BIOMETRIC AUTHENTICATION FOR ONLINE EXAMINATION

K.KANIMOZHI

(12PCS008)

A Project Report submitted to
Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore-641043

In Partial Fulfillment of the Requirements for the Master's Degree in Computer Science

March, 2014

BIOMETRIC AUTHENTICATION FOR ONLINE EXAMINATION

K.KANIMOZHI

(12PCS008)

A Project Report Submitted to

Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore-641043

In Partial Fulfillment of the Requirements for the Master's Degree in Computer Science

March, 2014

Signature of the Supervisor **Signature of the Head of the Department**

Signature of the External Examiner

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I would like to express my sincere thanks to God Almighty, for his constant love and grace that he has showered upon me.

I am very grateful to **Dr.T.S.K.Meenakshi Sundaram, M.A., M.Phil., Ph.D.,** Chancellor, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for his support and encouragement during the course of my study.

I heartily thank **Dr**. **(Mrs.) Sheela Ramachandran M.Sc., P. G. Dip., Ph.D.,** Vice Chancellor, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for extending all resources that facilitated the conduct of the present study.

I express my humble gratitude to **Dr. (Mrs.) Gowri Ramakrishnan M.Sc., M. Phil., Ph.D.,** Registrar, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing all facilities necessary for the study.

I am also thankful to **Dr.(Mrs) A.Parvathi M.Sc., Dip.Ed., M.Phil., Ph.D.,Dean** ,Faculty of Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for granting the facility required.

I wish to place on record my deep sense of gratitude to **Dr.(Mrs.)G.Padmavathi M.Sc., M.Phil., Ph.D.,** Professor and Head, Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing all the facilities to complete the project.

I take this unique opportunity to express my sincere thanks to my project Coordinator **Mrs.N.Valliammal M.Sc.,M.Phil., Assistant Professor,Department of Computer Science**, for her kind advice and knowledgeable suggestion, which helped me to complete my project successfully.

I owe great deal of gratitude to my esteemed guide **Mrs.N.Valliammal M.Sc.,M.Phil., Assistant Professor,** Department of Computer Science, for her guidance, imparting the tremendous assistance and knowledgeable suggestion, which helped me to complete my project successfully.

I express my humble gratitude to **Dr. (Mrs.).Saroja Prabhakaran, M.A,Dip.Ed.,Ph.D., Director, Hall of Residence**, for her support in completion of the project.

I have great pleasure in expressing my deep sense of gratitude to all other staffs and non-teaching staffs who stood behind the screen in making of project.

I would extend my hearty thanks to one and all that helped me directly or indirectly for successful completion of my project.

Last yet importantly, I would like to thank my parents, my grandfather, my brother and all my well-wishers for their kind inspiration.

SYNOPSIS

SYNOPSIS

The project entitled as "**FACE RECOGNITION"** recognize a sample face from a set of given authenticated student faces in order to provide more security. In this project principle component analysis (Eigen face approach) is applied to recognize a student face that is a face under different lightening and emotional condition.

For comparison and experimental analysis simple approach such as user name and password based authentication and finger print based authentication are used. The final result is analysed and the face recognition method produces best result.

The present invention of face recognition based authentication involves two phases such as, face detection which is the primary process and face recognition which is an authenticating phase.

Face detection involves four main concepts. Firstly, face localization which separates parameter space and object space using Hough method and skin color information method. The Second step is face normalization which extracts only the face by discarding all the surroundings. Third step is to locate facial characteristics using neural network .Finally the student face is extracted using Eigen face approach.

After extracting the features of the face, all these features will be basically stored as a template that will be used for recognition. In recognition phase, the student face is captured and checked for authentication. Only if the face matches with the store template the student will be allowed for the examination.

Like single face recognition multi face recognition system uses principle component analysis (PCA) technique. To perform PCA five steps to be undertaken. The first step is subtracting the Mean of the data from each variable. The second step is calculating and forms a covariance Matrix. The third step is calculating Eigenvectors and Eigen values from the covariance Matrix. The fourth step is to choose a Feature Vector. Final step is multiply the transposed Feature Vectors by the transposed adjusted data.

CONTENTS

CONTENTS

INTRODUCTION

1. INTRODUCTION

1.1. OVERVIEW OF THE PROJECT

Automatic skin detection is a common primitive for a range of human related image processing applications, one of which includes face detection. Pixel level skin searches can be very rapid and can greatly reduce the search space prior to higher level classification. Thus such detections are often used as front-end primitives to higher level person and face location system.

Complexity reduced face detection using probability-Based face mask pre filtering (Hough method) and pixel-Based Hierarchial Feature Adaboosting (face color information method) is used to detect the face is used because the training time is significantly reduced and the detection rate remains competitive to the traditional Adaboosting method.

It is an effective Eigen face approach and the implementation is straightforward. The simplicity and computation efficiency make this approach an excellent candidate for real-time surveillance system.

Multi Face Detection system provides a solution that can automatically detect faces in still images and real time video feeds. The system can detect an arbitrary number of faces at any scale and location.

The system takes photographic images or a video stream as input. The output consists of an array of rectangles which corresponds to the location and scale of faces detected. If it detects no faces, it will return an empty array.

SYSTEM CONFIGURATION

2. SYSTEM CONFIGURATION

2.1 HARDWARE SPECIFICATION

Processors will continue to get faster, smaller and cheaper, whereas memory will continue to get faster, larger and cheaper .The trend except to have reasonable memory to accompany a powerful processor.

PROCESSOR	:	Intel Pentium i3 1.8 GHz
MOTHERBOARD	:	Intel 915GVSR chipset board
RAM	:	4 GB DDR 3RAM
HARD DISK DRIVE	:	500 GB
FLOPPY DRIVE	:	1.44 MB
DVD/CD DRIVE	:	Sony 52 x Dual layer drive
MONITOR	:	17” Color TFT Monitor
KEYBOARD	:	Multimedia Keyboard 108 Keys
MOUSE	:	Logitech Optical Mouse
CABINET	:	ATX iball.
HUB	:	Compex 16 lines.
BANDWIDTH	:	100 mbps.

2.2 SOFTWARE SPECIFICATION

When an application project is considered the three basic software requirements are the platforms in which the project is developed, the front-end tool that provides the interaction with the users and the back-end tool that stores the data.

FRONTEND	:	VB.NET 2008
BACKEND	:	MS Access 2005
CODING LANGUAGE	:	C#
OPERATING SYSTEMS	:	Microsoft windows XP
DOCUMENTATION	:	Microsoft word 2007.
SCRIPTING LANGUAGE	:	Java Script

2.3 ABOUT THE SOFTWARE

2.3.1ABOUT FRONT END

.NET

This topic discusses about how Businesses are related to Internet, what .NET means to them and what exactly .NET is built upon. As per the product documentation from a Business perspective, there are three phases of the Internet. The First phase gets back to the early 1990's when Internet first came into general use and which brought a big revolution for Businesses. First phase describes about the Internet Businesses designed and launched their Website's and focused on the number of hits to know how many customers were visiting their site and interested in their products, etc. The Second phase is what we are in right now and in this phase Businesses are generating revenue through Online Transactions. We are now moving into the Third phase of the Internet where profit is the main priority. The focus here is to Businesses effectively communicated with their customers and partners, who are geographically isolated, participate in Digital Economy and deliver a wide range of services.

What is .NET ?

.NET is Microsoft's strategy of software that provides services to people any time, any place, on any device. An accurate definition of .NET is, it's an XML Web Services platform which allows us to build rich .NET applications, which allows users to interact with the Internet using wide range of smart devices (tablet devices, pocket PC's, web phones etc), which allows to build and integrate Web Services and which comes with many rich set of tools like Visual Studio to fully develop and build those applications.

What are Services?

Web Services are the applications that run on a Web Server and communicate with other applications. It uses a series of protocols to respond to different requests. The protocols on which Web Services are built are summarized below:

UDDI: Stands for Universal Discovery and Description Integration. It's said to be the Yellow Pages of Web Services which allows Businesses to search for other

Businesses allowing them to search for the services it needs, know about the services and contact them.

WSDL: Stands for Web Services Description Language, often called as whiz-dull. WSDL is an XML document that describes a set of SOAP messages and how those messages are exchanged.

SOAP: Stands for Simple Object Access Protocol. It's the communication protocol for Web Services.

XML, **HTTP** and **SMTP**: Stands for Extensible Markup Language, Hyper Text Transfer Protocol and Simple Message Transfer Protocol respectively. UDDI, WSDL and SOAP rely on these protocols for communication.

The image below shows the order of the protocols on which Web Services are built:

Example of a Web Services Application

If a customer accesses a Website and buys something .The Web services of the business will communicate with the inventory system to see if there is enough stock to fulfill the order. If not, the system can communicate with the suppliers to find one or all of the parts that make up the order before filling the order. At all stages the customer will be kept informed via messages. The end result is a seamless system communicating and exchanging information easily regardless of the platform they are all running on. The business don't need to worry about going to the wrong supplier because it asks the Web service running on the supplier system what it does. And the business doesn't have to worry about the other system's methods of handling data because they communicate via SOAP and XML.

Real World Application

Microsoft's passport service is an example of a .NET service. Passport is a Web-based service designed to make signing in to Websites fast and easy. Passport enables participating sites to authenticate a user with a single set of sign-in credentials eliminating the need for users to remember numerous passwords and sign-in names. You can use one name and password to sign in to all .NET Passport-participating sites and services. You can store personal information in your .NET Passport profile and, if

you choose, automatically share that information when you sign in so that participating sites can provide you with personalized services. If you use Hotmail for your email if needs .

What is .NET Built On?

.NET is built on the Windows Server System to take major advantage of the OS and which comes with a host of different servers which allows for building, deploying, managing and maintaining Web-based solutions. The Windows Server System is designed with performance as priority and it provides scalability, reliability, and manageability for the global, Web-enabled enterprise. The Windows Server System integrated software products are built for interoperability using open Web standards such as XML and SOAP.

Core Windows Server System Products include :

SQL Server2000: This Database Server is Web enabled and is designed with priority for .NET based applications. It is scalable, easy to manage and has a native XML store.

Application Center 2000: This product is designed to manage Web Applications.

Commerce Server 2000: This powerful Server is designed for creating E-Commerce based applications.

Mobile Information Server: This Server provides real-time access for the mobile community. Now Outlook users can use their Pocket PC's to access all their Outlook data while they are moving.

Exchange Server 2000: This is a messaging system Server and allows applications on any device to access information and collaborate using XML.

BizTalk Server 2000: This is the first product created for .NET which is XML based and allows to build business process that integrate with other services in the organization or with other Businesses.

Internet Security and Acceleration Server 2000: This Server provides Security and Protection for machines. It is an integrated firewall and Web cache server built to make the Web-enabled enterprise safer, faster, and more manageable.

Host Integration Server 2000: This Server allows for the Integration of mainframe systems with .NET.

When developing real world projects if you don't know how to use the above mentioned Server's which are built for .NET based applications do not worry. Your System Administrator is always there to help you.

.NET and XML

There is a lot of connection between XML and .NET. XML is the glue that holds .NET together. XML looks similar to HTML which is readable and text-based. XML is a method of putting structured data into a text file. XML is the specification for defining the structure of the document. Around this specification a whole family of optional modules are being developed. The reason why XML is linked so much to .NET is, it's platform independent and is well supported on any environment. To move the data contained in an XML file around different organizations using different software on different platforms it should be packed it into a protocol like SOAP.

About SOAP

SOAP, Simple Object Access Protocol is a simple, lightweight protocol for exchanging information between peers in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelop that describes what is in the message and how it should be processed, a set of encoding rules and a convention for representing remote procedure calls and responses.

.NET Framework

.NET is a "Software Platform". It is a language-neutral environment for developing rich .NET experiences and building applications that can easily and securely operate within it. When developed applications are deployed, those applications will target .NET and will execute wherever .NET is implemented instead

of targeting a particular Hardware/OS combination. The components that make up the .NET platform are collectively called the .NET Framework.

The .NET Framework is a managed, type-safe environment for developing and executing applications. The .NET Framework manages all aspects of program execution, like, allocation of memory for the storage of data and instructions, granting and denying permissions to the application, managing execution of the application and reallocation of memory for resources that are not needed.

The .NET Framework is designed for cross-language compatibility. Cross-language compatibility means, an application written in Visual Basic .NET may reference a DLL file written in C# (C-Sharp). A Visual Basic .NET class might be derived from a C# class or vice versa.

Common Language Runtime (CLR)

The CLR is described as the "execution engine" of .NET. It provides the environment within which the programs run. It's this CLR that manages the execution of programs and provides core services, such as code compilation, memory allocation, thread management, and garbage collection. Through the Common Type System (CTS), it enforces strict type safety, and it ensures that the code is executed in a safe environment by enforcing code access security. The software version of .NET is actually the CLR version.

Working of the CLR

When the .NET program is compiled, the output of the compiler is not an executable file but a file that contains a special type of code called the Microsoft Intermediate Language (MSIL), which is a low-level set of instructions understood by the common language run time. This MSIL defines a set of portable instructions that are independent of any specific CPU. It's the job of the CLR to translate this Intermediate code into a executable code when the program is executed making the program to run in any environment for which the CLR is implemented. And that's how the .NET Framework achieves Portability. This MSIL is turned into executable code using a JIT (Just In Time) complier. The process goes like this, when .NET programs are executed, the CLR activates the JIT complier. The JIT complier

converts MSIL into native code on a demand basis as each part of the program is needed. Thus the program executes as a native code even though it is compiled into MSIL making the program to run as fast as it would if it is compiled to native code but achieves the portability benefits of MSIL.

Class Libraries

Class library is the second major entity of the .NET Framework which is designed to integrate with the common language runtime. This library gives the program access to runtime environment. The class library consists of lots of prewritten code that all the applications created in VB .NET and Visual Studio .NET will use. The code for all the elements like forms, controls and the rest in VB .NET applications actually comes from the class library.

Common Language Specification (CLS)

If we want the code which we write in a language to be used by programs in other languages then it should adhere to the Common Language Specification (CLS). The CLS describes a set of features that different languages have in common. The CLS defines the minimum standards that .NET language compilers must conform to, and ensures that any source code compiled by a .NET compiler can interoperate with the .NET Framework.

Some reasons why developers are building applications using the .NET Framework:

- Improved Reliability
- Increased Performance
- Developer Productivity
- Powerful Security
- Integration with existing Systems
- Ease of Deployment
- Mobility Support
- XML Web service Support
- Support for over 20 Programming Languages
- Flexible Data Access

Windows XP for specialized hardware

The main menu from Windows XP Media Center Edition 2005.

Microsoft has also customized Windows XP to suit different markets. Six different versions of Windows XP for specific hardware were designed, two of them specifically for 64-bit processors.

Windows XP 64-bit Edition

This edition was designed specifically for Itanium-based workstations. This edition was discontinued in early 2005, after HP, the last distributor of Itanium-based workstations, stopped selling Itanium systems marketed as 'workstations'. However, Itanium support continues in the server editions of Windows.

Windows XP Professional x64 Edition

Not to be confused with the aforementioned Itanium edition of Windows XP, this edition is derived from Windows Server 2003 and supports the x86-64 extension of the Intel IA-32 architecture. x86-64 is implemented by AMD as "AMD64", found in AMD's Opteron and Athlon 64 chips, and implemented by Intel as "Intel 64" (formerly known as IA-32e and EM64T), found in Intel's Pentium 4 and later chips.

Microsoft had previously supported other microprocessors with earlier versions of the Windows NT operating system line (including two 64-bit lines, the DEC Alpha and the MIPS R4000, although Windows NT used them as 32-bit processors). The files necessary for all of the architectures were included on the same installation CD and did not require the purchase of separate versions.

Windows XP Media Center Edition

This edition is designed for media center PCs. Originally, it was only available bundled with one of these computers, and could not be purchased separately. In 2003 the Media Center Edition was updated as "Windows XP Media Center Edition 2003", which added additional features such as FM radio tuning. Another update was released in 2004, and again in 2005, which was the first edition available for System Builders. Many of the features of Windows XP Media Center Edition 2005 (including screen dancers, auto playlist DJ, and high end visual screen savers) were taken from the Windows XP Plus! packages. These were originally shipped as add ons to Windows XP to increase the users experience of their Windows XP machine.

Windows XP Tablet PC Edition

Intended for specially designed notebook/laptop computers called tablet PCs, the Tablet PC Edition is compatible with a pen-sensitive screen, supporting handwritten notes and portrait-oriented screens. It cannot be purchased separately from a Tablet PC without an MSDN (Microsoft Developers Network) subscription.

Windows XP Embedded

An edition for specific consumer electronics, set-top boxes, kiosks/ATMs, medical devices, arcade video games, point-of-sale terminals, and Voice over Internet Protocol (VoIP) components.

Windows Fundamentals for Legacy PCs

In July 2006, Microsoft introduced a "thin-client" version of Windows XP called **Windows Fundamentals for Legacy PCs**, which targets older machines (as early as the original Pentium). It is only available to Software Assurance customers. It is intended for those who would like to upgrade to Windows XP to take advantage of its security and management capabilities, but can't afford to purchase new hardware.

Windows XP Starter Edition

Windows XP Starter Edition is a lower-cost version of Windows XP available in Thailand, Turkey, Malaysia, Indonesia, Russia, India, Colombia, Brazil, Argentina, Peru, Bolivia, Chile, Mexico, Ecuador, Uruguay and Venezuela. It is similar to Windows XP Home, but is limited to low-end hardware, can only run 3 programs at a time, and has some other features either removed or disabled by default.

According to a Microsoft press release, Windows XP Starter Edition is "a low-cost introduction to the Microsoft Windows XP operating system designed for first-time desktop PC users in developing countries.

Specializations

The Starter Edition includes some special features for certain markets where consumers may not be computer literate. Not found in the Home Edition, these include localized help features for those who may not speak English, a country-specific computer wallpaper and screensavers, and other default settings designed for easier use than typical Windows XP installations.

In addition, the Starter Edition also has some unique limitations to prevent it from displacing more expensive versions of Windows XP. Only three applications can

be run at once on the Starter Edition, and each application may open a maximum of three windows. The maximum screen resolution is 1024×768, and there is no support for workgroup networking or domains. In addition, the Starter Edition is licensed only for low-end processors like Intel's Celeron or AMD's Duron and Sempron. There is also a 512 MB limit on main memory and a 120 GB disk size limit[(Microsoft has not made it clear, however, if this is for total disk space, per partition, or per disk). There are also fewer options for customizing the themes, desktop, and taskbar.

Market adoption

On October 9, 2006, Microsoft announced that they reached a milestone of 1,000,000 units of Windows XP Starter Edition sold. In the mass market, however, the Starter Edition has not had much success. Many markets where it is available have seen the uptake of cracked or pirated versions of the software instead.

2.3.2 ABOUT BACK END

SQL Server

SQL was originally invented by IBM researchers and has since become a strategic product for IBM based on its flagship DB2 database. SQL support is available on all major IBM product families, from personal computers through midrange systems (AS/400 and UNIX-based servers) to IBM mainframes. IBM’s initial work provided a clear signal of IBM’s direction for other database and system vendors to follow early in the development of SQL and relational databases. Later, IBM’s commitment and broad support speeded the market acceptance of SQL. IBM’s SQL reach today extends well beyond its own computer systems business. SQL-based products that IBM has developed or acquired now run across a broad range of hardware, in many cases from competing computer vendors such as Sun or Hewlett-Packard.

Microsoft Commitment (SQL Server, ODBC, and ADO)

Microsoft has long considered database access a key part of its Windows personal computer software architecture. Both desktop and server versions of Windows provide standardized relational database access through Open Database Connectivity (ODBC), a SQL-based call-level API. Leading Windows software applications (spreadsheets, word processors, databases, etc.) from Microsoft and other vendors

support ODBC, and all leading SQL databases provide ODBC access. Microsoft has enhanced ODBC support with higher-level, more object-oriented database access layers as part of its Object Linking and Embedding technology (OLE DB), and more recently as part of Active/X (Active/X Data Objects, or ADO). When Microsoft began its effort in the late 1980s to make Windows a viable server operating system, it introduced SQL Server as its own SQL-based offering. SQL Server continues today as a flagship Microsoft product, and a key component of its .NET architecture for web services.

Relational Foundation

SQL is a language for relational databases, and it has become popular along with the relational database model. The tabular, row/column structure of a relational database is intuitive to users, keeping the SQL language simple and easy to understand. The relational model also has a strong theoretical foundation that has guided the evolution and implementation of relational databases. Riding a wave of acceptance brought about by the success of the relational model, SQL has become the database language for relational databases.

High-Level, English-Like Structure

SQL statements look like simple English sentences, making SQL easy to learn and understand. This is in part because SQL statements describe the *data* to be retrieved, rather than specifying *how* to find the data. Tables and columns in a SQL database can have long, descriptive names. As a result, most SQL statements "say what they mean" and can be read as clear, natural sentences.

Interactive, Ad Hoc Queries

SQL is an interactive query language that gives users ad hoc access to stored data. Using SQL interactively, a user can get answers even to complex questions in minutes or seconds, in sharp contrast to the days or weeks it would take for a programmer to write a custom report program. Because of SQL's ad hoc query power, data is more accessible and can be used to help an organization make better, more informed decisions. SQL's ad hoc query capability was an important advantage over

nonrelational databases early in its evolution and more recently has continued as a key advantage over pure object-based databases.

Programmatic Database Access

SQL is also a database language used by programmers to write applications that access a database. The same SQL statements are used for both interactive and programmatic access, so the database access parts of a program can be tested first with interactive SQL and then embedded into the program. In contrast, traditional databases provides one set of tools for programmatic access and a separate query facility for ad hoc requests, without any synergy between the two modes of access.

Multiple Views of Data

Using SQL, the creator of a database can give different users of the database different *views* of its structure and contents. For example, the database can be constructed so that each user sees data only for his or her department or sales region. In addition, data from several different parts of the database can be combined and presented to the user as a simple row/column table. SQL views can thus be used to enhance the security of a database and tailor it to the particular needs of individual users.

Complete Database Language

SQL was first developed as an ad hoc query language, but its powers now go far beyond data retrieval. SQL provides a complete, consistent language for creating a database, managing its security, updating its contents, retrieving data, and sharing data among many concurrent users. SQL concepts that are learned in one part of the language can be applied to other SQL commands, making users more productive.

Dynamic Data Definition

Using SQL, the structure of a database can be changed and expanded dynamically, even while users are accessing database contents. This is a major advance over static data definition languages, which prevented access to the database while its structure was being changed. SQL thus provides maximum flexibility, allowing a database to adapt to changing requirements while online applications continue uninterrupted.

Client/Server Architecture

SQL is a natural vehicle for implementing applications using a distributed, client/ server architecture. In this role, SQL serves as the link between "front-end" computer systems optimized for user interaction and "back-end" systems specialized for database management, allowing each system to do what it does best. SQL also allows personal computers to function as front-ends to network servers or to larger minicomputer and mainframe databases, providing access to corporate data from personal computer applications.

Enterprise Application Support

The largest enterprise applications that support the daily operation of large companies and organizations all use SQL-based databases to store and organize their data. Data about business transactions (orders, sales amounts, customers, inventory levels, payment amounts, etc.) tends to have a structured, records-and-fields format, which converts easily into the row/column format of SQL. By constructing their applications to use enterprise-class SQL databases, major application vendors eliminate the need to develop their own data management software and can benefit from existing tools and programming skills. Because every major enterprise application requires a SQL-based database for its operation, new sales of enterprise applications automatically generate "drag-along" demand for new copies of database software.

Extensibility and Object Technology

The major challenge to SQL's continued dominance as a database standard has come from the emergence of object-based programming, and the introduction of object-based databases as an extension of the broad market trend toward object-based technology. SQL-based database vendors have responded to this challenge by slowly expanding and enhancing SQL to include object features. These "object/relational" databases, which continue to be based on SQL, have emerged as a more popular alternative to "pure object" databases and have perpetuated SQL's dominance through the last decade. The newest wave of object technology, embodied in the XML standard and web services architectures, has once again created a crop of "XML databases" and alternative query languages to challenge SQL. Previous history tends

to suggest that XML-based extensions to SQL and the relational model will once again meet this challenge and insure SQL's continuing importance.

Internet Database Access

With the exploding popularity of the Internet and the World Wide Web, and their standards-based foundation, SQL found a new role in the late 1990s as an Internet data access standard. Early in the development of the Web, developers needed a way to retrieve and present database information on web pages and used SQL as a common language for database gateways. More recently, the emergence of three-tiered Internet architectures with distinct thin client, application server and database server layers, has established SQL as the standard link between the application and database tiers. In the future, the role of SQL in the Internet will extend beyond web site architectures to include data management for collaborating applications and distributed objects in a web services architecture.

Java Integration (JDBC)

A major area of SQL development over the last five to ten years has been the integration of SQL with Java. Seeing the need to link the Java language to existing relational databases, Sun Microsystems (the creator of Java) introduced Java Database Connectivity (JDBC), a standard API that allows Java programs to use SQL for database access. JDBC received a further boost when it was adopted as the data access standard within the Java2 Enterprise Edition (J2EE) specification, which defines the operating environment provided by all of the leading Internet application servers. In addition to its role as a programming language from which databases are used, many of the leading database vendors have also announced or implemented Java support *within* their database systems, allowing Java to be used as a language for stored procedures and business logic within the database itself. This trend toward integration between Java and SQL will insure the continued importance of SQL in the new era of Java-based programming.

SYSTEM STUDY & ANALYSIS

3. SYSTEM STUDY AND ANALYSIS

The aim of the system study is primarily to understand the problem. The study of present system is through available documentation such as procedures, manuals, documents and the interview of the user staff and onsite observations.

3.1 FEASIBILITY STUDY

Systems analysis is the process of examining a business situation for the purpose of developing a system solution to a problem or devising improvements to such a situation. Before the development of any system can begin, a project proposal is prepared by the users of the potential system and/or by systems analysts and submitted to an appropriate managerial structure within the organization.

PROJECT PROPOSAL

The project proposal is the attempt to respond to or take advantage of a particular situation and is an essential element for correctly launching the system analysis. Although there are no hard and fast rules as to the form and content of the project proposal, the proposal should address the following points:

- Specifies the business situation or problem.
- The significance of the problem to the organization.
- Alternative solutions.
- The possible use of computer information systems to solve the problem.
- The various people interested in or possessing knowledge relevant to the problem.

System projects that are to be shared by a number of departments and users are usually approved by a committee rather than an individual. A project proposal is submitted to a committee that determines the merits of the proposal and decides whether or not to approve it. The committee is made up of people from various functional areas of the organization who have an interest in the operation and information of the proposed system.

THE SYSTEMS DEVELOPMENT LIFE CYCLE

The systems development life cycle (SDLC) describes a set of steps that produces a new computer information system. The SDLC is a problem-solving process. Each step in the process delineates a number of activities. Performing these activities in the order prescribed by the SDLC will bring about a solution to the business situation. The SDLC process consists of the following phases:

1. **Preliminary investigation**—the problem is defined and investigated.
2. **Requirements definition**—the specifics of the current system as well as the requirements of the proposed new system are studied and defined.
3. **Systems design**—a general design is developed with the purpose of planning for the construction of the new system.
4. **Systems development**—the new system is created.
5. **System installation**—the current operation is converted to run on the new system.
6. **Systems evaluation and monitoring**—the newly operational system is evaluated and monitored for the purpose of enhancing its performance and adding value to its functions.
7. Looping back from a later phase to an earlier one may occur if the need arises.

Each phase has a distinct set of unique development activities. Some of these activities may span more than one phase. The management activity tends to be similar among all phases.

The SDLC is not standardized and may be unique to a given organization. In other words, the names and number of phases may differ from one SDLC to the next. However, the SDLC discussed here is, to a large extent, representative of what is typically adopted by organizations.

At each phase certain activities are performed, the results of these activities are documented in a report identified with that phase. Management reviews the results of the phase and determines if the project is to proceed to the next phase.

The first two phases of the SDLC process constitute the systems-analysis function of a business situation. The following discussion will concentrate on phase one (Preliminary Investigation) and phase two (Requirements Definition) of the outlined SDLC process.

PRELIMINARY INVESTIGATION

The first phase of the systems development life cycle is preliminary investigation. Due to limited resources an organization can undertake only those projects that are critical to its mission, goals, and objectives. Therefore, the goal of preliminary investigation is simply to identify and select a project for development from among all the projects that are under consideration. Organizations may differ in how they identify and select projects for development. Some organizations have a formal planning process that is carried out by a steering committee or a task force made up of senior managers. Such a committee or task force identifies and assesses possible computer information systems projects that the organization should consider for development. Other organizations operate in an ad hoc fashion to identify and select potential projects. Regardless of the method used and after all potential projects have been identified, only those projects with the greatest promise for the well-being of the organization, given available resources, are selected for development.The objectives of the systems-investigation phase are:

- The preliminary-investigation phase sets the stage for gathering information about the current problem and the existing information system. This information is then used in studying the feasibility of possible information systems solutions.
- It is important to note that the source of the project has a great deal to do with its scope and content. For example, a project that is proposed by top management usually has a broad strategic focus. A steering committee proposal might have a focus that covers a cross-function of the organization. Projects advanced by an individual, a group of individuals, or a department may have a narrower focus.
- A variety of criteria can be used within an organization for classifying and ranking potential projects. For planning purposes, the systems analyst—with the assistance of the stakeholders of the proposed project—collects

information about the project. This information has a broad range and focuses on understanding the project size, costs, and potential benefits. This information is then analyzed and summarized in a document that is then used in conjunction with documents about other projects in order to review and compare all possible projects. Each of these possible projects is assessed using multiple criteria to determine feasibility.

- The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution. The analyst conducting the study gathers information using a variety of methods, the most popular of which are:

  - Interviewing users, employees, managers, and customers.
  - Developing and administering questionnaires to interested stakeholders, such as potential users of the information system.
  - Observing or monitoring users of the current system to determine their needs as well as their satisfaction and dissatisfaction with the current system.
  - Collecting, examining, and analyzing documents, reports, layouts, procedures, manuals, and any other documentation relating to the operations of the current system.
  - Modeling, observing, and simulating the work activities of the current system.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components. These components are:

- Economic feasibility—the economic viability of the proposed system. The proposed project's costs and benefits are evaluated. Tangible costs include fixed and variable costs, while tangible benefits include cost savings, increased revenue, and increased profit. A project is approved only if it covers its cost in a given period of time. However, a project may be approved only on its intangible benefits such as those relating to government regulations, the image of the organization, or similar considerations.

❖ Technical feasibility—the possibility that the organization has or can procure the necessary resources. This is demonstrated if the needed hardware and software are available in the marketplace or can be developed by the time of implementation.

❖ Operational feasibility—the ability, desire, and willingness of the stakeholders to use, support, and operate the proposed computer information system. The stakeholders include management, employees, customers, and suppliers. The stakeholders are interested in systems that are easy to operate, make few, if any, errors, produce the desired information, and fall within the objectives of the organization.

REQUIREMENTS DEFINITION

This phase is an in-depth analysis of the stakeholders information needs. This leads to defining the requirements of the computer information system. These requirements are then incorporated into the design phase. Many of the activities performed in the requirements definition phase are an extension of those used in the preliminary investigation phase. The main goal of the analyst is to identify what should be done, not how to do it. The following is a discussion of the activities involved in requirements definition.

INFORMATION NEEDS OF THE STAKEHOLDERS

Analysis of the information needs of the stakeholders is an important first step in determining the requirements of the new system. It is essential that the analyst understands the environment in which the new system will operate. Understanding the environment means knowing enough about the management of the organization, its structure, its people, its business, and the current information systems to ensure that the new system will be appropriate.

THE CURRENT INFORMATION SYSTEM

A comprehensive and detailed analysis of the current system is essential to developing a quality, new information system. The analyst should understand and document how the current system uses hardware, software, and people to accept and manage input data and to convert such data into information suitable for decision making. The documentation should be detailed and complete. For example, the

analyst should assess the quality of input and output activities that form the user's interface. In addition, the volume and timing of such activities may be documented.

THE CAPABILITIES OF THE COMPUTER INFORMATION SYSTEM

Functional requirements include the necessary hardware and software configurations along with the appropriate human resources. Specific functional requirements often include the following:

- User interface requirements—the input and output needs of the user that must be provided for by the new computer information system. These needs include layouts and definitions of input and output, volume, frequency, origination of input, and destination for reports.

- Processing requirements—the activities required for converting input into output, including calculations, decision rules, database operations, and other processing operations. In addition, requirements concerning capacity, throughput, turnaround time, response time, and the system's availability time are established.

- Storage requirements—the organization, content, and size of databases, and types and frequency of updates and inquiries. Furthermore, backup procedures and the length of time and rationale for retention of backups are delineated.

- Control requirements—the accuracy, validity, security, and adaptability requirements for the system's input, processing, output, and databases. Crash recovery and auditing requirements of the organization are further specified in this stage.

The analysis team, at the end of this phase, produces a document containing the functional requirements of the new computer information system. Additionally, the document contains preliminary schedules and a budget for the next phase. The task force or committee responsible for the project studies the document for the purpose of approving or not approving the work of the analysis team. In addition, the analysis team provides the committee with a demonstration. In essence, the analysis team walks the committee members, step by step, through the requirements definition phase. If the committee approves this phase, then the analysis team is funded and given the go-ahead to proceed to the next phase. However, if the committee does not approve this phase, then either the project is cancelled or, after appropriate modifications, the analysis team resubmits a new document to the committee.

A walk-through starts with a description of the project. From this point, the analysts delineate a set of well-defined goals, objectives, and benefits of the computer information system. Following that, the budgets and staffing requirements are articulated and the plans are shared with the committee. Specific, planned tasks are compared to actual accomplishments, and deviations, if any, are noted and accounted for. The plans for asset protection and business control are reviewed with the committee members. Finally, the analysts seek the committee's approval of the objectives, plans, time table, and budget for the next phase systems design.

Advantages of making Feasibility study

There are many advantages of making feasibility study some of which are summarized below:

- This study being made as the initial step of software development life cycle has all the analysis part in it which helps in analyzing the system requirements completely.
- Helps in identifying the risk factors involved in developing and deploying the system.
- The feasibility study helps in planning for risk analysis..
- Feasibility study helps in making cost/benefit analysis which helps the organization and system to run efficiently. .
- Feasibility study helps in making plans for training developers for implementing the system. .
- So a feasibility study is a report which could be used by the senior or top persons in the organization. This is because based on the report the organization decides about cost estimation, funding and other important decisions which is very essential for an organization to run profitably and for the system to run stable.

3.2 EXISTING SYSTEM

The existing system is full of manual process. Manual system maintains the limited number of transactions and storing the data's is very difficult. The existing system is found time consuming and complex procedure. It is difficult to get the details of all the particulars. Lots of paperwork is involved and human efforts are needed in the manual system. Record storage is irregular. This system is time-consuming and risky.

NEED FOR THE PROPOSED SYSTEM:

- User friendly
- To avoid paperwork
- Quick reference and understandable
- Processing time must be decreased.

3.3 PROPOSED SYSTEM

The main objective of the proposed system is to reduce time and make the system more user friendly, efficient, accurate and fast processing. It reduces the manual work and confusion occurring by maintaining files. The details of the transaction are added to the table automatically. The proposed system facilitates the admin to minimize the work load. More facilities can be added in future work.

ADVANTAGE OF PROPOSED SYSTEM

- More User friendly
- All information and requests will be computerized. No need for any special data backups.
- Provide High security
- Provides Uniqueness of facial characteristics
- Different options such as physiological and biological based authentication, which enhances security
- Avoid Compromising

SYSTEM DESIGN

4. SYSTEM DESIGN

Design is the first step in the development phase of any engineered system design can be defined as the transition from user's view to the view of programmers and database personnel. System design is a modeling process. It concentrates on transferring requirement specification to design specification. The inputs to the design are software requirements and the output will be the design specification applicable to all software design .This system design act as a bridge between the requirement specification and implementation phase.

As in the case of any systematic approach, this project too has undergone the best possible design phase fine – tuning all efficiency, performance and accuracy levels. This phase underwent preliminary and exhaustive design sub phases before implementation took over. The major steps in the phase are Input design, Output design.

4.1 INPUT DESIGN

Input design is one of the most important phases of the operation of computerized system and is often the major problem of a system. A large number of problems with a system can usually be tracked back to fault-input design and method .The decisions made during the input design are

- To achieve maximum accuracy, efficiency and reliability.
- To ensure the user-acceptable and user-understandable input format.
- To avoid errors made by the operators up to the maximum.

4.2 OUTPUT DESIGN

The readability and understandability of the output satisfies the customer. Therefore, the output should be such that the format is easy to understand and read for the customer. Output design is the most important part as it is the direct source for the censer efficiency and intelligence. The major form of the output is the hard copy, generated through the printers. The objectives to be fulfilled during the output design are:

- Output design should satisfy the needs of the information requirement.
- Select the appropriate methods for presenting the information.
- Create reports such that they are easily understandable and efficient.

For the end-users, the output is what they work for and so it must be useful for all the users. The output must satisfy all the needs of the users and the customer.

SYSTEM DEVELOPMENT

5. SYSTEM DEVELOPMENT

System development is a series of operations performed to manipulate data to produce output from a computer system. This is a highly dependent on the programming language used. The principal activities performed during the development phase can be divided into major related sequence. They are

- Internal
- Output

The major internal system development activities done for the system are computer program development and performance testing.

The major external system development activities are done by implementation, planning, equipment, acquisition and installation.

5.1 MODULE

This project consists of two modules.

- Face detection (and)
  - Face localization
  - Face normalization
  - Locating facial characteristics
  - Extracting facial features
- Face recognition

5.2 MODULE DESCRIPTION

5.2.1 FACE DETECTION

Face detection which is the primary process and face recognition which is an authenticating phase. Face detection involves four main concepts. Firstly, face localization which uses Hough method and skin color information method. Second, face normalization which extracts only the face by discarding all the surroundings. Third concept is, locating facial characteristics using neural network .Finally extracting the student face using Eigen face approach.

5.2.1 FACE RECOGNITION

After extracting the features of the face it will be basically stored as a template that will be used for recognition. In recognition phase same process of face detection is carried out and verifies whether the student is really authorized or not.

Methodology based on: Generation of Probability-Based convolution Face Mask Skin segmentation

Pixel-based face mask and probability.

Automatic skin detection is a common primitive for a range of human-related image processing applications, one of which includes face detection. Pixel-level skin searches can be very rapid and can greatly reduce the search space prior to higher level classification. Thus such detectors are often used as front-end primitives to higher level person and face location systems. The starting point for the following approach is assumed to be a colour image represented in the RGB colour space. The RGB space is known to be susceptible to intensity variations.

K – training positive images X1 of size M x N

Generating probability table p(x,y,GreyValue)

In most practical applications the RGB GreyValue space is quantised to 8 bits per colour, giving a discrete 3-D space of dimension 2^{24}. It is feasible to create a probability map covering all of these sixteen million-or-so pixel values, the main practical obstacle being the availability of a suitably large, labelled database. The distinct advantage of such a map is the computational speed with which a pixel can be tested: the discrete RGB values become a co-ordinate address in the probability map, resulting in an almost instantaneous pixel classification.

With the sample of 24 x 24 . i=l.....k;

Integrating face mask – Convolution Approach

D(x,y)sq/kmeans(xi(x,y)-M(x,y))2/k

Integrating face mask approach applies knowledge-based rules to intensity images, which are known for their sensitivity to changes in lighting conditions. This leads to non-faces being picked up in complex backgrounds that have similar intensity distributions to human faces. The approach adopted here, directly extends Integrating face mask work by utilising the additional information present in colour images. Skin segmentation is performed as a pre-processing step; only those image regions containing appropriate quantities of skin are considered for further processing.

SYSTEM TESTING & IMPLEMENTATION

6. SYSTEM TESTING AND IMPLEMENTATION

6.1 SYSTEM TESTING

TESTING METHODOLOGIES AND POLICIES

Testing is an activity to verify that a correct system is being built and is performed with the intent of finding faults in the system .However not restricted to being performed after the development phase is complete. But this is to carry out in parallel with all stages of system development, starting with requirement specification. Testing results once, gathered and evaluated, provide a qualitative indication of software quality and reliability and serve as a basis for design modification if required a project be set to be incomplete without proper testing.

System testing is a process of checking whether the development system is working according to the original objectives and requirements. The system should be tested experimentally with test data so as to ensure that the system works according to the required specification. When the system is found working, test it with actual data and check performance.

All tests should be traceable to customer requirements. The focus of testing will shift progressively from programs. Exhaustive testing is not possible. To be more effective, testing should be one, which has probability of finding errors.

The Following are the attributes of good test

A good test has a high probability of finding an error

A good test is not redundant.

A good test should be “Best of Breed”.

A good test should neither too simple nor too complex.

6.2 TYPES OF TESTING

The details of the software functionality tests are given below

The testing procedures that has been used is as follows

1. Unit Testing
2. Validation Testing
3. Output Testing
4. Performance Testing

6.1.1. UNIT TESTING

The first level of testing is called as unit testing. Here the different modules are tested and the specifications produced during design for the modules. Unit testing is essential for verification of the goal and to the test the internal logic of the modules. Unit test was conducted to the different modules of the project. Errors were noted down and corrected down immediately and the program clarity was increased the testing was carried out during the programming stage itself. In this step each module is found to be working satisfactory as regard to the expected output from the module. The modules like Customer Module, Employee Module, Stock Module etc., were tested. In this project Unit testing is performed for face detection and face recognition separately.

6.1.2. VALIDATION TESTING

The next level of testing is validation testing. Here the entire software is tested the reference document for this process is the requirement and the goal is to see if the software meets its requirements.

The requirement document reflects and determines whether the software function as the user expected. At the culmination of the integration testing, software is completely assembled as a package, interfacing and corrected and final serious of software test and validation test begins. The proposed system under construction has been tested by using validation testing and found to be working satisfactory. After finishing the integration testing, the modules were tested for validation.

6.1.3. OUTPUT TESTING

The output of the software should be acceptable to the system user. The output requirement is defined during the system analysis. Testing of the software system is done against the output requirements and the output testing was completed with success.

6.1.4. PERFORMANCE TESTING

This project is a system based project, and the modules are interdependent with the other modules, so the testing cannot be done module by module. So the unit testing is not possible in the case of this driver. So this system is checked only with their performance to check their quality.

In case of the unit testing the initialization module is first tested. Since read module and the write module is interdependent the performance testing is done.

6.3 IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

The stage consists of

- Testing the developed program with simple data.
- Detection's and correction of error.
- Creating whether the system meets user requirements.
- Testing whether the system.
- Making necessary changes as desired by the user.
- Training user personnel.

IMPLEMENTATION PROCEDURES

The implementation phase is less creative than system design. A system project may be dropped at any time prior to implementation, although it becomes more difficult when it goes to the design phase.

The final report to the implementation phase includes procedural flowcharts, record layouts, report layouts, and a workable plan for implementing the candidate system design into an operational one. Conversion is one aspect of implementation.

- The conversion portion of the implementation plan is finalized and approved.
- Files are converted.
- Parallel processing between the existing and the new system are logged on a special form.
- Assuming no problems, parallel processing is discontinued. Implementation results are documented for reference.

USER TRAINING

User Training is designed to prepare the user for testing & convening the system. There are several ways to train the user. They are

1) User Manual.

2) Help Screens.

3) Training Demonstration.

1. User Manual:

The summary of important functions about the system and software can be provided as a document to the user.

2. Help Screens:

This features now available in every software package, especially when it is used with a menu. The user selects the "Help" option from the menu. The system accesses the necessary description or information for user reference.

3. Training Demonstration:

Another User Training element is a Training Demonstration. Live demonstrations with personal contact are extremely effective for Training Users.

OPERATIONAL DOCUMENTATION

Documentation means of communication; it establishes design and performance criteria for phases of the project. Documentation is descriptive information that portrays the use and / or operation of the system.

Documentation Tools

Document production and desktop publishing tool support nearly ever aspect of software developers. Most software development organizations spend a substantial amount of time developing documents, and in many cases the documentation process itself is quite inefficient. It is not use unusual for a software development effort on documentation. For this reason, documentation tools provide an important opportunity to improve productivity.

Document Restructuring

Creating document is far too time consuming. If the system works, we'll live with what we have. In some cases, this is the correct approach. It is not possible to recreate document for hundreds of computer programs. Documentation must be updated, but we have limited resources. It may not be necessary to fully re-document an application. Rather, those portions of the system that are currently undergoing change are fully documented. The system is business critical and must be fully re-documented. Even in this case, an intelligent approach is to pare documentation to an essential minimum.

SYSTEM MAINTANANCE

Maintenance is actually the implementation of the review plan. As important as it is, many programmers and analysts are to perform or identify themselves with the maintenance effort. There are psychological, personality and professional reasons for this. Analysts and programmers spend far more time maintaining programs than they do writing them. Maintenance accounts for 50-80 percent of total system development.

Maintenance is expensive. One way to reduce the maintenance costs are through maintenance management and software modification audits.

- Maintenance is not as rewarding as exciting as developing systems. It is perceived as requiring neither skill not experience.
- Users are not fully cognizant of the maintenance problem or its high cost.
- Few tools and techniques are available for maintenance.
- A good test plan is lacking.
- Standards, procedures, and guidelines are poorly defined and enforced.
- Programs are often maintained without care for structure and documentation.
- There are minimal standards for maintenance.
- Programmers expect that they will not be in their current commitment by time their programs go into the maintenance cycle.

CONCLUSION

7. CONCLUSION

The existing system is full of manual process. Manual system maintains the limited number of transactions and storing the data's is very difficult. The existing system is found time consuming and complex procedure. It is difficult to get the details of all the particulars. Face recognition based biometric system overcomes the difficulties and provides more security.

The currently developed multi face recognition system has its high scope compared to single face recognition and it is found to be working accurately. It is tested for its effectives, flexibility, accuracy and user friendly. The system is found to be delightful running under the single window system.

The programming techniques used in the design of the system provides a scope for further expansion and implementation of any changes which may occur in future. The system has been tested with all sample data covering all possible options for each function. Its performance is satisfactory the system is under implementation.

Documentations are done in such a way, that any other software developer can easily undergo this system, before undergoing for development for any changes that is to be single window system to the present system.

SCOPE AND FUTURE ENHANCEMENT

8. SCOPE FOR FUTURE ENHANCEMENT

The system has been designed and developed according to the current requirements of the user. At the same time the system is very flexible and extensible, Hence, future enhancements, if needed can be made without much difficulty, so new applications can be developed and it be integrated with the existing one very easily.

The following future enhancements may be worthwhile to make the tool usable to a wider section of users.

- Multi biometric authentication such as veins, hand, fingerprint, iris, voice, signature and keystroke based authentication can be implemented simultaneously for better security
- The user can enhance the software by using the tabs like banners, images, movie files, by the multicolored frames.
- The software here produced provides an alternate security that is , description about the web explorer

BIBLIOGRAPHY

9. BIBILIOGRAPHY

Good Teachers are worth more than thousand books, we have them in Our Department

References Made From

1. A. Gunjan Dashore, B. Dr. V.Cyril Raj, 2012 , An Efficient Method For Face Recognition Using Principal Component Analysis (Pca), Volume 2, Issue 2, (IJATER) pp:23-28.

2. Rabia Jafri and Hamid R Arabnia, 2009,"A Survey of Face Recognition Techniques," Journal of Information Processing Systems, vol. 5, no. 2, pp. 41~68,.

3. OLIVER, N., PENTLAND, A., AND BERARD, F. 1997. Lafter: Lips and face real time tracker. InProc. Computer Vision and Pattern Recognition, 123–129.

4. PEER, P., KOVAC, J., AND SOLINA, F. 2003. Human skin colour clustering for face detection.In submitted to EUROCON 2003 – International Conference on Computer as a Tool.

5. PHUNG, S. L., BOUZERDOUM, A., AND CHAI, D. 2002. A novel skin color model in ycbcr color space and its application to human face detection. In IEEE International Conference on Image Processing (ICIP'2002), vol. 1, 289–292.

Books Referred:

1. More VB.NET (Teach Yourself), Lowell Mauer, Sams Publication,2001.
2. Guide to VB.NET ,Peter Norton, Sams Publication,1998.
3. Fundamentals of Database System, Ramez , Addison-Wesley,1999
4. Complete Guide to SQL server, Peter Norton, Sams Publication,1997.

Web links referred

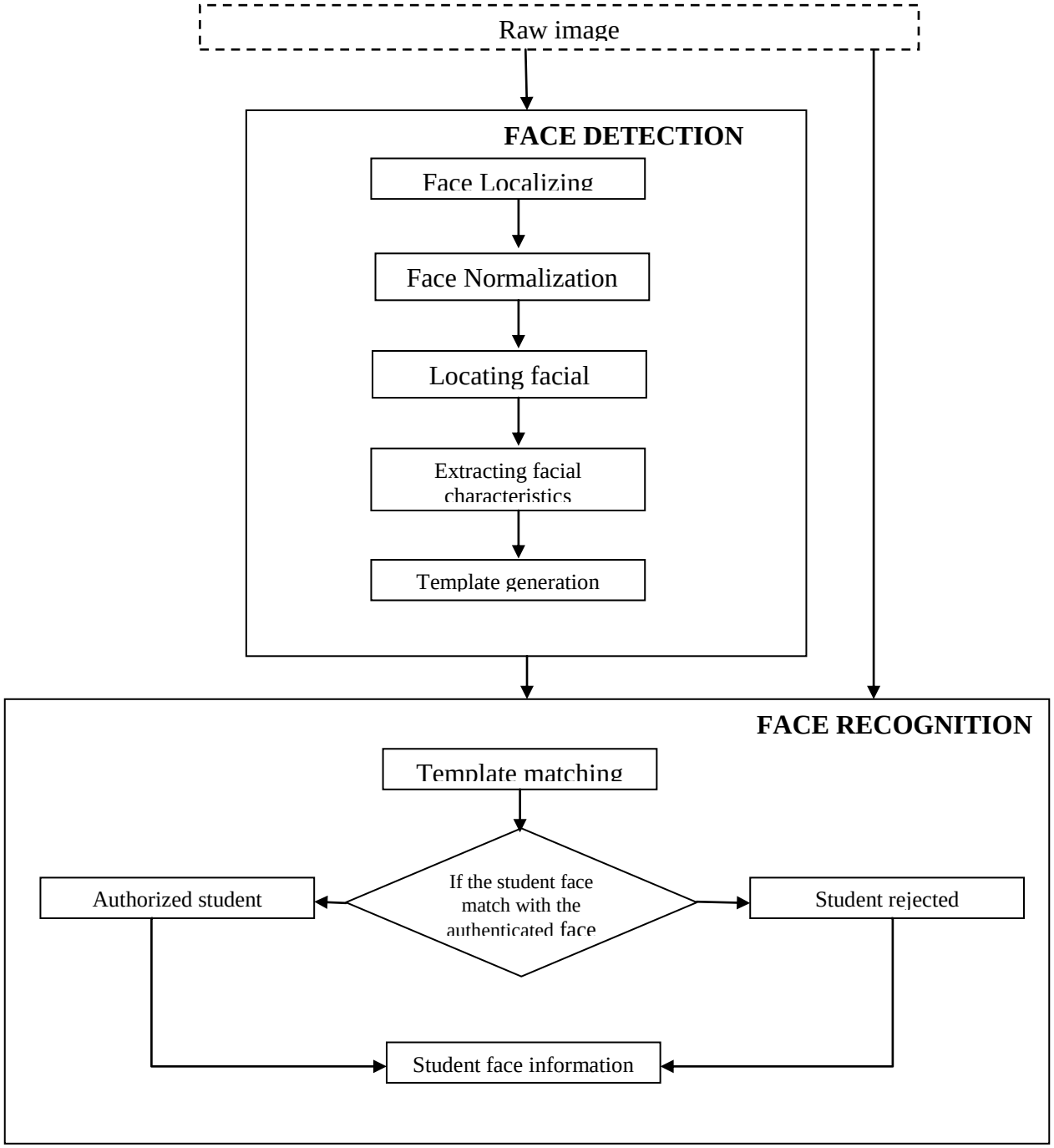
1. Http://www.Sourcecode.com
2. Http://www.dbms.co.in
3. Http://A1code.com
4. www.dodcounterdrug.com/facialrecognition

APPENDIX

10. APPEDIX

10.1 DATA FLOW DIAGRAM

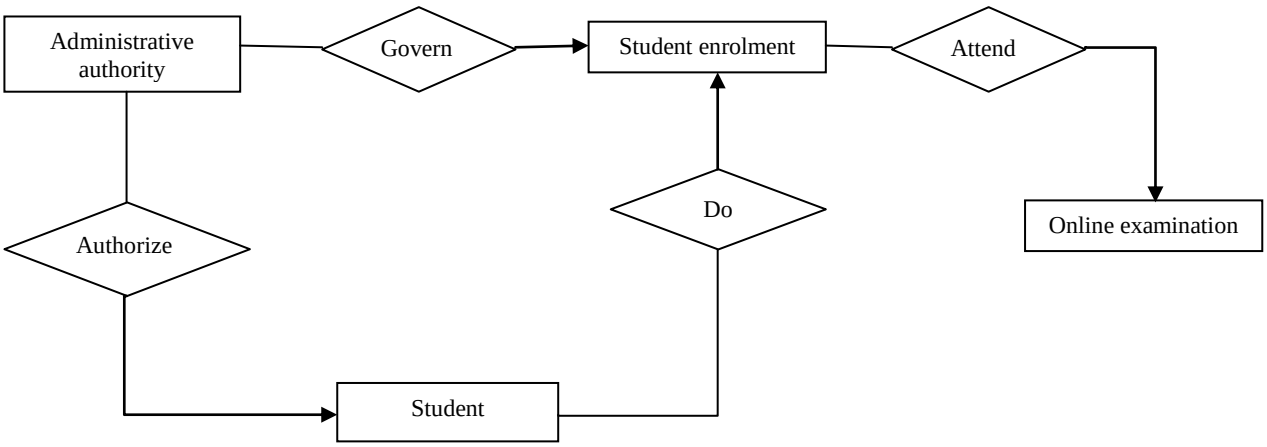
The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.



a) Data Flow Diagram

ENTITY RELATIONSHIP DIAGRAM

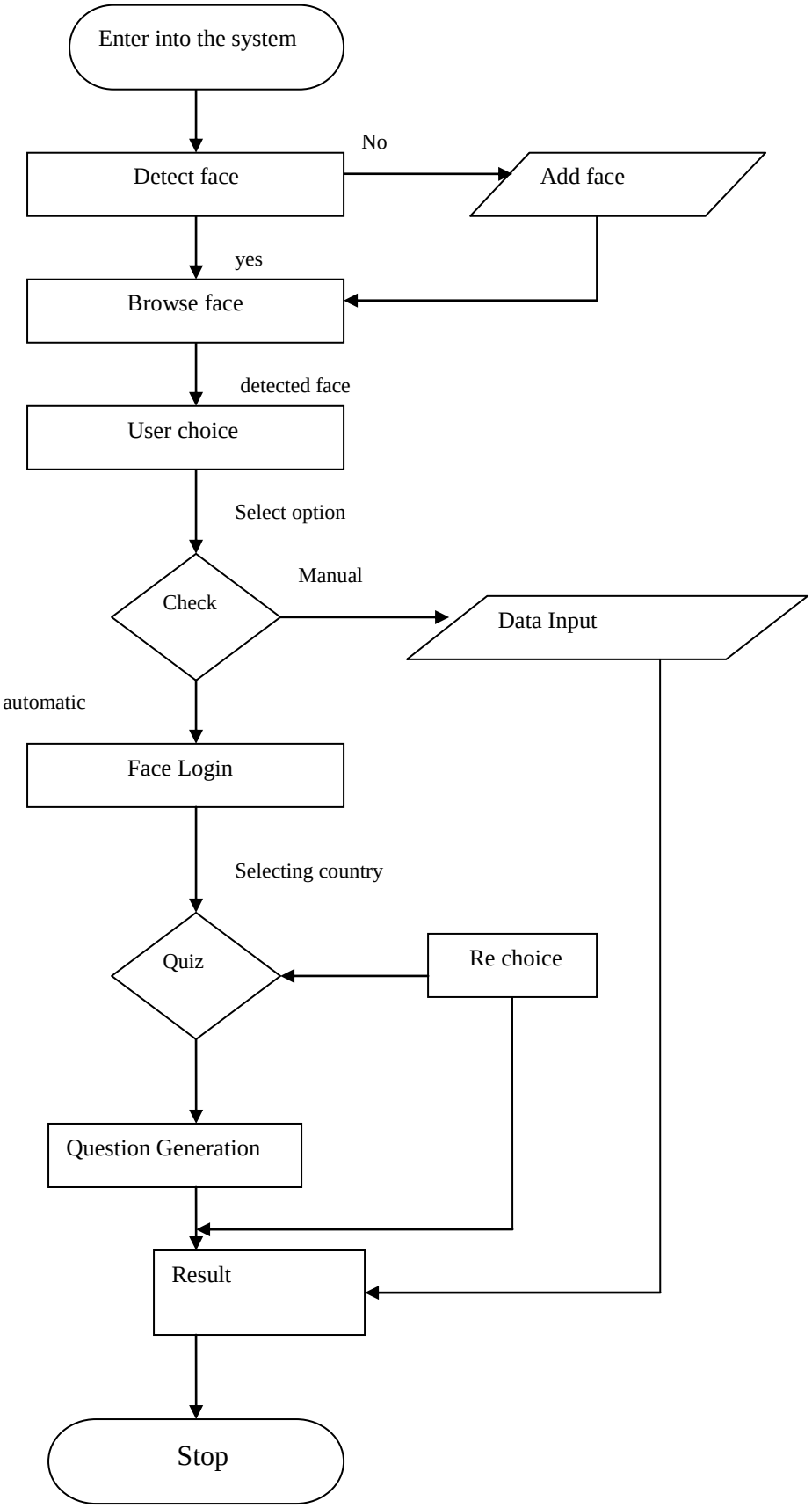
10.2 ENTITY RELATIONSHIP DIAGRAM



b) Entity Relationship Diagram

SYSTEM FLOW DIAGRAM

10.3 SYSTEM FLOW DIAGRAM



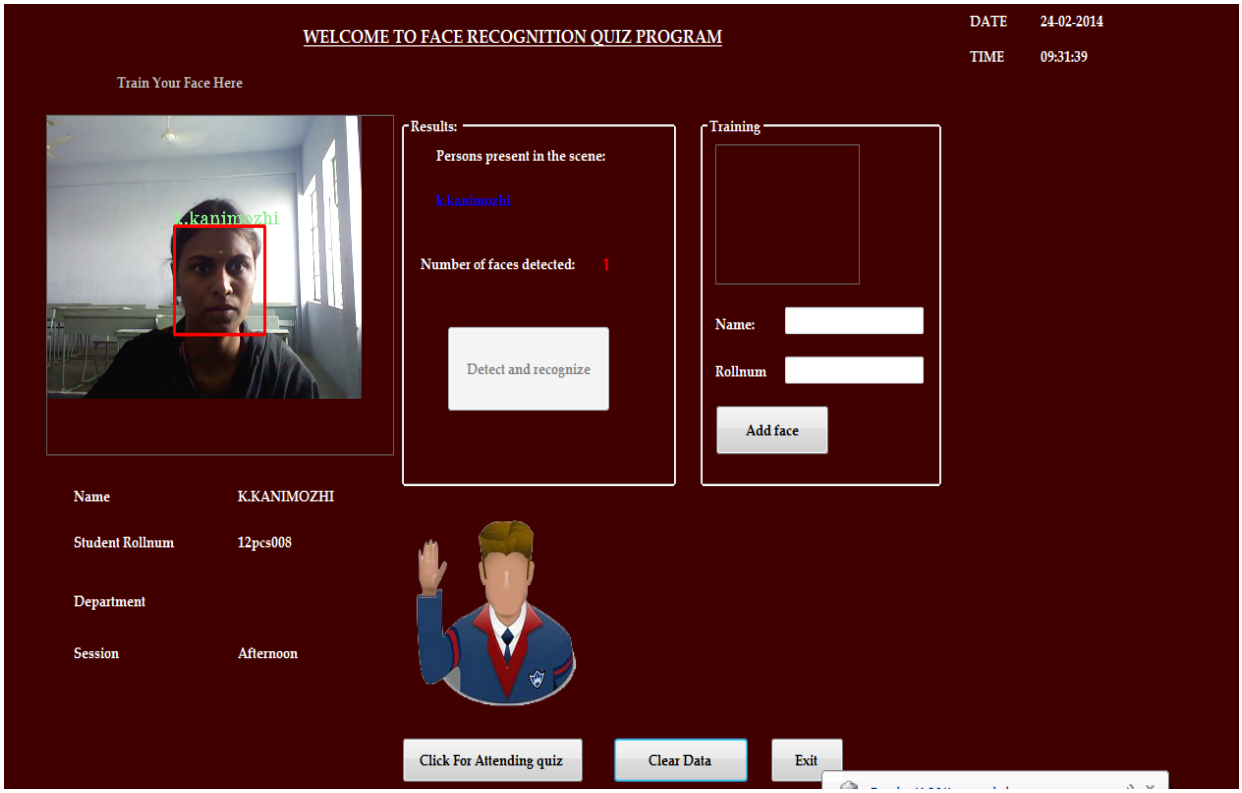
SCREEN SHOTS

10.4 SCREEN SHOTS

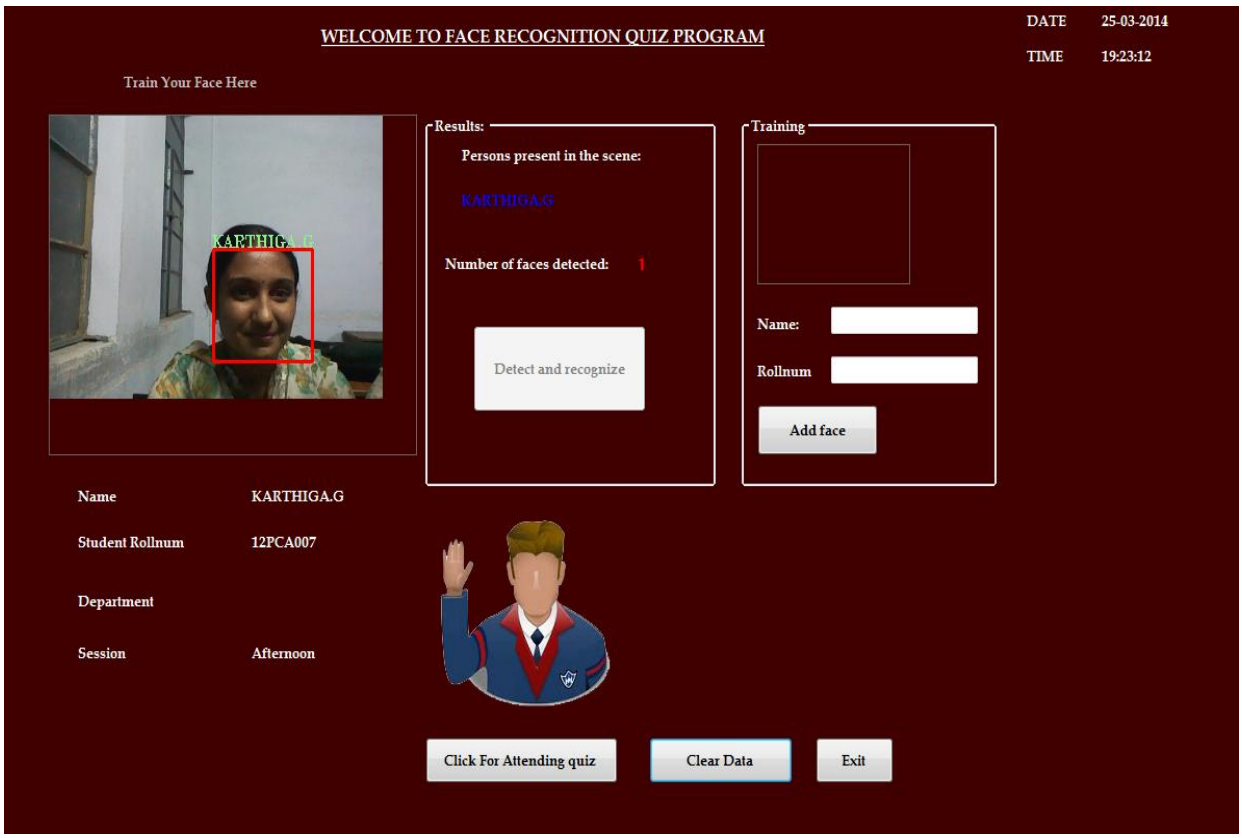
10.4.1. ONLINE EXAMINATION WELCOME PAGE

10.4.2 ONLINE EXAMINATION HOME PAGE

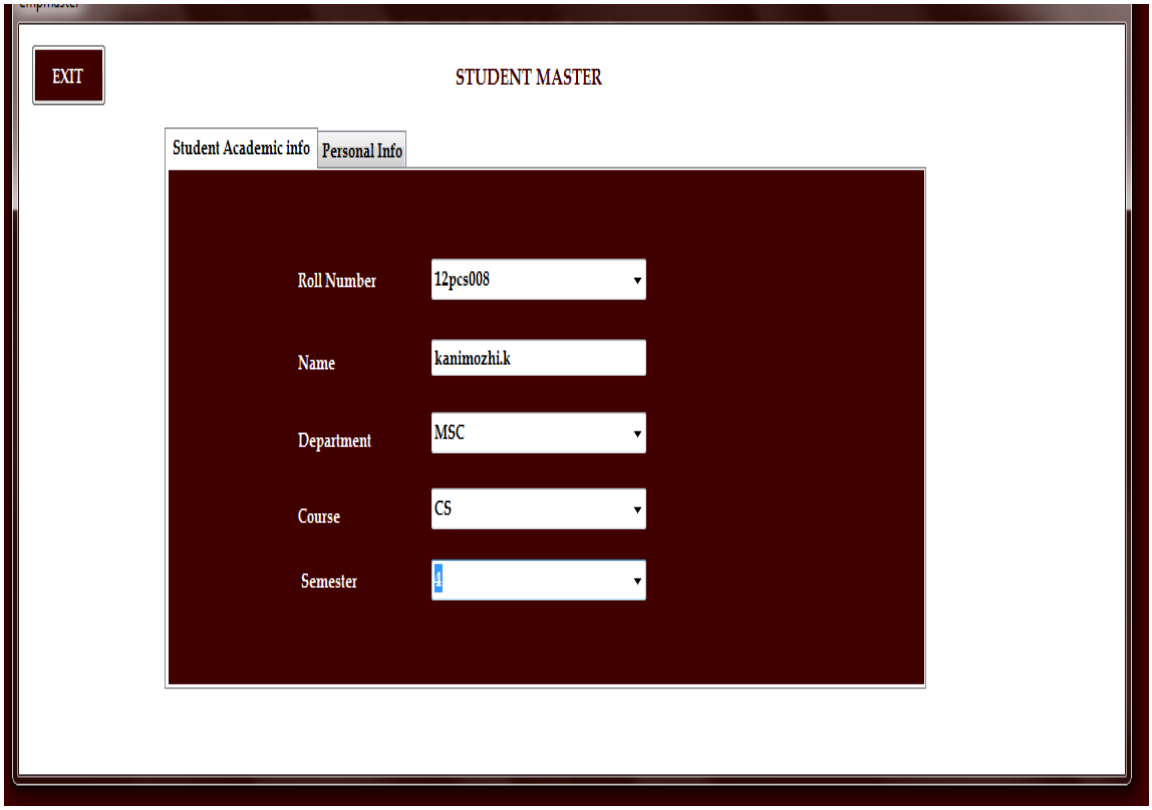
10.4.3.TRANING A NEW FACE



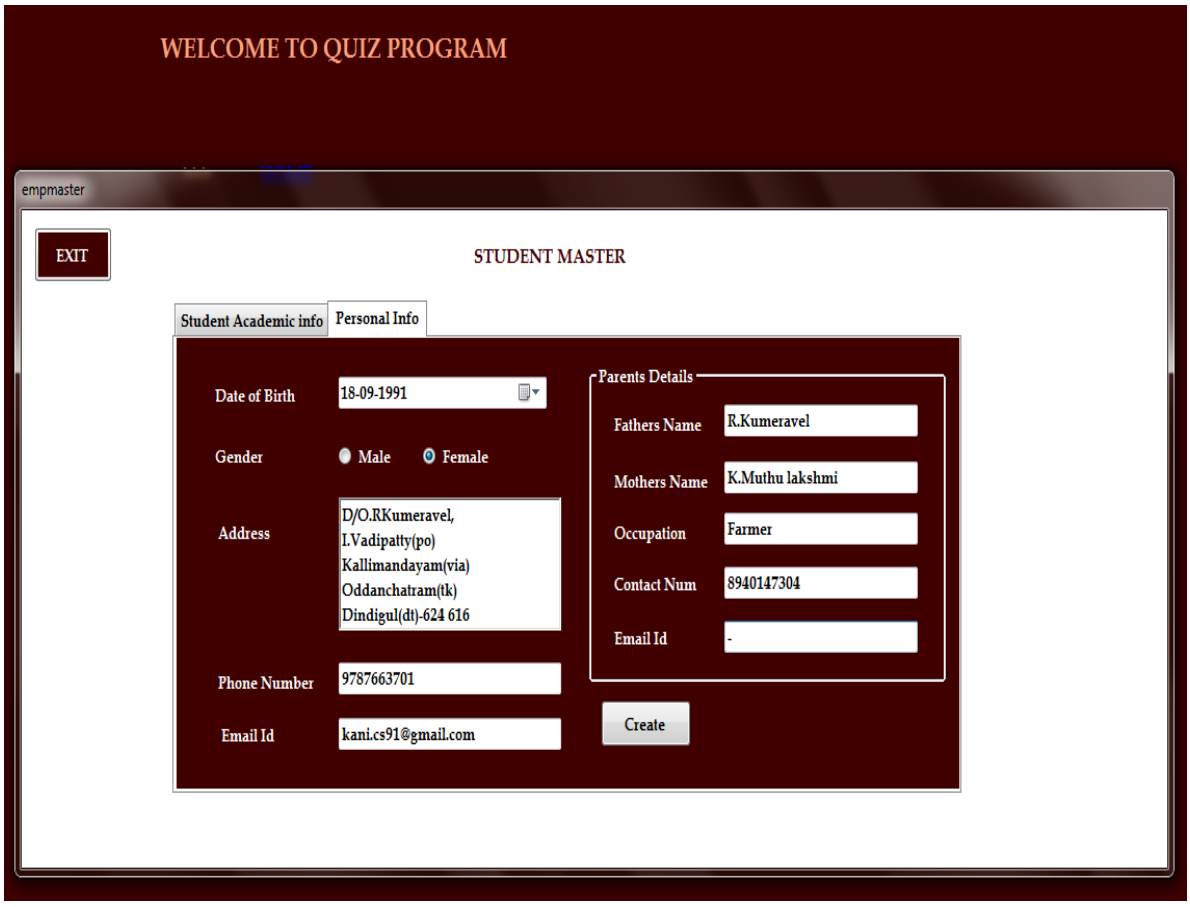
10.4.4 ATTENDING THE TEST



10.4.5 STUDENT ENROLLMENT(ACADEMIC INFORMATION)



10.4.6 STUDENT ENROLLMENT(PERSONAL INFORMATION)



SOURCE CODE

10.5. SOURCE CODE

```
//Multiple face detection and recognition in real time
//Using EmguCV cross platform .Net wrapper to the Intel OpenCV image processing library for C#.Net
//Writed by Sergio Andrés Guitérrez Rojas
//"Serg3ant" for the delveloper comunity
// Sergiogut1805@hotmail.com
//Regards from Bucaramanga-Colombia ;)
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
using Emgu.CV;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;
using System.IO;
using System.Diagnostics;
namespace MultiFaceRec
{
   public partial class FrmPrincipal : Form
   {
      //Declararation of all variables, vectors and haarcascades
      Image<Bgr, Byte> currentFrame;
      Capture grabber;
      HaarCascade face;
      HaarCascade eye;
      MCvFont font = new MCvFont(FONT.CV_FONT_HERSHEY_TRIPLEX, 0.5d, 0.5d);
      Image<Gray, byte> result, TrainedFace = null;
      Image<Gray, byte> gray = null;
      List<Image<Gray, byte>> trainingImages = new List<Image<Gray, byte>>();
      List<string> labels= new List<string>();
      List<string> NamePersons = new List<string>();
      int ContTrain, NumLabels, t;
      string name, names = null;
      public FrmPrincipal()
      {
         InitializeComponent();
         //Load haarcascades for face detection
         face = new HaarCascade("haarcascade_frontalface_default.xml");
         //eye = new HaarCascade("haarcascade_eye.xml");
```

```
            try
            {
                //Load of previus trainned faces and labels for each image
                string Labelsinfo = File.ReadAllText(Application.StartupPath +
"/TrainedFaces/TrainedLabels.txt");
                string[] Labels = Labelsinfo.Split('%');
                NumLabels = Convert.ToInt16(Labels[0]);
                ContTrain = NumLabels;
                string LoadFaces;
                for (int tf = 1; tf < NumLabels+1; tf++)
                {
                    LoadFaces = "face" + tf + ".bmp";
                    trainingImages.Add(new Image<Gray, byte>(Application.StartupPath + "/TrainedFaces/"
+ LoadFaces));
                    labels.Add(Labels[tf]);
                }
            }
            catch(Exception e)
            {
                //MessageBox.Show(e.ToString());
                MessageBox.Show("No Face available. please add at least a face.", "Trained faces load",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
        }
        private void button1_Click(object sender, EventArgs e)
        {
            //Initialize the capture device
            grabber = new Capture();
            grabber.QueryFrame();
            //Initialize the FrameGraber event
            Application.Idle += new EventHandler(FrameGrabber);
            button1.Enabled = false;
        }
        private void button2_Click(object sender, System.EventArgs e)
        {
            try
            {
                //Trained face counter
                ContTrain = ContTrain + 1;
               //Get a gray frame from capture device
```

```
            gray = grabber.QueryGrayFrame().Resize(320, 240,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
            //Face Detector
            MCvAvgComp[][] facesDetected = gray.DetectHaarCascade(
            face,
            1.2,
            10,
            Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING,
            new Size(20, 20));
            //Action for each element detected
            foreach (MCvAvgComp f in facesDetected[0])
            {
               TrainedFace = currentFrame.Copy(f.rect).Convert<Gray, byte>();
               break;
            }
            //resize face detected image for force to compare the same size with the
            //test image with cubic interpolation type method
            TrainedFace = result.Resize(100, 100, Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
            trainingImages.Add(TrainedFace);
            labels.Add(textBox1.Text);
            //Show face added in gray scale
            imageBox1.Image = TrainedFace;
            //Write the number of triained faces in a file text for further load
            File.WriteAllText(Application.StartupPath + "/TrainedFaces/TrainedLabels.txt",
trainingImages.ToArray().Length.ToString() + "%");
            //Write the labels of triained faces in a file text for further load
            for (int i = 1; i < trainingImages.ToArray().Length + 1; i++)
            {
               trainingImages.ToArray()[i - 1].Save(Application.StartupPath + "/TrainedFaces/face" + i +
".bmp");
               File.AppendAllText(Application.StartupPath + "/TrainedFaces/TrainedLabels.txt",
labels.ToArray()[i - 1] + "%");
            }
            MessageBox.Show(textBox1.Text + "´s face detected and added :)", "Training OK",
MessageBoxButtons.OK, MessageBoxIcon.Information);
         }
         catch
         {
            MessageBox.Show("Enable the face detection first", "Training Fail",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
         }
```

```
        }
        void FrameGrabber(object sender, EventArgs e)
        {
            label3.Text = "0";
            //label4.Text = "";
            NamePersons.Add("");
            //Get the current frame form capture device
            currentFrame = grabber.QueryFrame().Resize(320, 240,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
                //Convert it to Grayscale
                gray = currentFrame.Convert<Gray, Byte>();
                //Face Detector
                MCvAvgComp[][] facesDetected = gray.DetectHaarCascade(
               face,
               1.2,
               10,
               Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING,
               new Size(20, 20));
                //Action for each element detected
                foreach (MCvAvgComp f in facesDetected[0])
                {
                    t = t + 1;
                    result = currentFrame.Copy(f.rect).Convert<Gray, byte>().Resize(100, 100,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
                    //draw the face detected in the 0th (gray) channel with blue color
                    currentFrame.Draw(f.rect, new Bgr(Color.Red), 2);
                    if (trainingImages.ToArray().Length != 0)
                    {
                        //TermCriteria for face recognition with numbers of trained images like maxIteration
                    MCvTermCriteria termCrit = new MCvTermCriteria(ContTrain, 0.001);
                    //Eigen face recognizer
                    EigenObjectRecognizer recognizer = new EigenObjectRecognizer(
                      trainingImages.ToArray(),
                      labels.ToArray(),
                      3000,
                      ref termCrit);
                    name = recognizer.Recognize(result);
                        //Draw the label for each face detected and recognized
                    currentFrame.Draw(name, ref font, new Point(f.rect.X - 2, f.rect.Y - 2), new
Bgr(Color.LightGreen));
                    }
```

```
                NamePersons[t-1] = name;
                NamePersons.Add("");
            //Set the number of faces detected on the scene
            label3.Text = facesDetected[0].Length.ToString();
            /*
            //Set the region of interest on the faces
             gray.ROI = f.rect;
            MCvAvgComp[][] eyesDetected = gray.DetectHaarCascade(
              eye,
              1.1,
              10,
              Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING,
              new Size(20, 20));
            gray.ROI = Rectangle.Empty;
            foreach (MCvAvgComp ey in eyesDetected[0])
            {
                Rectangle eyeRect = ey.rect;
                eyeRect.Offset(f.rect.X, f.rect.Y);
                currentFrame.Draw(eyeRect, new Bgr(Color.Blue), 2);
            }
           */
        }
            t = 0;
            //Names concatenation of persons recognized
        for (int nnn = 0; nnn < facesDetected[0].Length; nnn++)
        {
            names = names + NamePersons[nnn] + ", ";
        }
        //Show the faces procesed and recognized
        imageBoxFrameGrabber.Image = currentFrame;
        label4.Text = names;
        names = "";
        //Clear the list(vector) of names
        NamePersons.Clear();
    }
private void button3_Click(object sender, EventArgs e)
{
        }
private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
   photo ph = new photo();
```

```
            ph.Show();
        }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace MultiFaceRec
{
    public partial class photo : Form
    {
        int height, width, h, w;
        string filepath;
        string fname;
        public photo()
        {
            InitializeComponent();
            Bitmap b;
                }
        private void button1_Click(object sender, EventArgs e)
        {
            openFileDialog1.Title = "Get Image";
            openFileDialog1.ShowDialog();

        }
        private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
        {
            textBox1.Text = openFileDialog1.FileName;
        }
        private void button2_Click(object sender, EventArgs e)
        {
            if (radioButton1.Checked == true)
            {
                height = 170;
                width = 132;
```

```
        }
        if (radioButton2.Checked == true)
        {
            height = 192;
            width = 192;
        }
        if (radioButton3.Checked == true)
        {
            height = 181;
            width = 125;
        }
        if (radioButton4.Checked == true)
        {
            height = 170;
            width = 136;
        }
        if (radioButton5.Checked == true)
        {
            height = 144;
            width = 181;
        }
        if ((radioButton1.Checked == false) && (radioButton2.Checked == false) &&
(radioButton3.Checked == false) && (radioButton4.Checked == false) && (radioButton5.Checked ==
false))
        {
            MessageBox.Show("Select Country");
        }
        else
        {
            fname = openFileDialog1.FileName;
             fname = fname.Substring(fname.LastIndexOf("\\"));
             fname = fname.Substring(1);
             fname = fname.Remove(fname.IndexOf("."), 4);
            if (textBox1.Text.Trim() != " ")
            {
                Image image = Image.FromFile(textBox1.Text);
                Bitmap result = new Bitmap(width, height);
                using (Graphics graphics = Graphics.FromImage(result))
                {
```

```
                graphics.CompositingQuality =
System.Drawing.Drawing2D.CompositingQuality.HighQuality;
                graphics.InterpolationMode =
System.Drawing.Drawing2D.InterpolationMode.HighQualityBicubic;
                graphics.SmoothingMode =
System.Drawing.Drawing2D.SmoothingMode.HighQuality;
                graphics.DrawImage(image, 0, 0, result.Width, result.Height);
            }
            pictureBox1.Image = result;
            filepath = Application.StartupPath;
            if (radioButton1.Checked == true)
            {
              pictureBox1.Image.Save(filepath + "/india/" + fname + ".jpg" , ImageFormat.Jpeg);
            }
            if (radioButton2.Checked == true)
            {
                pictureBox1.Image.Save(filepath + "/US/" + fname + ".jpg", ImageFormat.Jpeg);
            }
            if (radioButton3.Checked == true)
            {
                pictureBox1.Image.Save(filepath + "/china/" + fname + ".jpg", ImageFormat.Jpeg);
            }
            if (radioButton4.Checked == true)
            {
                pictureBox1.Image.Save(filepath + "/german/" + fname + ".jpg", ImageFormat.Jpeg);
            }
            if (radioButton5.Checked == true)
            {
                pictureBox1.Image.Save(filepath + "/aus/" + fname + ".jpg", ImageFormat.Jpeg);
            }
        }
          else
          {
            MessageBox.Show("Select Photo");
          }
        }
    }
    private void button3_Click(object sender, EventArgs e)
    {
       this.Close();
    }
```

```
        private void photo_Load(object sender, EventArgs e)
        {
        }
        private void button4_Click_1(object sender, EventArgs e)
        {
            if (textBox1.Text == "")
            {
                MessageBox.Show("Browse Photo");
            }
            else
            {
                h = Convert.ToInt32(txtheight.Text);
                w = Convert.ToInt32(txtwidth.Text);
                fname = openFileDialog1.FileName;
                fname = fname.Substring(fname.LastIndexOf("\\"));
                fname = fname.Substring(1);
                fname = fname.Remove(fname.IndexOf("."), 4);
                if (textBox1.Text.Trim() != " ")
                {
                    Image image = Image.FromFile(textBox1.Text);
                    Bitmap result = new Bitmap(w, h);
                    using (Graphics graphics = Graphics.FromImage(result))
                    {
                        graphics.CompositingQuality =
System.Drawing.Drawing2D.CompositingQuality.HighQuality;
                        graphics.InterpolationMode =
System.Drawing.Drawing2D.InterpolationMode.HighQualityBicubic;
                        graphics.SmoothingMode =
System.Drawing.Drawing2D.SmoothingMode.HighQuality;
                        graphics.DrawImage(image, 0, 0, result.Width, result.Height);
                    }
                    pictureBox1.Image = result;
                    filepath = Application.StartupPath;
                    pictureBox1.Image.Save(filepath + "/manual/" + fname + ".jpg", ImageFormat.Jpeg);
                }
            }
```