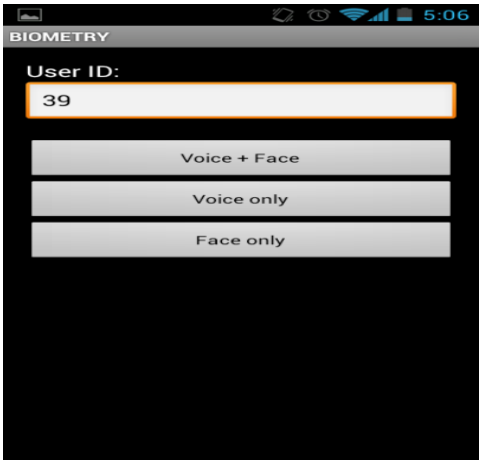


Figure 11 Biometry application: Choose activities

5.2 Application Issues:

The first version of Biometry application is based on the Android platform, which has three menus. The first one is voice and face. The second and third menus are the voice only and face only respectively. Software programmers of this application have developed this voice authentication system with 4 randomly generated digits. The voice authentication process validates the user's voice by checking the uttered random numbers which are displayed on the screen.

Similarly, there is a face detection option also present in this application. This system verifies the user by taking the picture of his face and sending it to the server. After sometime, it gives the QR code as a feedback to the client. If the face is not captured, then the validation process fails, and the application retries from the starting stage.

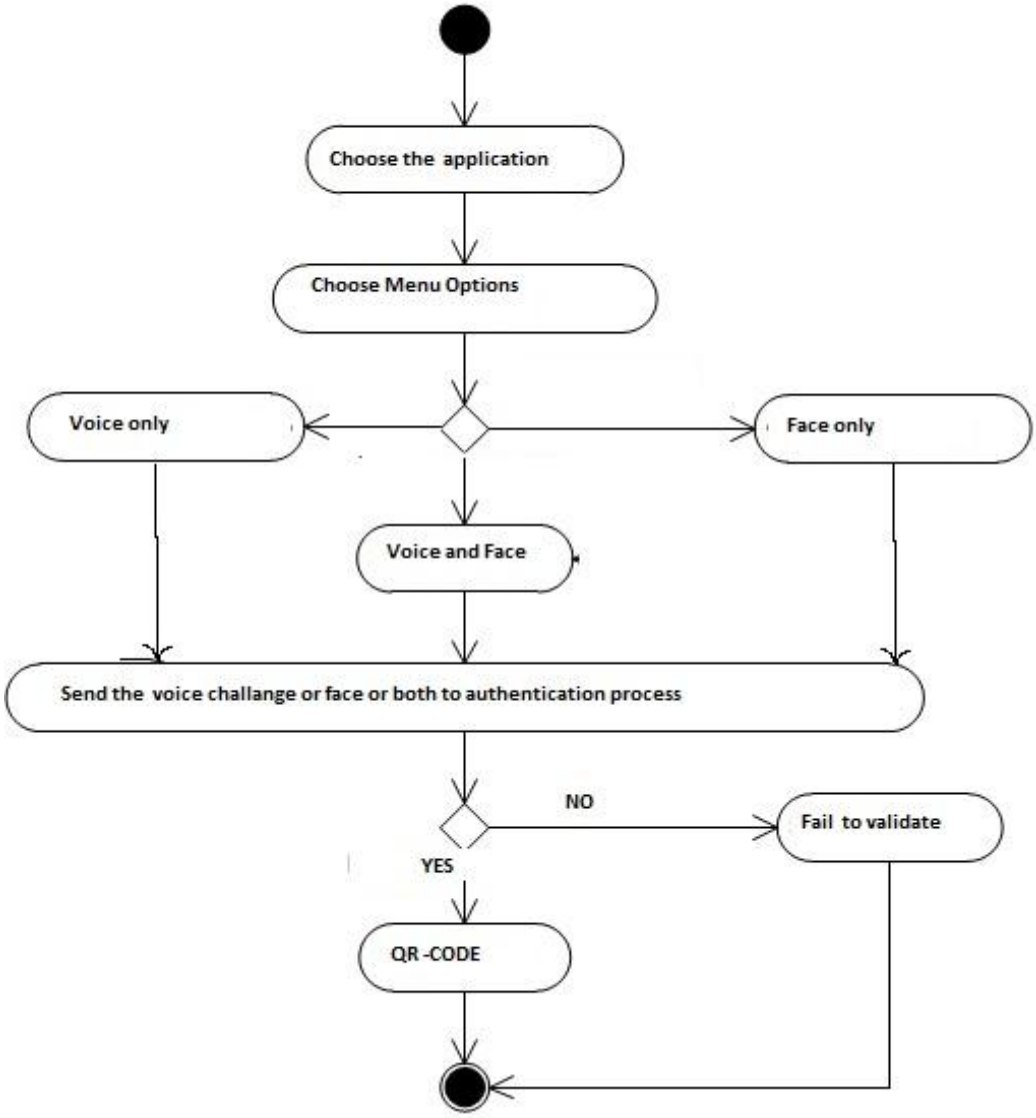


Figure 12 UML process of the Biometry

However, this research is based on enhancing the authentication steps. So the main important part of authentication phase is to train a face in a proper way**.** This Biometry application only used the simple camera and it doesn't have the specific face detection algorithm so on the client side it only captures the picture by using camera**.** To overcome this issue, the goal of this thesis is to make the Biometry application take accurate sample of face using face detection Android algorithm.

This thesis will integrate the face detection with random challenge for the accuracy and speed in the performance. For example ,if the user can train the data in more accurate way it controls the different unrelated action from the user end. The server will need to

validate the specific detected face which is cropped and thus smaller. Moreover, it controls the irrelevant actions as auto generated spams, malwares and viruses.

There are two main steps to take sample data from the client. The random challenge is used by the one single number implemented from the Android speaker recognition which will only take face detection action after the speaker speaks and successfully completes the challenge. The camera then captures the face if the face is around the dotted circle .If the face is not clear the detection will fail and the user can again try for the next challenge. So these settings and options attached into the work will assist the Biometry to overcome the vulnerability from the intruder and server load and ensures a fast validation process. The figure 3.1 is the snapshot of Biometry face capture.

Figure 13 Biometry face captured

The camera can take any picture and can send the image to the server. The Biometry server will first validate the picture and test whether the captured image is a real face or before it will start the actual matching process. There are risks on the client side that auto captured random image can pass frequently from the user end so the server will be overloaded and will finally crash.

Furthermore, we evaluated how the ShowFace application has speedup data rate and validation process on the Biometry server for the quick authentication.

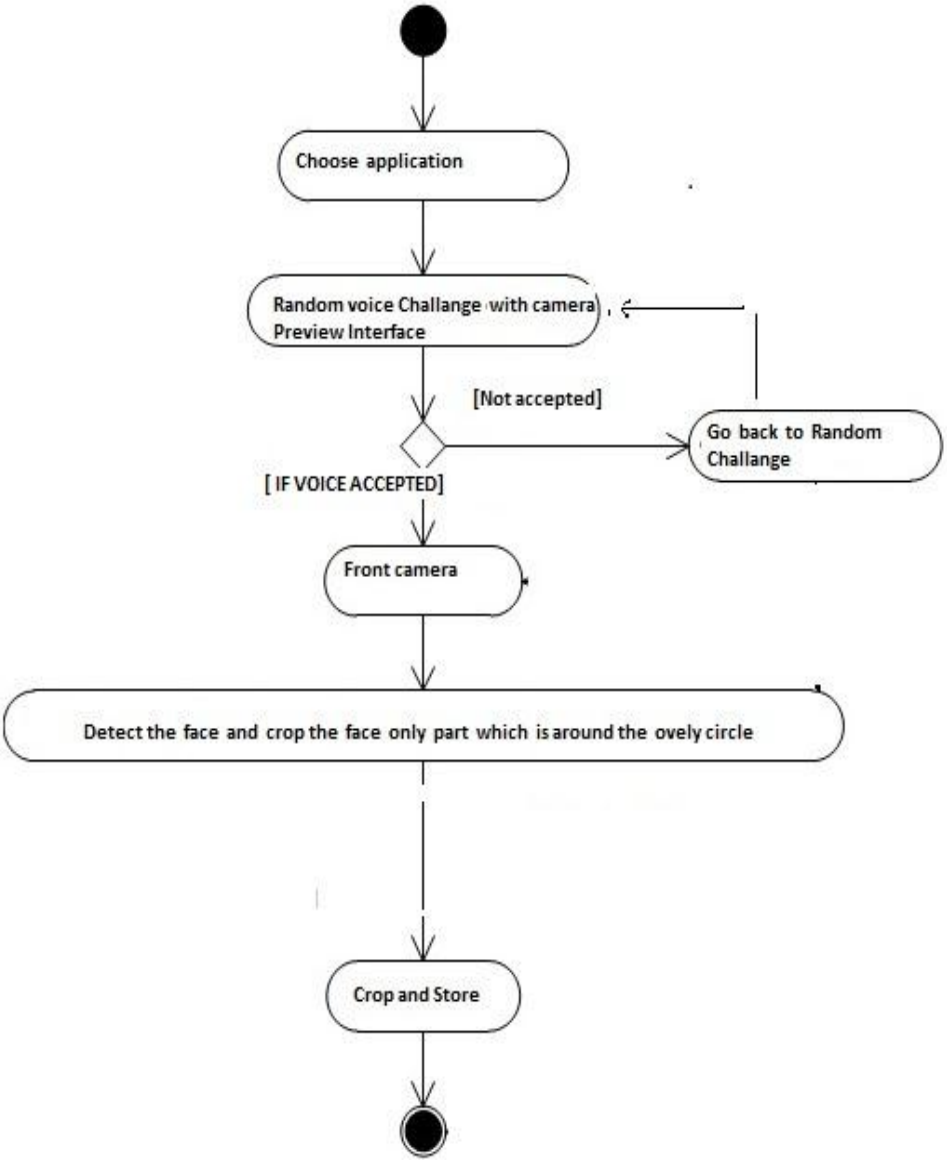


Figure 14 UML process of the ShowFace

5.3 Captured sample faces using ShowFace

The captured sample face form the camera is cropped around the face and it reduces bitmap which accelerates the processing with the server while ensuring that matching and authentication process is safe and faster.

There are some compare face snap shorts which is take by the ShowFace application.

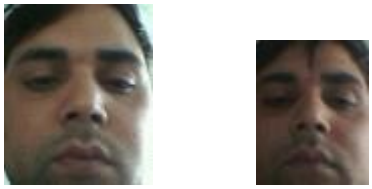


Figure 15 Face cropped after face detection

The goal of thesis is to capture the best accurate face samples for matching and comparing steps. Likewise, as we explained in previous chapter this show face application has implemented the face detection Android class to detect face by client camera**.** Face Detector is an useful Android class for detecting faces in a bitmap object. This class identify face objects in the given bitmap with an array of **Face** objects which contains information about each detected face such as the distance between the eyes, mid-point between eyes and the position of face with respect to the rotation of x, y and z axis.

Likewise, Android also developed the speaker recognition technology for the Android developer where the engine has sufficient response for our Show face random challenge. The engine used onCreate method which initialise the first created activity and handle the intents for the ACTION_RECOGNIZE_SPEECH.

The engine can handle the voice recognition with the start Voice Recognition Activity which can calls an activity to handle, after the calling the above activity it check to see the request is matches then passes it in for the required result or for the error then the onActivityResult is call-back to push out the Listview to display on the screen. So this is the speaker engine work for the random challenge.

So we also have initial verification process into the client side .If user successfully make challenge then automatically human face it will store by the client. But if it is not real face of human it automatically terminates. The main idea is to make the authentication more reliable for next matching step and comparing step in the server side.

5.4 Reducing the data processing in the Biometry Server

One of the goals of this thesis is also to reduce the data processing when validating the samples in the Biometry server. If there are numbers of frequent unnecessary images request from the client side then the data process and the load of the server will increase .In order to increase the efficiency of the server the sample taken by the show face application has to be exact and reliable for processing the verification steps . As a result it will be more effective and the assurance of the application will also be improved.

5.5 Summary

In this chapter, we saw how the ShowFace can overcome the Biometry application for the authentication process. We also discussed how the secure random challenge and auto capture crop face can be the best sample for the matching and validating steps .

Chapter 6

6.1 Conclusions and Future Work

6.1.2 Conclusions

The goal of current thesis was to develop a Mobile Challenge Recording Interface. This makes possible to provide a secure User Interface to capture the samples for the Biometry.com and uses a two-phase verification processes designed to minimize the noise that the website gets thus decreasing the server load while increasing the efficiency. We implemented Android face detection and Google speaker recognition system respectively in an Android platform. This thesis is tested on Android 4.0.3 Ice Cream Sandwich (developers, n.d.).

The main problems of Biometry application was faced while taking accurate samples within their user interface. The voice and face samples are using two different interfaces .The work provides an authentication interface with accurate samples for the matching and validation steps for enhancing the process, making it faster and more reliable and more accurate.

The authentication process is as follows:

First the ShowFace application opens with some randomly generated number with voice challenge in a camera preview overlay surface. Next it prompts the user to complete the voice challenge with the random number. If the voice challenge is passed, the next step is to detect the image bitmap which lies near the dotted circle. If no face is detected, the process is terminated and has to start from the beginning. If the face is detected, the picture of the face is cropped to reduce the file size. Despite the application is being in an initial development phase, the used components serve as a proof of concept. These extensions can be well integrated into the application for future enhancements .We believe that the developed can be valuable for the Biometry company in the authentication processes.

The core of the thesis was to study different biometric authentication techniques, derive requirements from these for a mobile interface, and implement and refine an interface prototype. The application of the thesis makes use of the Android speech recognition and face detector for the identification and verification purposes. One of the main advantages offered by this application is the reduction of bandwidth load to the Biometry server that is made possible two different features. First of all the qualifying audio challenge limits the spam and

unintended verification requests. On the other hand, the image of the face that is transmitted to the server is cropped thus it is reduced in size.

6.1.3 Future Work

Towards the end of development some new ideas emerged, which did not make it to the current thesis but can be extensively exploited in future papers and projects. Some of the rearrangements that we have come up with but not been able to implement due to time constraints are briefly explained below:

In the voice challenge, four numbers are generated randomly. The user dictates the first number, then an image is taken and the face is cropped. Then the user should voice the second number, after which a second face image is cropped. So four different images are cropped and saved after the each four numbers spoken in an order.

The face detection technology would be more real times if Opencv could be used .It is the emerging technology in Android platform which is open source of C and C++ libraries. It is BSD license based for research and commercial use .It has real time capture, face and body detection, basic image treatment as brightness, contrast, threshold and many other features that can be used to enhance the current features of the application.

Mobiilne biomeetriline tuvastusliides

Roshan Lamichhane

Magistritöö
Sisukokkuvõte

Autentimine arvutivõrkudes on protsess, mis üritab kontrollida digitaalset identiteeti, kasutades paroole. Parooli olemasolu aitab tagada kasutaja autentsuse. Kasutajaid saab biomeetiriliselt kindlaks teha ka nende sõrmejälje, hääle või näo järgi kasutades kindlat riistvara ja parooli toimingu kinnitamiseks. Samas, juhul kui paljud kasutajad on seotud sedatüüpi autentimisega, võib selline lahendus muutuda liiga aeglaseks ja suhteliselt kulukaks. Seetõttu vajab edukas biomeetriline autentimise süsteem põhjalikku läbimõeldust paljude tegurite osas (PBworks, 2007).

Minu magistritöö eesmärk on uurida erinevaid biomeetrilise autentimise meetodeid, luues kirjeldused mobiilseks kasutamiseks ning töötada välja ja realiseerida vastav prototüüp. Kui biomeetrilise testi andmed on kogutud, näiteks video või hale kaudu, siis saadud andmed läbivad töötluse, peale mida kasutatakse ainult unikaalset osa andmetest. See osa andmetest salvestatakse andmebaasi. Kuid, enne andmete kasutamist, tuleb telefonis rakendada mitmeid meetmeid.

Biomeetria puhul kasutatakse kahte peamist stsenaariumit: kontrollimine ja isikuandmete tuvastamine. Kontrollimise puhul võrreldakse ainult ühte andmebaasi, mis on kirjeldatav kui üks-ühele võrdulus. Isikuandmete tuvastamisel kasutatakse üks-mitmele andmebaaside võrdlust. Mõlemal juhul peavad salvestatud andembaasid toetama vastavaid stsenaariume. Magistritöös valmis vastav kasutajaliides nõutud omadustega.

Bibliography

Android, 2011. *Android.* [Online]
Available at: http://www.openhandsetalliance.com/android_overview.html
[Accessed 10 June 2012].

android, 2012. *android developer.* [Online]
Available at: http://developer.android.com/guide/basics/what-is-android.html
[Accessed 12 may 2012].

android, 2012. *android developer.* [Online]
Available at: http://developer.android.com/reference/android/speech/RecognizerIntent.html
[Accessed 11 may 2012].

android-apps, 2012. *http://www.android-app-market.com.* [Online]
Available at: http://www.android-app-market.com/android-activity-lifecycle.html
[Accessed 12 April 2012].

Anon., 2006. *http://www.processor.com.* [Online]
Available at: http://www.processor.com/Articles%5CPDFMagazine%5CGood%5CP___2811.pdf?guid=
[Accessed 3 MAY 2012].

Anon., 2012.. *Mobi Thinking.* [Online]
Available at: http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats#subscribers
[Accessed 01 MAY 2012].

Anon., n.d. *ElectroPoliticsImage.* [Online]
Available at: http://www.technomicon.com/TechnomiconImages/ElectroPoliticsImages/EP-11-15-10Images/JavaAndroidFlowchart.jpg
[Accessed 22 march 2012].

Biometry.com, 2010. *http://biometry.com/.* [Online]
Available at: Biometry.com
[Accessed 20 APRIL 2012].

Biometry, 2011. *Biometry.* [Online]
Available at: http://biometry.com/products.html
[Accessed 22 April 2012].

Cattin, P. C., 2002. *http://e-collection.library.ethz.ch.* [Online]
Available at: http://e-collection.library.ethz.ch/eserv/eth%3A25753/eth-25753-02.pdf
[Accessed 20 April 2012].

d'Aix-en-Provence, A. h. a. t. É. S. d., n.d. *OpenCV.* [Online]
Available at: http://ubaa.net/shared/processing/opencv/
[Accessed 8 May 2012].

developer, a., 2012. *android developer.* [Online]
Available at: http://developer.android.com/reference/android/app/Activity.html
[Accessed 22 april 2012].

Developer, A., 2012. *http://developer.android.com.* [Online]
Available at: http://developer.android.com/guide/basics/what-is-android.html
[Accessed 01 04 2012].

developers, g., n.d. *Binaries for Nexus Phones.* [Online]
Available at: https://developers.google.com/android/nexus/drivers#crespoiml74k
[Accessed 12 June 2012].

Emeraldinsight, 2010. *http://www.emeraldinsight.com.* [Online]
Available at: http://www.emeraldinsight.com/content_images/fig/1610220405001.png
[Accessed 2 march 2012].

Fergytech, d. n., 2010. *http://www.fergytech.com.* [Online]
Available at: http://www.fergytech.com/android/
[Accessed 12 may 2012].

google, 2011. *Voice Actions for Android.* [Online]
Available at: http://www.google.com/mobile/voice-actions/
[Accessed 17 June 2012].

Hibben, M. W., 2010. *http://www.technomicon.com.* [Online]
Available at: http://www.technomicon.com/ElectroPolitics/ElectroPolitics-11-15-10.html
[Accessed 02 April 2012].

http://gocsi.com, 2010. . [Online]
Available at: http://gocsi.com/survey
[Accessed 1 May 2012].

IBM, 2010. *Build in Not bolted on.* [Online]
Available at: http://www-03.ibm.com/systems/au/itsolutions/security/
[Accessed 03 May 2012].

Reed, J., n.d. *https://sites.google.com.* [Online]
Available at: https://sites.google.com/site/androidappcourse/labs/lab-1
[Accessed 15 april 2012].

sites, g., 2008. *https://sites.google.com.* [Online]
Available at: https://sites.google.com/site/io/anatomy--physiology-of-an-android
[Accessed 12 april 2012].

Stefano Arca, P. C. R., n.d. *http://homes.dsi.unimi.it.* [Online]
Available at: http://homes.dsi.unimi.it/~campadel/Articoli/AVBPA2003.pdf
[Accessed 20 April 2012].

Subesi, E. K. a. A., 2011. *http://www.nauss.edu.sa.* [Online]
Available at:
http://www.nauss.edu.sa/En/DigitalLibrary/Researches/Documents/2011/articles_2011_3137.pdf
[Accessed 20 April 2012].

Technologies, R., 2011. *http://www.ravirajtech.com.* [Online]
Available at: http://www.ravirajtech.com/facialrecognitionsolution.html
[Accessed 20 04 2012].

Appendix

Source Code

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:Android="http://schemas.Android.com/apk/res/Android"
    package="org.rider.cam"
    Android:versionCode="1"
    Android:versionName="1.0">
  <uses-sdk Android:minSdkVersion="10" />
<uses-feature Android:name="Android.hardware.camera" />
        <uses-feature Android:name="Android.hardware.camera.front"
                Android:required="false" />
        <uses-permission Android:name="Android.permission.CAMERA" />
        <uses-permission
Android:name="Android.permission.WRITE_EXTERNAL_STORAGE" />
        <uses-permission Android:name="Android.permission.RECORD_AUDIO" />
  <application Android:icon="@drawable/icon" Android:label="@string/app_name">
    <activity Android:name=".FaceDetectionClientActivity"
            Android:label="@string/app_name">
    </activity>
    <activity Android:name=".AndroidUIActivity"
            Android:label="@string/app_name"
            Android:screenOrientation="landscape"
            Android:theme="@Android:style/Theme.Black.NoTitleBar.Fullscreen">
        <intent-filter>
          <action Android:name="Android.intent.action.MAIN" />
          <category Android:name="Android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
                <activity Android:name=".TestActivity"
            Android:label="@string/app_name">
    </activity>
  </application>
</manifest>
```

XML Layout:

Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:Android="http://schemas.Android.com/apk/res/Android"
   Android:layout_width="fill_parent"
   Android:layout_height="match_parent"
   Android:orientation="vertical"
   Android:padding="10dip" >

   <ImageView
      Android:id="@+id/imageView1"
      Android:layout_width="280dp"
      Android:layout_height="wrap_content"
      Android:layout_weight="0.81"
      Android:src="@drawable/icon" />

      <TextView
      Android:layout_width="fill_parent"
      Android:layout_height="wrap_content"
      Android:text="@string/app_info"
      Android:gravity="center_horizontal"
      Android:layout_weight="1.0"/>

      <RatingBar
         Android:id="@+id/ratingBar1"
         Android:layout_width="wrap_content"
         Android:layout_height="wrap_content"
         Android:layout_weight="0.81" />



   <Button
      Android:layout_width="wrap_content"
      Android:layout_height="wrap_content"
      Android:id="@+id/take_picture"
      Android:layout_margin="5dip"
      Android:text="@string/take_picture"
      Android:layout_gravity="center_horizontal"/>
```

```
</LinearLayout>
```

Camera.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:Android="http://schemas.Android.com/apk/res/Android"
   Android:layout_width="match_parent"
   Android:layout_height="match_parent" >



   <SurfaceView
      Android:id="@+id/surface"
      Android:layout_width="match_parent"
      Android:layout_height="match_parent"
      Android:keepScreenOn="true" />




   <ImageView
      Android:id="@+id/imageView1"
      Android:layout_width="match_parent"
      Android:layout_height="match_parent"
      Android:padding="10dp"
      Android:scaleType="centerInside"
      Android:src="@drawable/circule" />


   <TextView
      Android:id="@+id/textView1"
      Android:layout_width="wrap_content"
      Android:layout_height="wrap_content"
      Android:layout_gravity="center_horizontal"
      Android:layout_weight="1"
      Android:gravity="center"
      Android:text="Five"
      Android:textAppearance="?Android:attr/textAppearanceLarge"
       />

   <ImageView
      Android:id="@+id/reloadView"
```

```
        Android:layout_width="wrap_content"
        Android:layout_height="wrap_content"
        Android:padding="10dp"
        Android:src="@drawable/reload" />

</FrameLayout>
```

Test.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:Android="http://schemas.Android.com/apk/res/Android"
   Android:layout_width="match_parent"
   Android:layout_height="match_parent"
   Android:orientation="vertical" >


   <Button
      Android:id="@+id/button1"
      Android:layout_width="wrap_content"
      Android:layout_height="wrap_content"
      Android:layout_gravity="center_horizontal"
      Android:text="Button" />

</LinearLayout>
```

Important java files

Face detection

```
int facesFound = detector.findFaces(bitmap565, faces);
                        PointF midPoint = new PointF();
                        float eyeDistance = 0.0f;
                        float confidence = 0.0f;
```

```
                    Log.i("FaceDetector", "Number of faces found: " + facesFound);

                    if (facesFound > 0) {
                            for (int index = 0; index < facesFound; ++index) {
                                    // Get the eye distance, detected eye mid point and
                                    // confidence
                                    faces[index].getMidPoint(midPoint);
                                    eyeDistance = faces[index].eyesDistance();
                                    confidence = faces[index].confidence();

                                    Log.i("FaceDetector", "Confidence: " + confidence
                                                    + ", Eye distance: " + eyeDistance
                                                    + ", Mid Point: (" + midPoint.x + ", " +
midPoint.y
                                                    + ")");

                            }
                    }

                    String filepath = Environment.getExternalStorageDirectory()
                                    + "/facedetect" + System.currentTimeMillis() + ".jpg";

                    try {
                            FileOutputStream fos = new FileOutputStream(filepath);

                            Bitmap finalBitmap = Bitmap.createBitmap(bitmap565,
                                            (int) (midPoint.x - eyeDistance),
                                            (int) (midPoint.y - eyeDistance),
                                            (int) (eyeDistance * 2), (int) (eyeDistance * 2 +
16))

                            ;

                            finalBitmap.compress(CompressFormat.JPEG, 90, fos);

                            fos.flush();
                            fos.close();
                            ImageView imageView = (ImageView)
findViewById(R.id.image_view);

                            imageView.setImageBitmap(finalBitmap);
                    } catch (FileNotFoundException e) {
                            e.printStackTrace();
                    } catch (IOException e) {
```

```
                    e.printStackTrace();

                }
```

Speaker Reconigation

```
intent.putExtra(RecognizerIntent.EXTRA_MAX_RESULTS, 5);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_PREFERENCE, "en-US");
            rec.startListening(intent);
    }
    public void setTarget(String text){
            target = text;
            targetArray = target.toCharArray();
            position = 0;
    }
    public void setSpeechMatchListener(SpeechMatchListener listener){
            matchListener = listener;
    }
    public void finish(){
            Log.e(TAG, "speech destroy");
            rec.destroy();
            matchListener = null;
            listener = null;
            rec = null;
    }
    public void startListening(){
            rec.startListening(intent);
    }
    private RecognitionListener listener = new RecognitionListener(){

            @Override
            public void onBeginningOfSpeech() {
                    // TODO Auto-generated method stub
                    Log.e(TAG, "speech begin");

            }
```