DATA CAPTURING & PRE-PROCESSING OF SIGNATURES FOR OFFLINE VERIFICATION

Project work done at

System Logic Solutions Ltd., Bangalore

PROJECT REPORT P-785

Submitted in partial fulfillment of the requirements

for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Of Bharathiar University, Coimbatore.

Submitted by

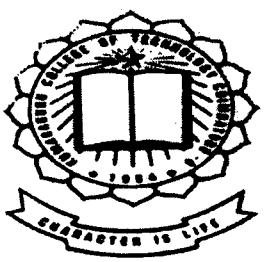
Prabhu Chakravarthy.J.

9938M0623

GUIDED BY

Mr.Mahesh Jagirdar ME.,
EXTERNAL GUIDE

Ms.V.Geetha M.C.A.,
INTERNAL GUIDE



Department of Computer Science & Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore – 641 006

May 2002

Certificate

Department of Computer Science & Engineering

Kumaraguru College of Technology

(Affiliated to Bharathiar University)

Coimbatore – 641 006

CERTIFICATE

This is to certify that the project work entitled

DATA CAPTURING & PRE-PROCESSING OF SIGNATURES FOR OFFLINE VERIFICATION

Done by

Prabhu Chakravarthy.J.

9938M0623

Submitted in partial fulfillment of the requirements

for the award of the degree of

Master of Computer Applications of Bharathiar University.

S.Thangasamy

Dr.S.Thangasamy

Professor & Head

Ms.V.Geetha

Internal Guide

Submitted to University Examination held on 10.05.2002

10/5/02

Internal Examiner

External Examiner

temLogic
ons that work

SystemLogic Solutions Ltd.,
1281, 21st Main,
2nd Phase, J P Nagar,
BANGALORE - 560 078.

Phone : +91-80-659 3005, 659 6842
Fax : +91-80-659 6653
Web : www.systemlogicindia.com
E-mail : inquiry@systemlogicindia.com

Mar 25, 2002

TO WHOMSOEVER IT MAY CONCERN

This is to certify that ***Mr. Prabhu Chakravarthy, MCA*** *(VI Semester) Student of* ***Kumaraguru College of Technology, Coimbatore,*** *has successfully completed a Project at our concern, as per the details given below.*

Project Title: ***Forgery Detection & Analysis of Offline Signatures.***
Project Period: ***December 2001 to March 2002.***
Technologies: ***C++ 3.0, OTL 4.0, MS-Access.***

This Project has been successfully implemented, and has met our requirements. He has completed the following modules – Data Capturing & Preprocessing of Signatures for Offline verification. His conduct was found excellent through out the project period. I also rate him to be technically very good and innovative in his approach to solving problems.

I wish him all success in his future endeavors.

With Best Regards,

For SystemLogic Solutions Ltd.

Vivekananda Hallur
Vice President

Declaration

DECLARATION

I hereby declare that the project entitled ***'Data Capturing & Pre-Processing of Signatures for Offline Verification'****, submitted to Bharathiar University as the project work of Master of Computer Applications Degree, is a record of original work done by me under the supervision and guidance of* ***Mr.Mahesh Jagirdar ME., Project Manager, System Logic India Pvt Ltd., Bangalore*** *and* ***Ms.V.Geetha M.C.A., Senior Lecturer, Kumaraguru College of Technology, Coimbatore*** *and this project work has not found the basis for the award of any Degree / Diploma / Associationship / Fellowship or similar title to any candidate of any university.*

Station: *Coimbatore*

Date: 26/04/2002

Signature of the student,

Prabhu Chakravarthy.J

Prabhu Chakravarthy.J.

Acknowledgement

Table of Contents

Acknowledgements

I wish to express my deep sense of gratitude to Dr.K.K.Padmanabhan B.Sc.,(Engg), M.Tech.,Ph.D., Principal, Kumaraguru College of Technology, Coimbatore and Dr.S.Thangaswamy BE.,(Hons)., Ph.D., Professor, Head of Department, Computer Science & Engineering, Kumaraguru College of Technology, Coimbatore, for providing me with an opportunity to take up this project.

I would like to express my sincere regards and a big word of thanks to Ms.V.Geetha M.C.A., Senior Lecturer, Department of Computer Science & Engineering, Kumaraguru College of Technology, Coimbatore for her technical support, encouragement, consistent guidance and inspiration all through my association with her.

I am very much thankful to Mr.Mahesh Jagirdar ME., Project Manager, System Logic India Pvt Ltd., Bangalore, for his constant support, guidance, encouragement and help in learning many new things during my project period.

With profound sense of gratitude, I acknowledge the guidance and support extended by Ms.N.Chandrika M.C.A., Mr.R.Raghavendra BE., SystemLogic India Pvt Ltd., Bangalore, for their technical assistance, incessant encouragement and invaluable technical support have been of immense help in realizing this project. I am also thankful to all the members of our team, SystemLogic India Pvt Ltd., Bangalore. Their timely suggestions have greatly contributed in successful completion of this project.

I take a great pleasure in thanking all the faculty members of Department of Computer Science & Engineering, for their co-operation and encouragement during my study period.

It was my parents and friends who really stood behind me giving all the support for my work.

Prabhu Chakravarthy.J.

Synopsis

Synopsis

This signature verification system intends to identify skilled forgeries met with offline signatures. It was designed using Unified Modeling Language with Rational Software Corporation's, Rational Rose 2000 and was programmed using Turbo C++ v3.0 for System Logic India Pvt Ltd., Bangalore.

Signature is a proof of authorship of any individual and therefore is a content of value when represented over any legal document. Hence signature verification is now an active research area that aims to identify whether a signature is genuine or forged.

The thesis introduces an effective approach for capturing the signature data and pre processing the same with respect to the offline signature verification system. Specimen signatures, which are true samples of author and the questioned signature, which is suspected to be forged, are fed into the system in the windows bitmap file format. These bitmap files are fetched and read by the system from which the signature contents are extracted by utilizing proper clipping methodologies and are zoomed to predefined resolutions. These processed images can be used for feature extractions and their authenticity could be verified.

The system also achieves a database connectivity upon a user login and discriminates the user levels into three types namely the administrator, privileged user and as the general user. The access rights of these users are limited at each user level, since the system is a security-based system.

Contents

Introduction

1. Introduction

1.1 About the Organization

SystemLogic is a software solutions company, which was established in the year 1994 in US and later spread its wings to India in 1996.

The company offers comprehensive enterprise-wide and e-business solutions to a wide range of clients across the world.

SystemLogic, today, boasts of a client base of topnotch Indian companies and multinationals based in India and overseas with employee strength of 200 and a turnover of USD 7.0 million.

1.2 Problem Description

Signature is the way a person writes his name. It may be stylish or unconventional and have many personal characteristics that are challenging to reproduce by anyone other than the original author. For this reason, signatures are used and accepted as proof of authorship or consent on personal checks, credit purchases and legal documents.

Signature verification is an extremely active research area. Although systems exist on the market, there are few that can promise sufficiently high accuracy rates at a reasonable level of efficiency.

Currently signatures are verified informally in many environments, but the rapid development of computer technology has stimulated great interest in research on automated signature verification and forgery detection.

There are three types of forgeries namely Simple, Substitution and Freehand or Skilled. In this project, we focus on detection of skilled forgeries of offline signatures.

1.3 Platform

The developed project is coded using Turbo C++ thus capturing the object orientation features of the language and can be a very useful tool in investigations of alleged authentication forgeries, as a part of the security system in the banking sector, or in the matters involving any legal documents.

Problem Definition & Feasibility Analysis

2. Problem Definition & Feasibility Study

The idea of building a software intensive system is envisaged at this stage, which involves the understanding of the problem and the rationale for solving it. This stage in the Software Development Life Cycle (SDLC) serves as a proof of understanding the user requirement.

This section details the primary requirements of the new system based on the limitations over the existing system and the need to switch over to a new system.

Feasibility study tends to measure the factor of possibility in building the system. This could be performed either from the developer's side or from the client's side.

2.1 Problem Definition

The primary requirement for an efficient standalone system, for verifying the signatures of various individuals in order to authenticate their genuineness.

In addition to this, the System must have proper access restrictions to prevent unauthorized users from tampering with the sensitive data. Periodic report generation is also needed.

2.2 Existing System-Limitation

For authenticating a signature there are three techniques available namely,

- Manual verification,
- Offline verification, &
- Online verification.

The existing system is a manual verification system where comparison is based on clearly distinguishable features. This technique mainly depends on the person's power of observation. The developed system is an offline verification system whereby the computerized comparison technique, which takes into account the less prominent, but significant, features of a signature for comparison purposes.

It is assumed that there are certain characteristics in a person's signature, which are difficult to reproduce. There is a chance that these features might be overlooked when performing the verifications manually. Moreover, clearly distinguishable features are easy to forge, and a skilled forger can easily reproduce a signature with proper practice. So, the percentage of accuracy reduces in a manual verification system.

Hence the need for a automated system arises. Still the factors of an online system, like speed of signing, hand pressure etc., cannot be utilized in an offline system. The existing technique followed is a manual verification system.

An offline system unlike the online system gives a more powerful thought in detecting forgeries. For an instance if a person's signature has to be verified in his absence, or signatures from old documents, of people who are no longer existent needs a offline system to undergo signature verification. This accounts for the significance of building up an offline forgery detection system.

2.3 Client Requirements

- An efficient stand alone offline signature verification system to verify the genuineness of the signatures.

- This being a security based system needs a proper authentication module for restricting the user entry.

- The system should analyze signatures, arrive at proper conclusions and justify the same using proper images & text. No batch processing is required.

- Ad hoc reports generation provision should be included.

- Provision for further enhancements to promote it into an online system & support more image formats.

2.4 Proposed System

The developed system is a standalone "Offline Signature Verification" system that attempts to authenticate the genuineness of a questioned signature by comparing it with various copies of the original signature.

Signature acquisition in an offline system is carried out long after the writing process actually occurs. Since the processing is done on static data these systems are often called as static systems.

The system produces justified conclusions about the genuineness of a signature after processing the specimens & the questioned samples.

2.5 System Inputs & Outputs

All the inputs to the system are in the form of bitmap images (.BMP) .These are then processed further by clipping the images and zooming them to a predefined resolution. The process is detailed below.

2.5.1 Inputs

The main input for the system will be in the form of a scanned bitmap (. bmp) file containing the signature image. There can be n number of reference signatures, which are the original signatures, and a test signature, which is the questioned signature. In case the given reference signatures fail to arrive at conclusions, the system may ask the end user for more specimens.

Since the system handles sensitive data, unauthorized usage and manipulation of persistent data is to be restricted. For this the system has User Authentication procedures, wherein the user has to enter his user-id and password with which his access rights will be determined.

2.5.2 Outputs

For both the reference & questioned signature samples the system produces clipped & zoomed signature files and pass on to further feature extraction processes.

2.6 User characteristics

The user of the system is required to be familiar with Windows, so as to scan the signature samples and put them in the appropriate directories specified from where the System can access them. He should also be aware of the file format in which the image has to be stored, as the system deals with .bmp files only.

2.7 Feasibility Study

The aspect of conducting a feasibility study on building a system depends on three factors namely, the technological feasibility, economic feasibility and the operational feasibility.

2.7.1 Technological Feasibility

The availability of the right technology and the equipment are the essentials for a system to achieving a technological feasibility. On focusing the light with respect to the developer's environment, the pre requisites are available. Thus proving the aspect of technological feasibility.

2.7.2 Economic Feasibility

The comfort level and the ability of the client to satisfy the developer's economic needs are fine and hence the system is economically feasible.

2.7.3 Operational Feasibility

On completion of the system as an engineered product the client intends to make use of it once when it is deployed. So the system is socially feasible.

Programming Environment

3. Programming Environments

The overall necessities to support the effective implementation of the system, which are the technical requirements, are detailed here.

Also the description of various tools used to design the system and the reasons for adopting them are presented below.

3.1 Technical Requirements

The forgery detection system needs a scanner for the purpose of obtaining the bitmap images of the reference signature and the test signature. But, since the client does not wish to dedicate a scanner exclusively to the system, there is no direct interface with the device. With regard to software requirements, the system needs Windows 9x or higher for its implementation.

3.2 Description of software & the tools used

The following are the various tools used for the program,

System Design	: UML with Rational Rose 2000
Coding	: C++ v3.0, OTL 4.0 with VC editor
Platform	: Windows'98
Database	: MS Access

Reasons for the choice of C++

It is a programming language that brings that technology into being. The software was coded using turbo C++ 3.0, thus utilizing to the language's Object Oriented Features such as abstraction, inheritance, encapsulation, polymorphism. C++ has a large user community, multiparadigm language, performance & legacy code access.

C++ is an object-oriented language with a very broad base of users. This large & thriving user community has led to high quality compilers & other development tools for a wide range of system.

Much of power of c++ comes from its support for new ways of programming. With that much support, investing in c++ is a relatively safe undertaking.

Being a multiparadigm language it supports both object based & procedure based programming thus allowing developers to choose the style of programming that is right for the task at hand. Performance wise the software is memory efficient provided it is designed properly. Well designed object oriented software is normally comprehensive & therefore amenable to performance tuning.

C++ is (mostly) backward compatible with C. This is useful in very large legacy systems where migration towards object orientation normally occurs. This backward compatibility makes it relatively inexpensive to compile legacy C codes with C++ compiler, allowing the old, non object oriented system to coexist with new object oriented subsystems.

Furthermore simply compiling the legacy C codes with a C++ compiler subjects the non-object oriented subsystem to the relatively stronger type safety checks of a C++ compiler. In today's quality sensitive culture this makes a good business sense.

Reasons for choice of System Design tools (UML with Rational Rose 2000)

UML as a modeling language conceived by the Rational Software corporation is a simple, extensible, and a very expressive visual modeling language for software intensive systems.

This is a powerful designing tool used in order to derive, the data flow, control flow, the dependencies between various modules and their associations with one another.

The system design methodology with UML comprises of,

- **Notations** which are used for representations,
- **Process** which represent the step by step procedure followed to accomplish a task, and
- **Tools** are the collection of notations & processes.

UML is called as a language because it has its own syntax and semantics. It also provides a set of notations to represent the system diagrams.

System Design

4. System Design

In the process of Software Development life Cycle (SDLC), this is the stage which charts out the system vision and its architecture.

The elaboration of the system's requirement are expressed and documented in this phase of engineering the system. All the logical and physical elements involved in the processes are distinguished in detail at this phase.

4.1 Overall Architecture

The system deals with the detection of forged signatures by analyzing the variations that arise when a set of specimen signatures are fed in and compared with that of a signature which is suspected to be forged.

For this the system initially examines the specimen signatures, which are the reference signatures of an individual, & compares it with that of the questioned signature with an intention to identify patterns and comes up with the variations that can occur in between these two signatures and deducts forgeries.

A brief study over the choice of Windows bitmap (BMP) file format for the program would facilitate the flow of moving with the system's architecture. The system on whole deals with that of the bitmap image formats only.

Bitmaps are defined as a regular rectangular mesh of cells called pixels, with each pixel containing a color value. A brief study over the image format will facilitate understanding the state of art with ease.

Graphics file formats are nothing more than the way your applications write their data to disk, one bit (the fundamental unit of computing - either a one or a zero) at a time. Some formats are definitely right for a particular task, while some are certainly wrong.

There are three basic types of graphics file formats namely,

- **Bitmap Files,**
- **Vector Files, &**
- **Meta Files.**

Bitmap files store pictures as matrices (rows and columns) of squares known as pixels, with each pixel having a particular gray or color value (also known as a gray depth, color depth, or bit depth). Image-editing programs such as Adobe Photoshop typically create Bitmap files, or by the software we use to run your scanner. TIFF (tag image file format), BMP (Windows bitmap), Mac Paint, and PCX (PC Paintbrush) are all examples of bitmap-format graphics files.

Vector files contain sets of instructions for drawing objects - typically geometric shapes such as lines, ellipses, polygons, rectangles, and arcs.

CAD programs such as AutoCAD typically create vector files. DXF (dynamic exchange format) files are examples of vector-format graphics files. Adobe PostScript paths and type, such as those we could find in an Adobe Illustrator EPS (encapsulated PostScript) file, are other examples of vector elements, but they're usually contained in a metafile.

Metafiles can contain both vector and bitmap graphics, but they don't have to have both - sometimes we would find metafiles that contain only an image, for instance. Macintosh PICT, Adobe Illustrator, EPS, CGM (computer graphics metafile), and WMF (Windows metafile) formats are all examples of metafiles.

There are a lot of different ways to talk about the files saved in these three format types. Bitmap files are generally referred to as "images" and vector files as "drawings." Metafiles are normally called "illustrations" if they contain small pieces of artwork or "publications" if they contain information for a printed page. The bitmap is a two-color data files that store an image, each bit in the data file represents one pixel; a 1 meant the pixel was on, a 0 meant the pixel was off.

Bitmap file formats involves, many file formats for storing bitmaps, such as RLE, JPEG, TIFF, TGA, PCX, BMP, PNG, PCD and GIF. One of the easiest 256-color bitmap file format is Windows' BMP. This file format can be stored uncompressed, so reading BMP files is fairly simple; most other graphics formats are compressed, and some, like GIF, are difficult to decompress.

A typical bitmap file format includes,

Data	Description
WORD Type;	File type. Set to "BM".
DWORD Size;	Size in BYTES of the file.
DWORD Reserved;	Reserved. Set to zero.
DWORD Offset;	Offset to the data.
DWORD headerSize;	Size of rest of header. Set to 40.
DWORD Width;	Width of bitmap in pixels.
DWORD Height;	Height of bitmap in pixels.
WORD Planes;	Number of Planes. Set to 1.
WORD BitsPerPixel;	Number of bits per pixel.
DWORD Compression;	Compression. Usually set to 0.
DWORD SizeImage;	Size in bytes of the bitmap.
DWORD XpixelsPerMeter;	Horizontal pixels per meter.
DWORD YpixelsPerMeter;	Vertical pixels per meter.
DWORD ColorsUsed;	Number of colors used.
DWORD ColorsImportant;	Number of "important" colors.

One of the most important things in creating a user-friendly interface is the use of bitmaps. Without bitmaps, there would be no icons, no fancy buttons, and mouse pointers would have to be made of lines.

The approach over the offline problem by establishing a local correspondence between a model and a questioned signature.

The system on its run prompts the user & seeks his/her authentication with proper user-id and password supplied. Once the user information is fetched the system connects to the database with the help of the database connectivity functions where the user information are preserved and distinguishes the user as an administrator or as a privileged user or as a general user.

An authorized user has the rights either to use the system or to generate reports (ad-hoc). The administrator has the right to restrict or grant access for a privileged user. The administrator login is the only key for handling the systems user authorities. No other users except the administrator are granted permission to manipulate with the sensitive data's but are eligible to generate reports.

On using the system the search for the reference directory begins and on locating the directory the contents of the directory are located and if found to be bitmap files they are taken in a linked list, leaving behind all other unspecified file formats if present.

For a specimen signature whose tolerance values are already stored in the systems database the system directly takes the previously calculated tolerance values and straight away looks into the availability of the test directory and takes a single bitmap file only.

Once the bitmap file linked list is formed, for both the reference & the test signatures the bitmap contents are studied based on which the image extraction takes place. For every signature file present the system clips only the signature and zooms them.

The system uses proper clipping algorithms to achieve this signature clipping. The treatment is continued for each and every bitmap file available in the linked list and once after completion of the same the files are stored again with the same file names.

Now these pre-processed signatures are ready for the further processes programmed for the system.

For a quick understanding on how further treatment follows to these pre processed bitmap files the following details are gathered from the system. The questioned signature is segmented into consecutive stroke segments that are matched to the stroke segments of the model. Comparing a set of geometric properties of the corresponding sub-strokes and computing a weighted sum of the property value differences determine the cost of the match.

The least invariant features of the least invariant sub-strokes are given the biggest weights, thus emphasizing features that are highly writer-dependent. Signature normalization is considered an important step of the process. Worries about rotation and other location transformation, uses global features that do not related directly to the shape of the signature.

Various factors taken into consideration for formulating the various. Threshold is the value selected as that gray value, which corresponds to 70% of the gray value, which has the highest frequency of occurrence in the Image. These values are calculated to come out with the Tolerance Levels for each Reference Signature and are stored in the systems database. The overall Tolerance of the original signature is thus obtained.

Random forgeries are detected when a good correspondence cannot be found, i.e. the process of making the correspondence yields a high cost. Many simple forgeries can also be identified in this way

Using the local correspondence between the model and a questioned signature, we perform skilled forgery detection by examining the writer-dependent information and try to capture unballistic motion and tremor information in each stroke segment.

Among signature 2 types of variability is observed. They are,

Intra-class variability:

It is the variation among genuine signatures.

Inter-class variability:

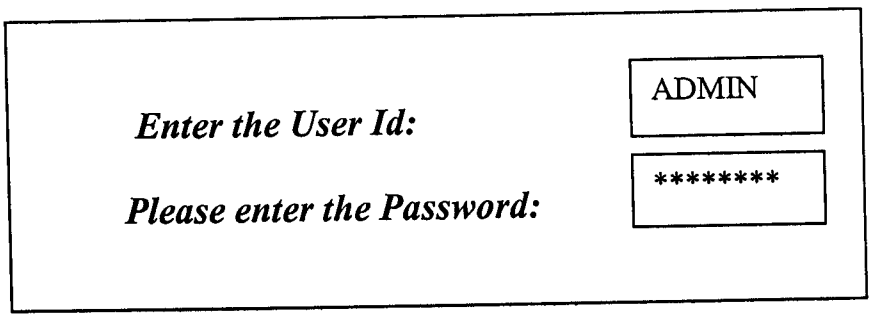
Variation between signatures of two different persons is termed as inter-class variability.

Similar procedures are repeated with the test signature and the values are compared with that of the tolerance level fixed. If the values are beyond the acceptance level the test signature is identified to be forged and the reasons for arriving at such a conclusion are listed along with the zoomed copies of all the signatures.

4.2 Input & Output Design

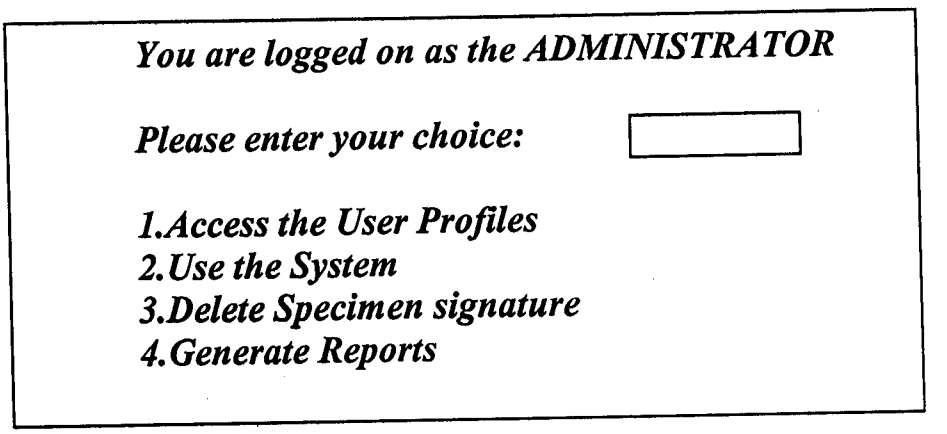
The following represent the user interface to the proposed system. As the primary interface the system prompts for the User information such as the User Id and the User password. The user id, being the primary key, uniquely identifies the record. For the user to be the administrator the user id will always be ADMIN.

Following is the content of the systems interface with the user to accept the user id and the password, in order to authenticate the user.

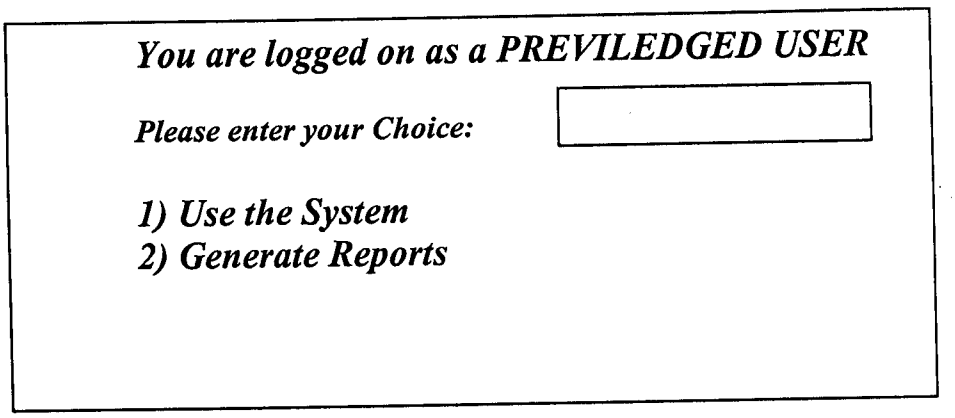


For the three user levels such as the administrator, first level user, and the general user the following are the interfaces,

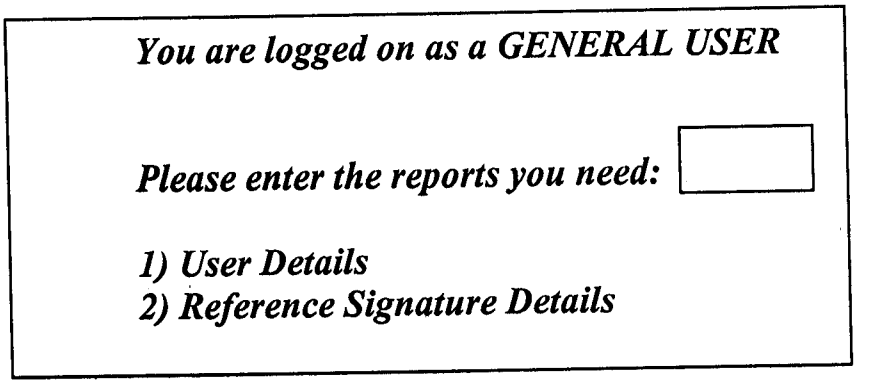
Administrator Login as shown below,



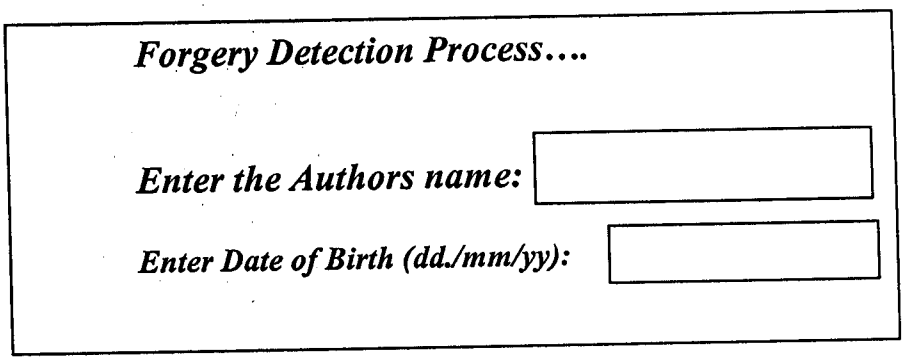
The **Privileged User's interface** is shown as follows,



Interface for the **General User,**



When the Administrator or the Privileged user opts to use the system, the interface through which the authors details are gathered appears as follows,



After the system completes the searching process in the database the following screen is displayed,

Searching the Database...

0 entries found.

Searching the Directory for .bmp files...

5 files found.

Warning: Files of other formats will be skipped!!

When the system completes searching the test directory for the questioned signature files it produces the following is projected to the user,

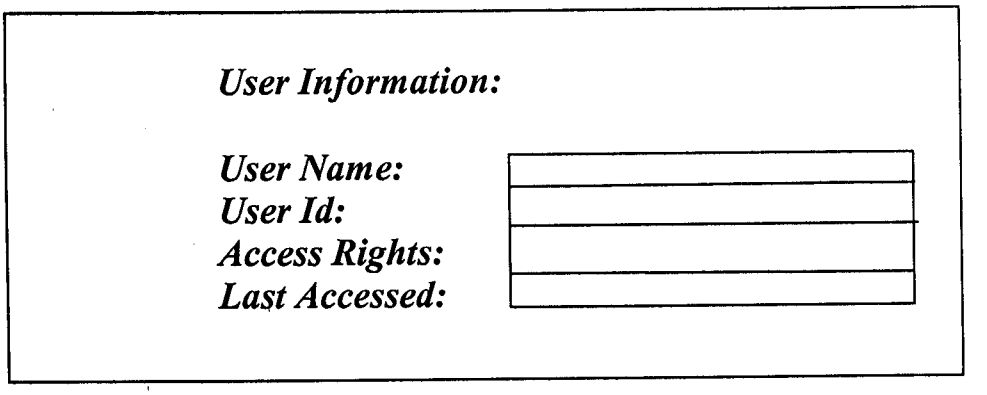
Searching Test Directory....

!!!!! Directory Empty !!!!

Please check the destination directory of the Test Signature.

Warning: Files of other formats will be skipped !!

The user information in the user table is shown as follows,



The System is capable of generating reports about the details of the users who are eligible to log on as well as the details of data stored in the Reference Table.

4.3 Database Design

The Reference Signatures of a person when input for the first time gets stored in the Reference Table. For future verifications of the same signature, the calculated values are directly fetched from the database.

Fields of the User Table

1) UserId : The serial number of the user

2) UserName : Name of the user

3) UsrPswd : User's password

4) UsrDesgn : User Designation

Composite Primary Key

4.4 Process Design

The process comprises of authenticating a valid user to the system. On using the system, it locates the destination directories namely the Reference directory where the specimen signature bitmap files are located and the Test directory, which holds the questioned signature bitmap file.

These bitmap files are taken in a linked list for further processing where they are clipped and zoomed. These are again stored back with the same file names and are ready for further processing defined in the system.

Ad hoc report generation is facilitated by the system as and when the user prompts for it fetching the information from the system database.

4.5 Supporting Diagrams

These are the diagrams, which are designed in order to facilitate an easy understanding of the system. Given below are some brief narrations about the methodologies followed for the visual depiction of the system.

4.5.1 Primary Process Control Flow Diagram

In order to present a picture on how the control flows through the system, the primary process control flow is represented below. The flow of control for the entire signature verification system is given a thought with this visual illustration. **Refer Figure (1)** (page no 27).

All the other diagrams used can be understood with ease with the following details provided.

4.5.2 DFD's & System Views

DFD (Data Flow Diagrams)

As the name indicates, data flow diagrams otherwise known as DFD's are the means of representing the flow of control among the system constituents and represent three levels as given below.

Context Level:

The context level is also known as the level 0 DFD which will give a general outline sketch of the system. **Refer Figure (2)**(page no 28).

Level 1 DFD:

The Level 1 data flow diagrams will give a representation of the overall skeleton of the system and its constituents. **Refer Figure (3)**(pg no 29)

Level 2 DFD:

These provide the control flow among the various constituents considering each one of them individually. **Refer Figure (4)**(page no 30).

System Views

The views of the system are seen with the elements of unified modeling language.

Elements of UML:

The following are the elements of UML,

- **Views,**
- **Diagrams and**
- **Model Elements**

Model elements are the simplest form of representations in UML.

The **UML** has five views. They are based on the requirements for each stage of the software development life cycle. The various views are as given below,

Usecase View:

This view makes use of the use case diagrams and is written at the analysis level. The use case diagrams make use of **Actors & Usecase,**

Actors, which come outside the boundary of the problem domain. For example we can consider a user, database from another system. Use Cases, are used to capture the functions involved. **Refer Figure (5)** (page no 31).

Behavioural View:

Under this view the Sequence diagrams, Collaboration diagrams, Start Chart diagrams are used. These can be used at both at the analysis

and at the design phases. Sequence diagrams act as an input for the class diagrams or vice versa. They are written at the analysis level and show he interaction between the various modules These are two dimensional diagrams since they grow horizontally and vertically. The lifeline goes on growing with the increasing interaction between the objects. **Refer Figure (6)**(Page no 32).

Structural View:

This serves as a view to chart out the elements of the system and makes use of the Class diagrams. These are drawn at the design level and are very rarely used at the analysis level also for identifying the ALC's (Analysis Level Classes), from where we identify the good classes which are taken for designing.

Class Diagrams represent the various classes associated and relate their association ship as uni-directional or bi-directional. **Refer Figure (7).** (page no 33)

Implementation View:

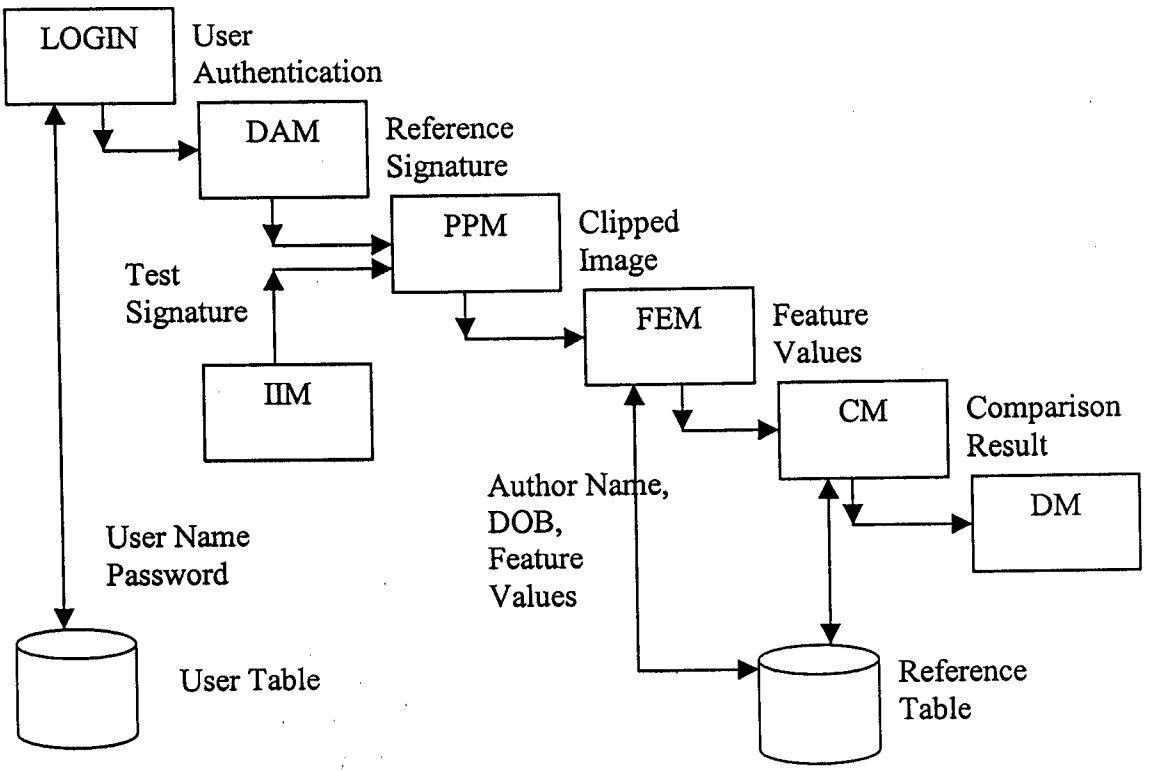
When it comes to the representation of files we describe the files used and the relationship among them. Here we use the component diagrams.

Environmental View:

This is the view which narrates about the client environment where the developers deploy the engineered product and this can be used at any phase of the software development life cycle.

4.5 Supporting Diagrams

PRIMARY PROCESS CONTROL FLOW



DAM	**Data Acquisition Module**
PPM	**Pre-processing Module**
FEM	**Feature Extraction Module**
CM	**Comparison Module**
DM	**Decision Module**
IIM	**Identification Input Module**

Figure (1)

Data Flow Diagram

CONTEXT LEVEL DFD

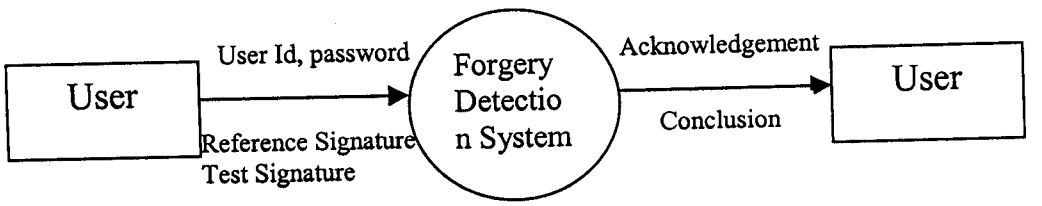


Figure (2)

LEVEL 1 DFD

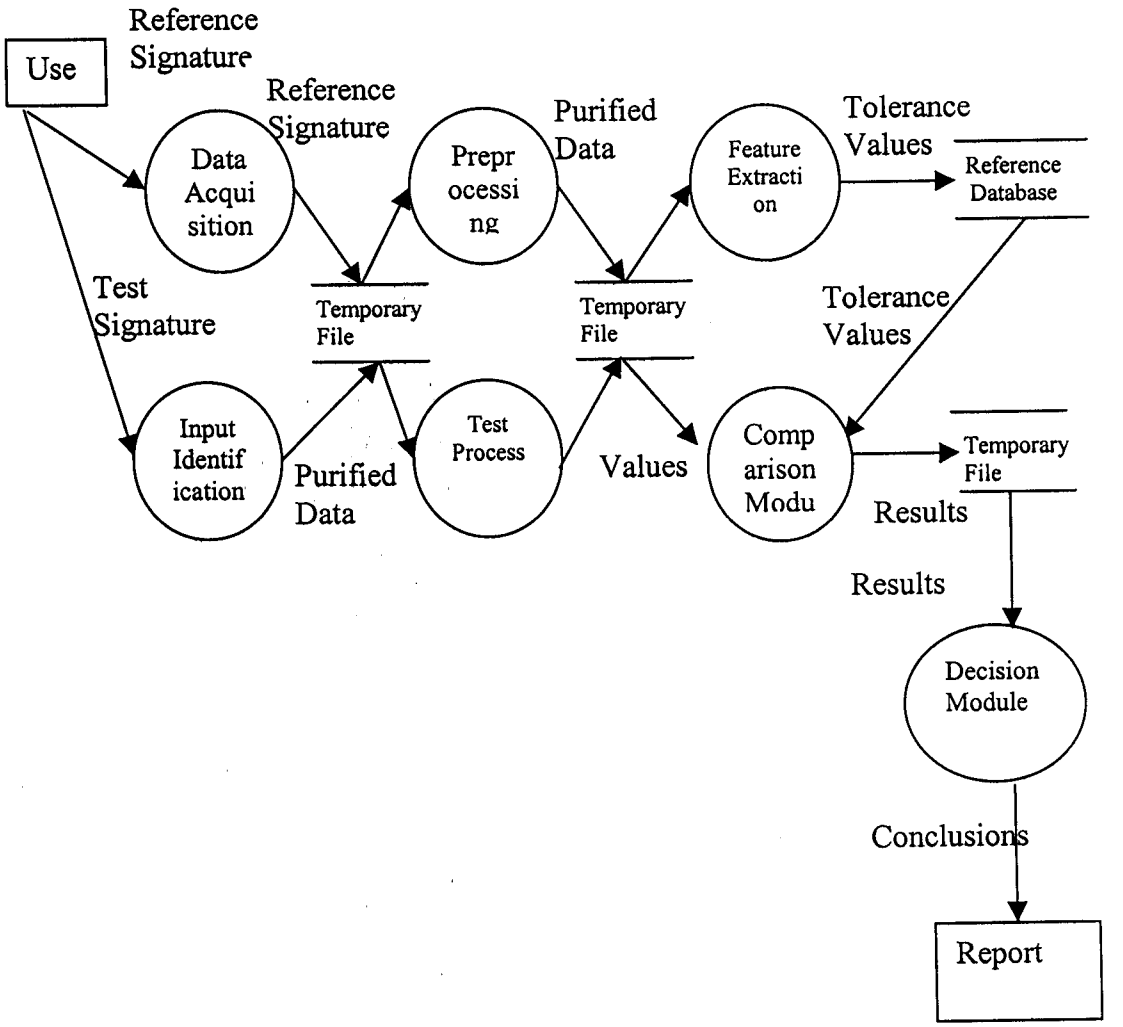


Figure (3)

LEVEL 2-DFD

LOGIN PROCESS

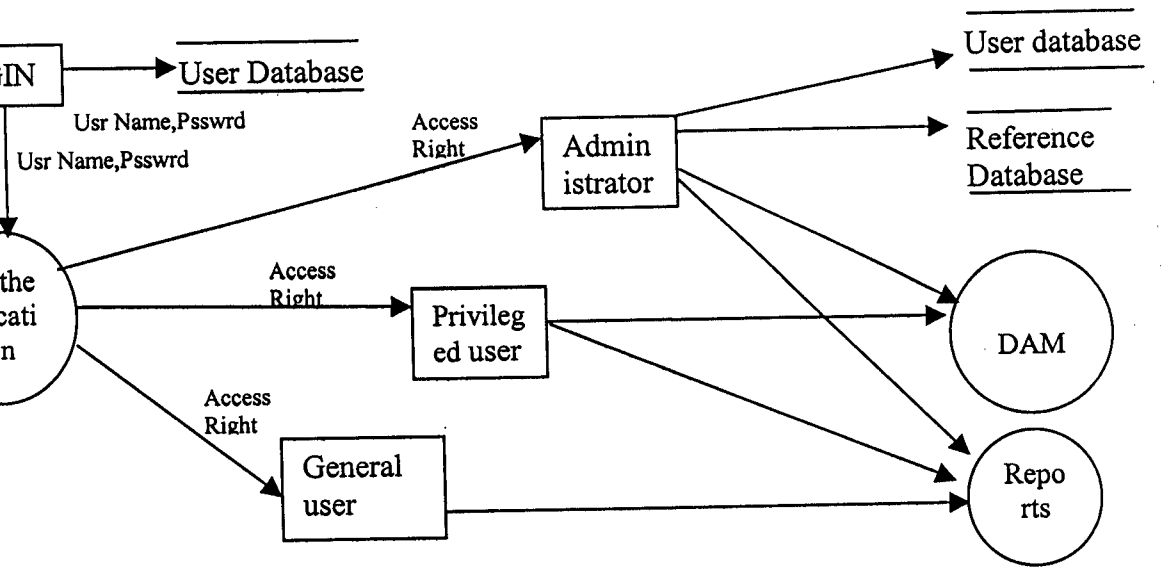


Figure (4)

Level 2 DFD

DATA ACQUISITION

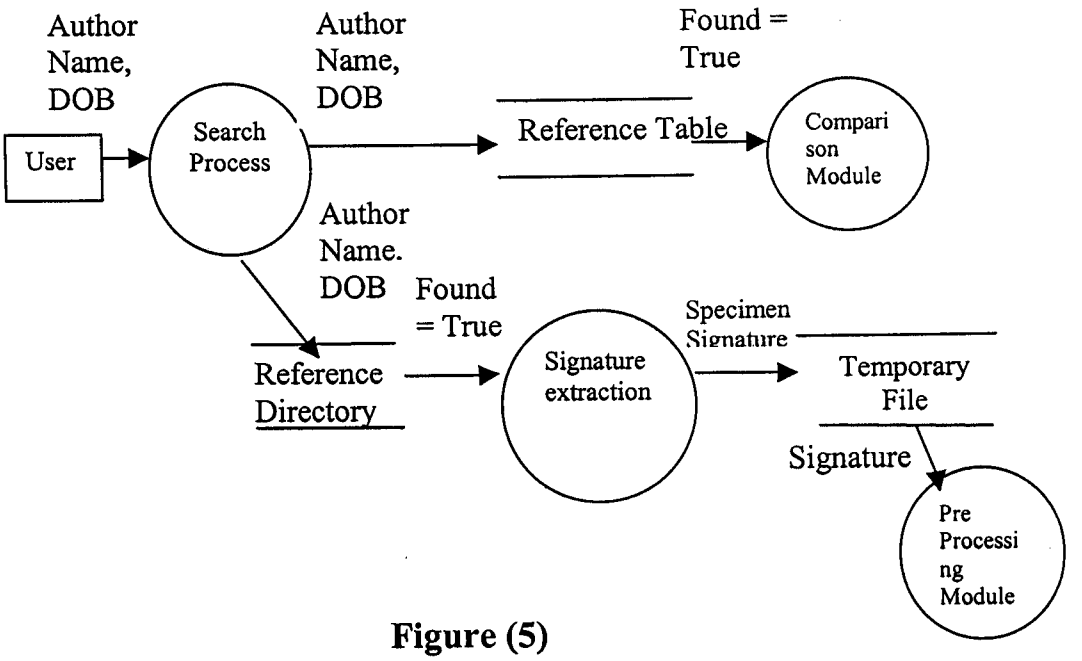


Figure (5)

Level 2 DFD

INPUT IDENTIFICATION

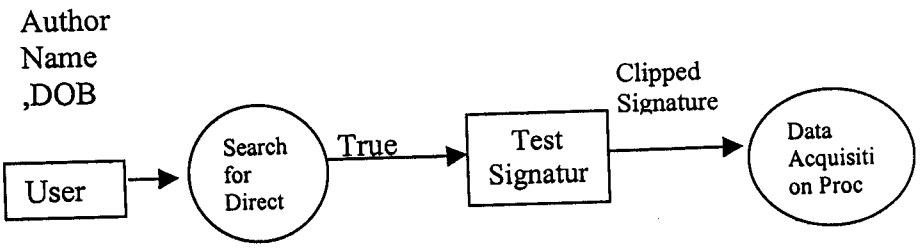


Figure (6)

USE CASE DIAGRAMS [Using UML Notations]

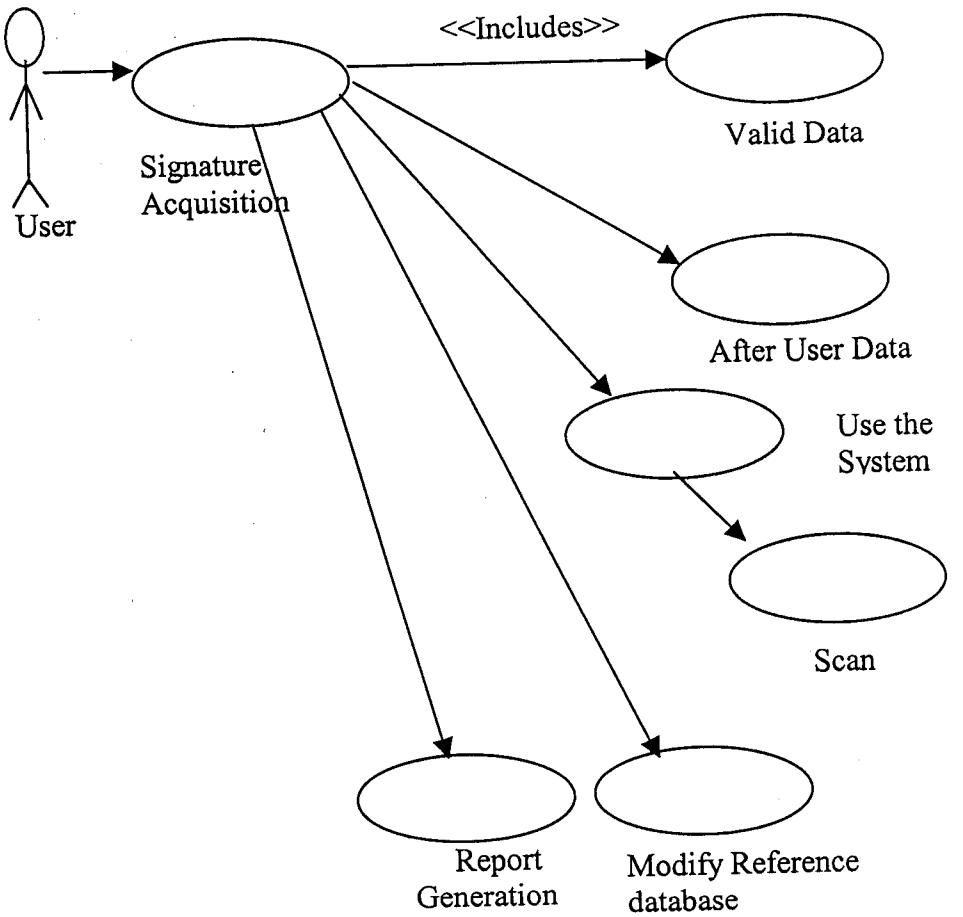


Figure (7)

SEQUENCE DIAGRAM

(For OOAD)

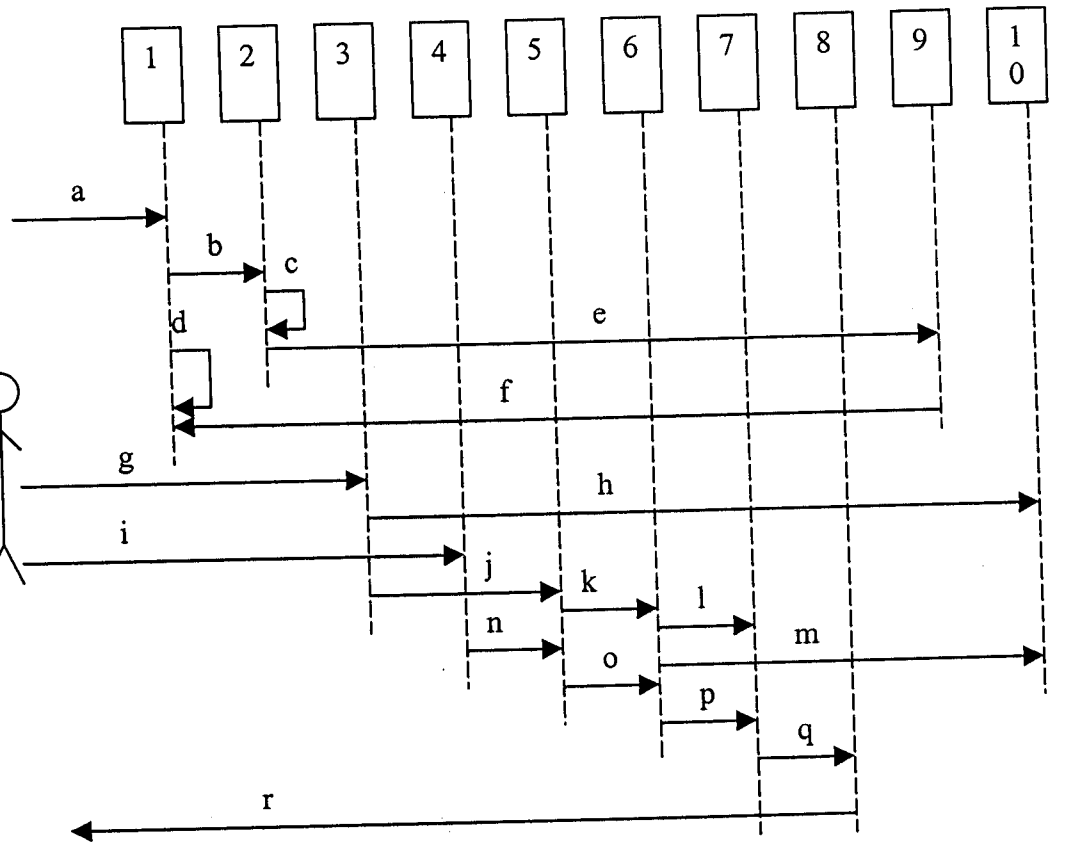


Figure (8)

1.Login Form Object

2.Connection Object

3.Reference Signature Object

4.Test Signature Object

5.Preprocessing Object

6.Feature Extraction Object

7.Comparison Object

8.Decision Object

9.AuthenticationObject

10.ReferenceRecord Object

a) User id, Password

b) Connect

c) Self-Message

d) self message

e) Authenticating the user

f) Acknowledgement

g) Reference signature(RF)

h)Copy of RF

i) Test Signature (TS)

j) Reference Signature

k) RF for Feature Extraction

l) Tolerance Values of RF

m) Store in Reference Database

n) Test Signature

o) Test Signature for Feature Extraction

p) Calculated Values for TS

q) Compared Values

r) Result to the user

CLASS DIAGRAM (For OOAD)

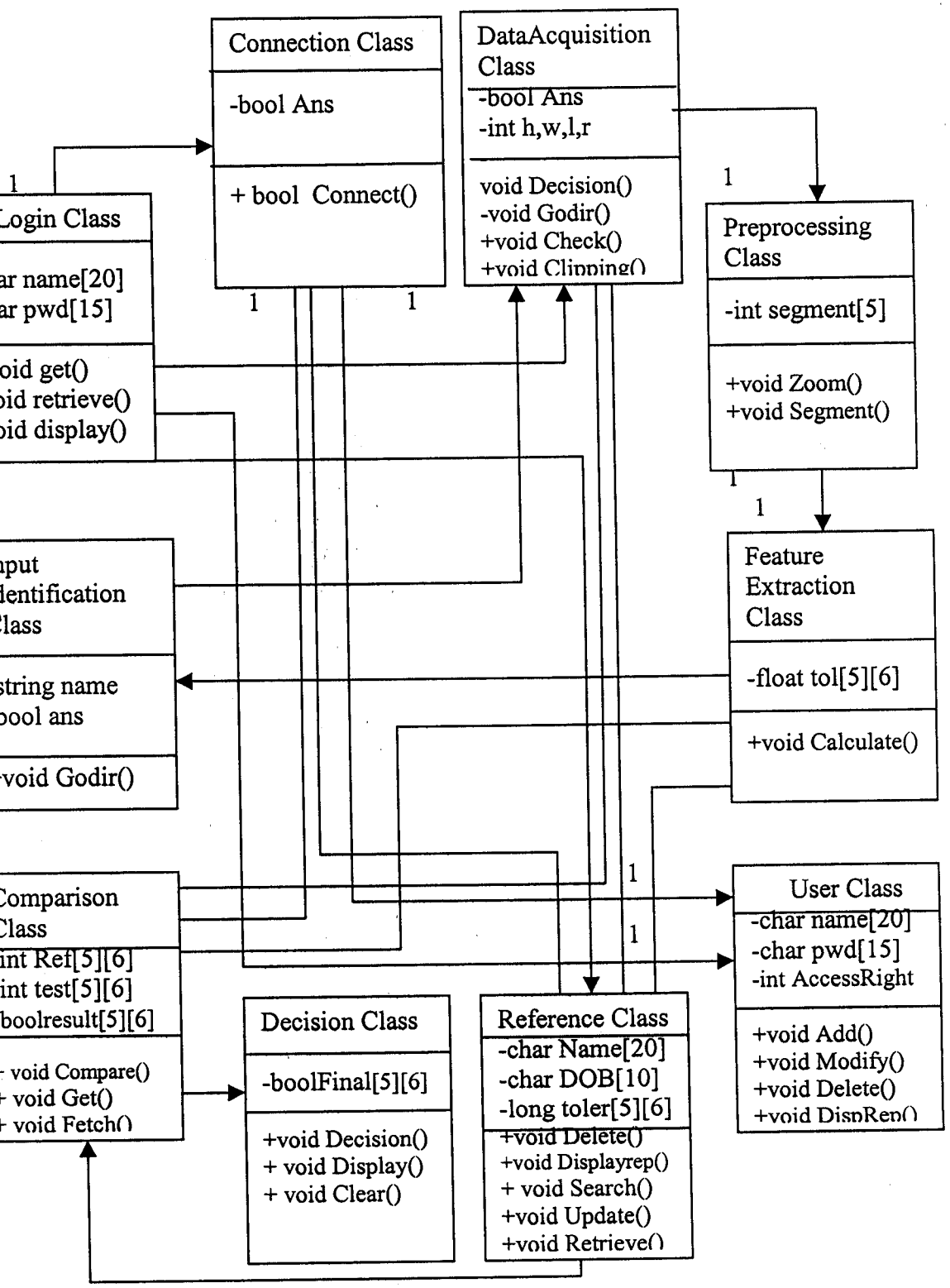


Figure (9)

System Implementation & Testing

5. System Implementation

5.1 Program Modules

The entire procedure followed by the System has been broken down into various sub processes for clarity & reduction of complexity which are as follows,

- **Login Module**
- **Connection Module**
- **Data Acquisition Module**
- **Input Identification Module**
- **Preprocessing Module**

5.2 Description of Modules

Login Module,

This module takes care of the user identification. Being a security based system all the users need to be authenticated by the system so as to use it or to generate report. The system discriminates three levels of users namely the administrator, privileged user & the general user.

The access rights of each level of user are restricted accordingly to these user levels. The administrator is allowed to use the system and he will be the only person who can give rights to other users and to

manipulate the author information, ie. To add or delete the author information from the database.

A privileged user can always use the system upon proper authentication and can generate reports on ad hoc basis. He is restricted to to manipulate with the author information's stored previously. Users at the general level can only generate reports from the system and cannot use it.

Connection Module,

Once the user information is fetched by the system database connectivity should be accomplished. So the system connects to the database with the help of the database connectivity functions where the user information are preserved.

Upon comparing the user information from the database with that of the information furnished by an user on login interface, the system gives access or denies access depending on the validity of the information.

Data Acquisition Module,

The process searches for the given name of Author and references in the database. In the case of the given author name being new and not found in the database with the previously stored information, the class makes use of functions which will search for the Reference directory which is created along with the installation of the system.

After locating the reference directory, the contents of it are studied and only the bitmap images are taken into a Linked list leaving behind all other file formats recognizing them as unknown.

The given author name if found to be stored already in the database all the sensitive data stored with respect to the author are taken into consideration. So the search for the destination directory is just skipped and the control passes on to the Input Identification module.

Input Identification Module,

The process searches for the Test directory. After locating the Test directory it checks for the contents of questioned signature file in the windows bitmap format leaving behind all other file formats (if present), recognizing them as unknown.

If more than one bitmap file of a questioned signature is present the system just reads in a single bitmap file leaving behind the rest.

Pre Processing Module,

After the Signatures are read into the System, be it a Reference Signature or a Test Signature, they need to be pre processed. The Preprocessing procedure, extracts every signature file, clips the signature content by making use of proper clipping procedures and zooms it to a predefined resolution.

These pre processed signature contents are once again stored in the same file names, which are then taken over by further processes defined to the system.

Output Screen

Enter the User Id: ADMIN

Please enter the Password: ********

****

You are logged on as the ADMINISTRATOR

Please enter your choice: 2

1.Access the User Profiles
2.Use the System
3.Delete Specimen signature
4.Generate Reports

****

Data Acquisition Process....

Enter the Authors name : Sunil Antony

Enter Date of Birth (dd./mm/yy): 23-10-1963

Gender : Male

****

Searching the Database...

0 entries found.

Searching the Directory for .bmp files...

3 files found.

Images have been successfully clipped!

Warning: Files of other formats will be skipped!!

5.3 Major Software Functions

The following are the major software functionalities programmed for this system.

Establishing the database connectivity & authenticating a user level and enabling access rights over the system either to use it or to generate reports.

Creating new users or deleting user information for the system under the supervision of the administrator. Restricting access rights to the user levels except for the administrator to manipulate with the author information stored.

Locating the destination directories namely the reference & the test directories and searching for the contents of bitmaps file formats. Skipping the search for reference directory in case of the author information found to be already stored in the database and directly switching over to the test directory definitions.

Pre processing the signature files by clipping the signature image only and then zooming the same to a pre defined resolution is yet another significant task.

5.4 File Structure and Global Data

File Structure

The Author's name and the Date of Birth, which are gathered from the user, are stored in a temporary file called "author.txt". This is a normal text file from which the details can be fetched in order to retrieve the corresponding record from the database.

Global Data

Since C++ is used for programming language, data is given more importance and via proper encapsulation and abstraction the data is shielded to prevent unwanted manipulations. The data flows between classes are made possible by setting associations and aggregation relationship between classes. The System uses Temporary Files to enable some data to be globally used by all the class.

5.5 Testing

Testing of the software ensures the extent of software reliability. This is the only phase in the software development life cycle (SDLC) which is destructive in nature, where all the steps are taken to make a program fail. This is the transitive phase in the SDLC process where the software is implemented and is tested for defects. The defects identified are documented and corrected, so that the software is coded to form an executable base and is now ready to be deployed.

5.6 Test Cases

Test cases are those which are followed to test a software intensive system for its functionalities. Also the internal working of the software is studied with the help of these test cases. Here we have two test cases namely black box testing and white box testing.

5.6.1 Opaque Box Testing

This is also called as black box testing and this is a test for validating the software's functionality. Black box testing is carried out at the later stages of testing process.

5.6.2 Translucent Testing

This is also called as glass box or white box testing. The testing concentrates on the internal working of the system and its dependencies. All the specifications are ignored and concentration is given only to the thought of how it is being done. Path coverage which checks whether all paths are covered or not is a relevant example.

5.7 System Testing

The strategies under system testing are charted out as given below.

Security Testing: Tests whether the system is possible to crack and how safe it is to use.

Stress Testing: Tests for measuring the load capacity of the system

Performance Testing: This testing strategy varies based on the results of the stress test.

Integration Testing: Testing is done with respect to the interfaces of the system and is related to the hierarchical models, either top-down or bottom-up models.

Regression Testing: This test to ensure whether or not the changes done to the system, have effect on the modules.

5.8 Test Methodologies

The modules of the System underwent Unit Testing as and when they are completed. After the results are found to be satisfactory, the modules are integrated to form the entire system, which will be subject to Integration Testing.

Finally an Acceptance Testing taken place as proposed to successfully validate the System. Regression Tests have been conducted whenever a change has been made in any of the modules to ensure that it hasn't affected the rest of the system.

5.8.1 Test cases, test data, and expected results

Test Case	Test data	Expected Result	Obtained Result
1	Wrong User Name	"Incorrect User Name! Re Enter Correct User Name"	True
2	Wrong User Id	"Incorrect User Id! Re Enter Correct User Id"	True
3	Incorrect choice of options	"Wrong choice! Please enter correct information"	True
4	Usage of literal other than characters for Author Name	"Author Name cannot contain numerals etc.,!"	True
5	Absence of Reference Directory	"Reference Directory is not found!"	True
6	Absence of Test Directory	"Test Directory is not found!"	True
7	Absence of Specimen files in Reference Directory	"Reference Directory is Empty!"	True
8	Absence of Questioned Signature files in Test Directory	"Test Directory is Empty!"	True
9	Presence of file formats other than Bitmap files	"Unknown File Format!"	True
10	Attempt to feed more than one questioned signature into the Test Directory	"Only the first bitmap file will be taken as questioned. All others will be omitted!"	True

5.8.2 The criteria on which the completion of the test is judged

If the obtained results of the testing is same as that of the expected result of the test case then the system is considered stable. When values of the expected results vary from those obtained, necessary steps have been taken to correct the differences.

Unit testing, followed by integration testing are repeated. Regression testing is done to ensure that the changes to a module have not affected the other parts of the software.

5.9 Corrective Actions

In case of mismatch found between the expected & the obtained results, corrective measures have been taken in terms of re-designing and re-coding to ensure the accuracy of the test results.

Special Notes

6. Special Notes

6.1 Alternative procedures

The alternative procedure depicts the alternative flows taken by the program incase there occurs exceptions.

An user submitting incorrect user name & user id will be alerted to re enter proper information. Whenever the user chooses or enters a wrong choice will be prompted to correct it. Presence of all other literals except for characters in the place of author name will be asked for correct name.

If a privileged or a general user attempting to make use of the system beyond his access rights, the system should not allow the user to do it & prompt to the user that his access are rights are restricted for using that particular functionality.

When the Reference or the Test Directory's are not found to be present, the system should prompt indicating the absence of the same.

While searching for the contents of bitmap files in the respective Reference & Test Directory's and when no such files are found, the system indicates that the directories are empty.

In both the directories the presence of any file formats other than bitmaps are recognized as unknown formats and a message indicating that such files will be skipped by the system appears.

Presence of more than one bitmap file in the Test directory will be indicated to the user that only a single bitmap file will be taken as the questioned signature and the rest will be omitted.

6.2 Excerpts from other related documents

The system related information, which was a part of the subject, included with the other system documents are mentioned below.

6.2.1 General Constraints and Dependencies (from Software Requirements Specification version 1.2)

Some of the cases that can occur with an accepted specimen,

Authentic:

The author's signature is very much like the suspected signature.

Imitated:

Performed by j who is not the author and his signature j has a good similarity with that of the author.

Degenerate:

One of the true samples is rejected because the author's signatures vary considerably.

False:

If performed by some j who is not the author of the reference signature and the test and the reference signatures are not alike.

This system is incapable of measuring velocity & the hand pressure when an individual signs. The system works under the assumption that no two persons can sign exactly alike and that however hard one may try he cannot succeed in making an exact replica of a genuine signature.

6.2.2 Major Design Constraints

The proposed System, being an Offline one, doesn't take into consideration the dynamic features of the signature like the hand pressure and velocity while signing. Since there is no direct interface with the scanner, we have to adopt alternate methods to ensure that the system captures the input signatures.

Conclusion

7. Conclusion

The System was successfully designed and implemented. The accuracy level of the system is satisfactory but slightly less than the proposed percentage. The expected accuracy was 90%, where as the System achieved 75% accuracy.

Since the System comes as an initial attempt by the Company at authenticating signatures and since further up-gradations have been proposed, the results produced by the system are concluded to be satisfactory.

Further Enhancements

8. Further Enhancements

With suitable device drivers & with a dedicated scanner the system could be made to communicate directly with the scanner, thus the system-scanner interface could be achieved.

The system can be made to support with more file formats as it is now dependent only on windows bitmap format. Enhancing this feature would make the system more prominent.

When the system is subject to such enhancements the need for a larger database would arise. So a much more efficient database could be made use of.

The system being coded in an internationally recognized object oriented tool, promoting the same to other object oriented languages could be achieved with ease.

With proper usage of embedded coders & code converters, more dimensions could be added to the system, so that the system is trained to be transformed into yet another intelligence system such as a neural network or AI related system.

However, Offline verification techniques form the backbone of all other verification systems and are more powerful than that of the online systems. Still by adding further features, the System can be used as a full-fledged Online Signature Verification system.

Reference Documents

9. Reference documents

Existing software documentation

The project has a revised Software Requirements Specification (version no 1.2) which describes the objective and scope of the System. It also describes the data and control flow through the System, the expected Input and Output of the Software and the Performance bounds.

The Design Document

This documents lists in detail the overall structure of the Software in general, as well as the structure of the modules. Supporting diagrams have been included which depicts the dependencies between the various modules of the software.

Technical reference

G.Rigoli, A.Kosmala, "A Systematic Comparison Between on-line and off-line Methods for Signature Verification with Hidden Markov Models", Information gathered from "Automatic Signature Verification", 2001.

N.Mohandrishnan, W.Lee, and M.Paulik, "A performance Evaluation of a New Signature Verification Algorithm", Information gathered from a published paper on "A Modern Approach to Signature Authentication", 2001.

T.Sebastian, P.Klein, B.Kimia ,"On Aligning Curves", McGraw Hill, 1992.

William Harington, "Computer Graphics", Prentice Hall of India, 1994.

Websites

- Versign.com
- Astalavista.net
- Simplythebest.net\ Image File Formats Information.htm

Appendices

Appendices

Keywords

Authentication: Verification & conformation of the genuineness signature.

Bitmaps: An image format accepted by the system.

Forgery: A deliberate attempt to counterfeit the identity of the person, by replicating his signature.

Forgery detection: Identification of forgeries.

Offline system: A system relying on the static features of the signature.

Online system: Enhanced version of the offline system where some dynamic features such as velocity, hand pressure while signing etc., are considered.

Reference Signature: Genuine signature given as input.

Signature: The way a person writes his name for authentication purpose.

Simple forgery: Forged signature with notable difference.

Substitution forgery: Signing a totally different name instead of the original.

Skilled forgery: Expert forgeries.

Stroked Segment: One of the columns to which the signature gets split for analysis.

Test Signature: The signature suspected to be forged.

Tolerance: Acceptable degree of differences between two signatures of the same person.