|  |
| --- |
|  |
| Face Recognition System |
| Synopsis |
|  |
|  |
|  |

|  |
| --- |
|  |

Table of Contents

[Introduction & OBJECTIve 3](#_Toc344539141)

[INTRODUCTION 3](#_Toc344539142)

[Objective 3](#_Toc344539143)

[PROJECT CATEGORY 3](#_Toc344539144)

[Hardware and Software Specification 3](#_Toc344539145)

[Hardware Requirement 4](#_Toc344539146)

[Software Requirement 4](#_Toc344539147)

[REQUIREMENTS AND ANALYSIS 4](#_Toc344539148)

[Problem Definition 4](#_Toc344539149)

[Existing System 4](#_Toc344539150)

[Documents maintained 6](#_Toc344539151)

[Work To Be Done 7](#_Toc344539152)

[Requirements Specification 7](#_Toc344539153)

[Functional Requirements 7](#_Toc344539154)

[Database Update 7](#_Toc344539154)

[Create User 7](#_Toc344539154)

[Face Recognition 7](#_Toc344539154)

Non [Functional Requirements 7](#_Toc344539154)

[technical specification 10](#_Toc344539155)

[Planning and Scheduling 10](#_Toc344539156)

[Gantt chart 10](#_Toc344539157)

[Tracking Gantt 11](#_Toc344539158)

[Pert chart (Network Diagram) 12](#_Toc344539159)

[Scope of the Solution 12](#_Toc344539160)

[Analysis 13](#_Toc344539161)

[Context Diagram 13](#_Toc344539162)

[Data Flow Diagram 13](#_Toc344539163)

[Level 0 DFD 13](#_Toc344539164)

[Level 1 DFD 14](#_Toc344539165)

[Level 2 DFD 16](#_Toc344539166)

[E-R Diagram 16](#_Toc344539167)

[Class Diagram 20](#_Toc344539168)

[Database & Table Details 20](#_Toc344539169)

[COMPLETE DATA STRUCTURE 22](#_Toc344539170)

[Module Description 22](#_Toc344539171)

[Face Recognigation Server 22](#_Toc344539172)

[Face Recognigation Database 23](#_Toc344539174)

[estimation 23](#_Toc344539175)

[Data Structure 24](#_Toc344539176)

[Implementation Methodology 27](#_Toc344539177)

[List of Reports 27](#_Toc344539178)

[SECURITY MECHANISM 27](#_Toc344539179)

[FUTURE SCOPE AND FURTHER REQUIREMENTS 28](#_Toc344539180)

[bIBLIOGRAPHY 28](#_Toc344539181)

# Introduction& OBJECTIVE

## Introduction

Face modeling for recognition of people is a challenging problem which has received much attention during recent years due to its many applications in different fields. Face recognition has become a very active area of research in recent years mainly due to increasing security demands and its potential commercial and law enforcement applications. The last decade has shown dramatic progress in this area, with emphasis on such applications as human-computer interaction (HCI), biometric analysis, content-based coding of images and videos, and surveillance.

Face recognition and detection is a reliable technique to authenticate a user’s identification. So facial recognition system and detection is a [computer application](https://en.wikipedia.org/wiki/Application_software) for automatically  [identifying](https://en.wikipedia.org/wiki/Identification_of_human_individuals) or [verifying](https://en.wikipedia.org/wiki/Authentication) a [person](https://en.wikipedia.org/wiki/Person) from a [digital image](https://en.wikipedia.org/wiki/Digital_image) or a [video frame](https://en.wikipedia.org/wiki/Film_frame) from a [video](https://en.wikipedia.org/wiki/Video) source any source. One of the ways to do this is by comparing selected [facial features](https://en.wikipedia.org/wiki/Face) from the image and a facial [database](https://en.wikipedia.org/wiki/Database_management_system).



Fig: Overview of Face Recognition System

Face Recognition System will enable users to authenticate themselves using their facial images which will be compared to existing database using Gabor Analysis Algorithm.



Face Recognition Algorithm will follow below mentioned steps to identify the face.



Fig: Flow of Face Recognition Process

## Objective

It’s a true challenge to build an automated system which equals human ability to recognize faces. Although humans are quite good identifying known faces, we are not very skilled when we must deal with a large amount of unknown faces. The computers, with an almost limitless memory and computational speed, should overcome human limitations. Face Recognition System will facilitate face recognition and enable user authentication using this.

Face recognition and detection typically used in security system and can be compared to other biometrics such as face or eye iris recognition systems.

Admin has to register their face images from different angel and images will be stored in database. Whenever a user wants to authenticate, the Face Recognition System will take picture of the user and compare with existing database using Gabor Analysis Algorithm.

There are following feature in face recognition system:

* Recognition from outdoor facial images.
* Recognition from non-frontal facial images.
* Preprocesses the captured images (removes background, scales size, and so on)
* Add , modify users
* Authenticate user based on registered user facial features
* Integration with Google Open ID
* Admin user account & normal user account

# PROJECT CATEGORY

This software will follow Object Oriented Programming Paradigm and use below mentioned areas:

OOP Language: Matlab, C, Java, C#.

RDBMS: MySQL 5.5.15

Networking: TCP/IP

Applications: Expert Systems

# Hardware and Software Specification

## Hardware Requirement

* **Disc capacity :** 10 MB of available hard disk space
* **RAM :** 1 GB (32 Bit) or 2 GB (64 Bit)
* **Processor :** 1.6GHz or faster
* DVD-ROM Drive / USB **Port**

## Software Requirement

* Windows XP (x86) with Service Pack 3 / Windows Vista (x86 & x64) with

Service Pack 2 / Windows 7 (x86 & x64)

# REQUIREMENTS AND ANALYSIS

## Problem Definition

### Existing System

The existing system is traditional paper books and ledger system where several records are stored and to track other details about the human. Password based authentication are kind cumbersome job for a user as he needs to remember it. Passwords are prone to be hacked. To overcome these kind of authentication issues Face Recognition System will provide an automated way of authentication using face recognition algorithms.

### Work To Be Done

We will incorporate the above mentioned workflow of a Face Recognition System in an automatic computerized way.

**Face registration – Database Creation**

* Admin will register new users by taking pictures from different angle and storing them in to the database for future use.

**Face verification–Am I who say I am?**

–a one-to-one match that compares a query face image against a template face image whose identity is being claimed

**Face identification–Who am I?**

–a one-to-many matching process that compares a query image against all template images in a face database to determine the identity of the query face

–a similarity score is found for each comparison

**Watch list–Are you looking for me?**

–is an open-universe test, the test individual may or may not be in the system database

–perform face identification first and rank similarity scores, if the highest similarity score is higher than a preset threshold, an alarm is raised

## Requirements Specification

### Functional Requirements

#### Register user Images

##### Introduction

Admin has to register their face images from different angel and images will be stored in database.

##### Input

Admin will be storing ten photos for each person from different predefined angles.

##### Processing

These images will be stored in database for processing.

##### Output

Images stored successfully to the database and ready to be accessed by face recognition system.

#### Create user account for authentication

##### Introduction:

Create account for a new User.

##### Input:

Relevant User data like user name, images, etc.

Processing:

The Face Recognition will create a new user entry.

Output:

The face recognition will generate a user to reach his face identification.

#### Face recognition

##### Introduction:

Face recognition users has to provide an image which will be matched with existing image stored in database using face recognition techniques.

##### Input:

User images captured from different resources like video, images.

##### Process:

Input image which will be matched with existing image stored in database using face recognition techniques.

##### Output:

User images will be matched and authenticated by admin.

#### Face recognition from images of varying angles

##### Introduction:

Face recognition users has to provide images from any of the predefined angles which will be matched with existing image stored in database using face recognition techniques.

##### Input:

User images captured from webcam

##### Process:

Input image which will be matched with existing image stored in database using face recognition techniques.

##### Output:

User images will be matched and authenticated.

#### Integration with google Open id

##### Introduction:

Face recognition users has to provide images from any of the predefined angles which will be matched with existing image stored in database using face recognition techniques and authenticate users corresponding Google open id account.

##### Input:

User images captured from webcam

##### Process:

Input image which will be matched with existing image stored in database using face recognition techniques and retrieve the corresponding google open id from the database.

##### Output:

User images will be matched and authenticated. Thus the user will be login to Google account without entering password manually.

### NON- Functional Requirements

This project is intended to meet the following nonfunctional requirements: -

* This face recognition software should be available on the Internet, to enable the users to use , download it any time.
* The program should be platform independent.
* Accuracy should be more
* Authentication process should not take more than one min.

### Technical specification

**Front End/ GUI Tools:** Windows Presentation Framework (WPF)

**IDE:** Matlab , Visual Studio 2010

**Framework:** Microsoft .NET 4.0 , MATLAB Imaging Framework

**Database:** MySQL

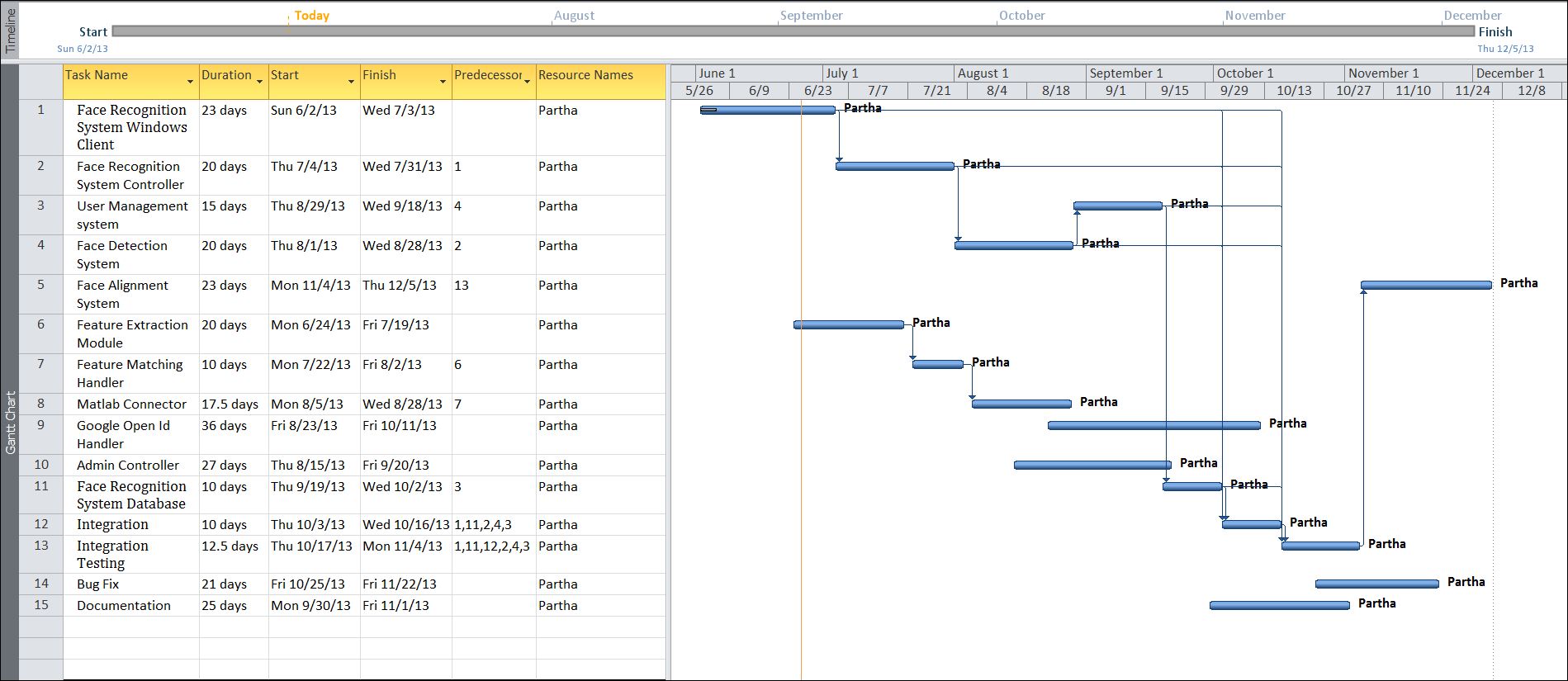
**Database Tool:** MySQL workbench CE

**Operating Systems**: Windows XP, Windows 7

**Cloud Technology**: Google Drive, Google forms, Google Open ID

## Planning and Scheduling

### Gantt chart



### Tracking Gantt



### Pert chart (Network Diagram)



# Scope of the Solution

Currently this software is aimed for a human Face Recognition. User input photo should be clear and recognizable.

We have developed this for Desktop Computers running on Windows Operating System. It can be enhanced to support UNIX / Linux, MAC OSX Operating systems.

Our software will not be integrated with Mobile Application right now. But in future we can easily extend to support that.

# Analysis

## Context Diagram



## Data Flow Diagram

### Level 0 DFD



### Level 1 DFD



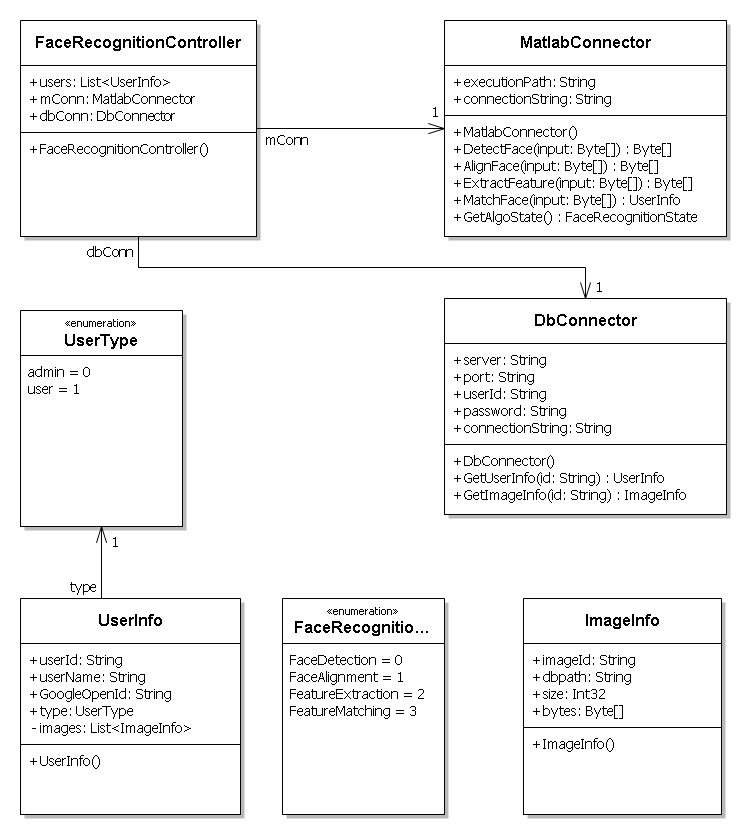


### Level 2 DFD



## E-R Diagram

## Class Diagram



# Database & Table Details

The database used for this software is called **frsdb**. A screenshot from the MySQl workbench is given below. It shows the tables and its columns. The first row is the primary key.

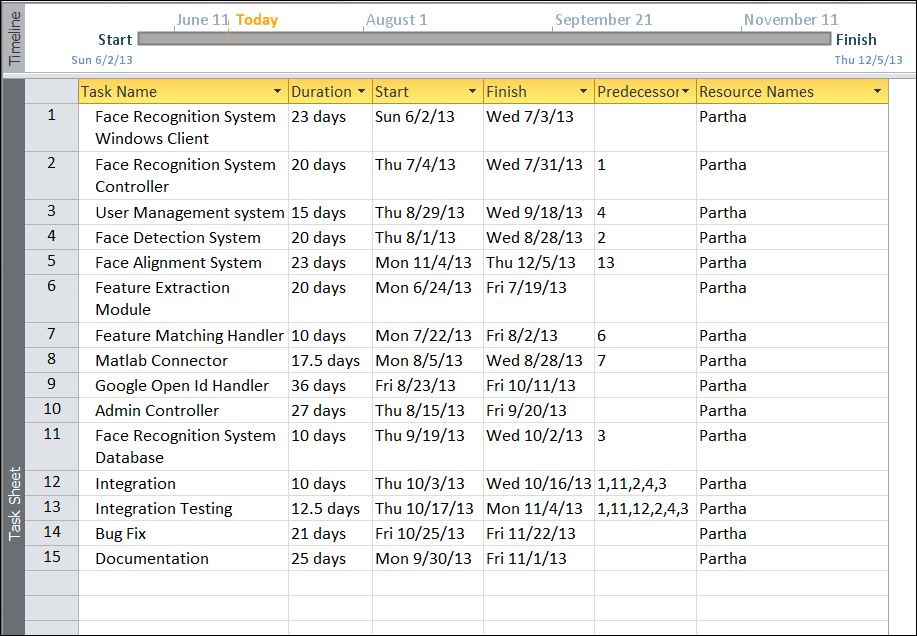
# COMPLETE DATA STRUCTURE

## Module Description

Face Recognition System is divided into following main modules such as:

1. **Face Recognition System Windows Client**: This is the GUI for interacting with face Recognition System. This will be developed in WPF.
2. **Face Recognition System Controller**: This is the main controller of the whole system. We are following MVC architecture
3. **User Management system**: this module keeps track of the user of this software. It manages user privileges.
4. **Face Detection System**: Given a single image or video, an ideal face detector must be able to identify and locate all the present faces regardless of their position, scale, orientation, age, and expression.
5. **Face Alignment System**: In face alignment, facial components, such as eyes, nose, and mouth, and facial outline are located, and thereby the input face image is normalized in geometry and photometry.
6. **Feature Extraction Module**: Feature Extraction is done using Linear Subspace Approaches such as:
   1. -Principal Component Analysis (PCA)
   2. -Linear Discriminant Analysis (LDA)
   3. -Independent Component Analysis (ICA)
7. **Feature Matching Handler**: The extracted features will be match with User input to determine a match.
8. **Matlab Connector**: Image Processing algorithms will be implemented in MATLAB. This connector enables the bridge between .NET & Matlab code.
9. **Google Open Id Handler**: This will be Google app engine plugin which will use Google scripts to interact with Google account using face recognition system.
10. **Admin Controller**: This will be an admin panel where admin can manage users, provide access, modify data.
11. **Face Recognition System Database**: Face Recognition System will have a unified database for storing all the information. It can be a networked database or a database situated in the server machine

## Estimation



## Data Structure

|  |
| --- |
| ImageInfo |
| public class ImageInfo  {  public string imageId;  public string dbpath;  public int size;  public byte[] bytes;  } |

|  |
| --- |
| UserType |
| public enum UserType  {  admin,  user  } |

|  |
| --- |
| UserInfo |
| public class UserInfo  {  public string userId;  public string userName;  public string GoogleOpenId;  public UserType type;  List<ImageInfo> images;  } |

|  |
| --- |
| FaceRecognitionState |
| public enum FaceRecognitionState  {  FaceDetection,  FaceAlignment,  FeatureExtraction,  FeatureMatching  } |

|  |
| --- |
| DbConnector |
| public class DbConnector  {  public string server;  public string port;  public string userId;  public string password;  public string connectionString;  public UserInfo GetUserInfo(string id)  {  throw new NotImplementedException();  }  public ImageInfo GetImageInfo(string id)  {  throw new NotImplementedException();  }  } |

|  |
| --- |
| MatlabConnector |
| public class MatlabConnector  {  public string executionPath;  public string connectionString;  public byte[] DetectFace(byte[] input)    {  throw new NotImplementedException();  }  public byte[] AlignFace(byte[] input)  {  throw new NotImplementedException();  }  public byte[] ExtractFeature(byte[] input)  {  throw new NotImplementedException();  }  public UserInfo MatchFace(byte[] input)  {  throw new NotImplementedException();  }  public FaceRecognitionState GetAlgoState()  {  throw new NotImplementedException();  }  } |

|  |
| --- |
| FaceRecognitionController |
| public class FaceRecognitionController  {  public List<UserInfo> users;  public MatlabConnector mConn;  public DbConnector dbConn;  } |

## Implementation Methodology

* Object Oriented Programming methodology will be adopted and Java will be used as programming language.
* Apache tomcat web server will be used to implement the server
* User interface development will be done in MVC architecture using SWT (Standard Widget Toolkit).
* Relational FRSS MySQL will be used to implement & execute SQL query to database.
* Agile Software Development model will be used while developing this software.

# SECURITY MECHANISM

* Face Recognition software is password protected software. It will be developed such a way that the admin will have complete control data.
* Admin create account can register new faces in to database.
* The data of the face will be stored in the database with an encrypted format so even if someone hacks the database somehow still he can make no real harm.
* The software will provide a backup and restore feature in case of loss of data.

# bIBLIOGRAPHY

* P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible illumination conditions? *Int. J. Computer Vision*, 28(3):245–260, July 1998.
* D. Beymer. Face recognition under varying pose. In *IEEE Conf. on Comp.*
* *Vision and Patt. Recog.*, pages756–761, 1994
* <http://en.wikipedia.org>
* <http://msdn.microsoft.com/en-us/>
* <http://www.microsoft.com/en-us/default.aspx>
* <http://www.codeplex.com/>
* <http://stackoverflow.com/>
* <http://www.codeguru.com/>
* [http://www.w3schools.com](http://www.w3schools.com/)
* [www.mysql.org](http://www.mysql.org)
* School Professionals
* **Programming Java** - E. R. Balaguruswamy