|  |
| --- |
|  |
| Photo Studio Management System |
| Synopsis |
|  |

**Prithvijit Francis Dey ( )**

|  |
| --- |
|  |

Table of Contents

[1. Introduction & Objective 3](#_Toc344542041)

[1.1 Introduction 3](#_Toc344542042)

[1.2 Objective 3](#_Toc344542043)

[3. Project Category 4](#_Toc344542044)

[4. Hardware and Software Specification 4](#_Toc344542045)

[4.1 Hardware Requirement 4](#_Toc344542046)

[4.2 Software Requirement 4](#_Toc344542047)

[5. REQUIREMENTS AND ANALYSIS 4](#_Toc344542048)

[5.1 Problem Definition 4](#_Toc344542049)

[5.2 Requirements Specification 6](#_Toc344542050)

[5.2.1 Functional Requirement 6](#_Toc344542051)

[5.2.2 Technical Specification 8](#_Toc344542052)

[5.3 Planning and Scheduling 9](#_Toc344542053)

[5.3.1 Gantt chart 9](#_Toc344542054)

[5.3.2 Tracking Gantt 9](#_Toc344542055)

[5.3.3 Pert Chart 10](#_Toc344542056)

[6. Scope of the Solution 10](#_Toc344542057)

[7. Analysis 11](#_Toc344542058)

[7.1 Context Diagram 11](#_Toc344542059)

[7.2 0-Level DFD 11](#_Toc344542060)

[7.3 1-Level DFD 12](#_Toc344542061)

[7.4 2-Level DFD 14](#_Toc344542062)

[7.5 E-R Diagram 15](#_Toc344542063)

[7.6 Class Diagram 18](#_Toc344542064)

[8. Database & Table Details 19](#_Toc344542065)

[9. Complete Structure 19](#_Toc344542066)

[9.1 Module Description 19](#_Toc344542067)

[9.2 estimation 20](#_Toc344542068)

[9.2 Data Structure 21](#_Toc344542069)

[9.4 Implementation Methodology 23](#_Toc344542070)

[9.5 List of Reports 23](#_Toc344542071)

[11. Implementation of Security Mechanism at Various Levels 23](#_Toc344542072)

[12. Future Scope & Further Enhancement of the Project 24](#_Toc344542073)

[13. Bibliography 24](#_Toc344542074)

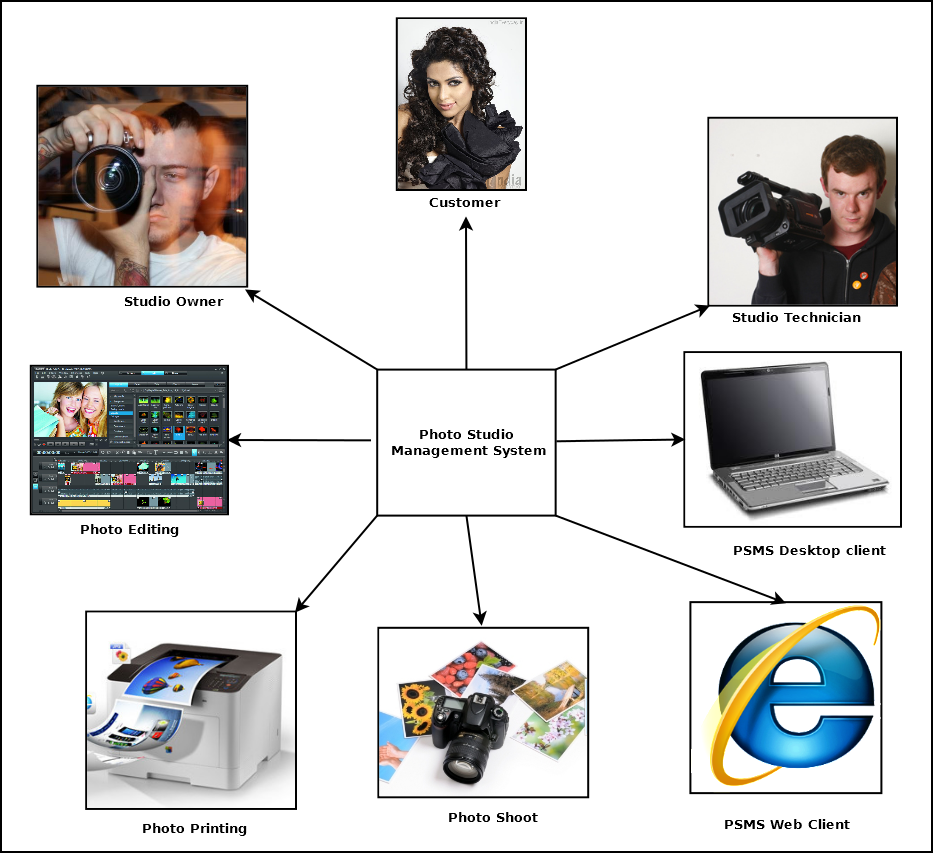
# Introduction & Objective

## Introduction

With the advance of Information Technology and digitalization, work of a photo studio has become more and more complicated as technology now provides lots of scope in the area of a photography and media. Eventually the demands of the customers are increasing and it is being more hectic for a photo studio owner to cop up with these. Customers now expect that the studio would keep track of his photos developed in the studio forever, not only that, they want the studio to customize their photos and keep track of them as well. They expect a quick delivery without any delay. It is really difficult to meet the expectations without proper tracking system. That’s where Photo Studio Management System comes in. It is going to be developed to fulfill all the needs of a photo studio.

## Objective

Photo Studio Management System (will be referred as **PSMS** in this document) is an under development application that contains all the features a photo studio manager expects to manage his studio. It is going to be developed such a way that the Graphical User interface would be very easy to understand as a photo studio owner might not be very user friendly. It has many features like, a person can manage his employee details, a person can manage his customers, keep track of his work progress, his income/ expense, his delivery date, work progress etc.



A photo editor will also be available for editing photos and save them in database. A trainee management feature has also been entered if a studio provides training to people. In short, this software will be sufficient to control all the things in a photo studio.

# 3. Project Category

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

**OOP Language:** C#

**RDBMS**: MySQL 5.5.15

**Networking**: TCP/IP

**Applications**: Expert Systems

# Tools/Platform, Hardware and Software Specification

## 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)

* Microsoft .NET 4.0

## Tools/Platforms used (Hardware/Software):

### Hardware Used

* Laptop with 2GHZ processor
* 2 GB RAM
* 320 GB hard disk (NTFS File System)

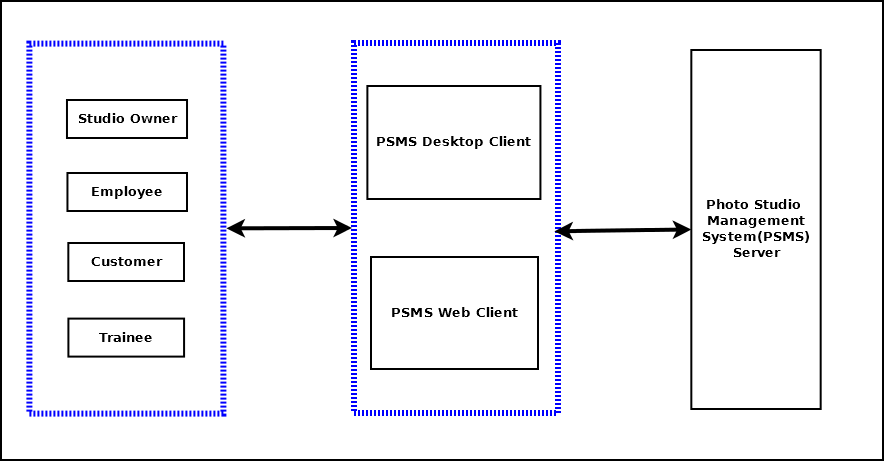
### Software Used

* Windows XP (x86) with Service Pack 3 / Windows Vista (x86) with Service Pack 2 / Windows 7 (x86)
* Microsoft .NET 4.0
* Windows Presentation Framework(WPF)
* Windows Communication Framework(WCF)
* Visual Studio 2010 Express Edition (IDE)
* MySQL Workbench
* Dia for Diagram Drawing & Modelling.

# REQUIREMENTS AND ANALYSIS

## Problem Definition

Advancement of information technology has taken media and photo editing world to another level. Customers want us to take his photo, process it, edit it, and deliver it in time and save the copies for the future use as well. It is really irritating to keep records using pen and paper. It is almost impossible to keep track of the delivery date of many customers and deliver them in time. People want to print their photos and edited photos in future and it is necessary to find them easily as well. So it is really necessary to use software like Photo Studio Management System to take complete control of studio business.



## Requirements Specification

### 5.2.1 Functional Requirement

#### 5.2.1.1 Add Employee

**Introduction:**

Add a new employee.

**Input:**

Relevant employee data like name, address, contact number, educational qualification.

**Processing:**

Admin will enter the data in the **PSMS** and create a new employee entry.

**Output:**

The **PSMS** will generate an employee details I for future reference.

#### 5.2.1.2 Customer Requirement storing and tracking

**Introduction:**

Admin can add customer requirement in database and track them.

**Input:**

Admin will enter data provided by customer like work type, work description, completion date etc.

**Processing:**

The **PSMS engine** will store the information in the database.

**Output:**

The **PSMS engine** will show them to the employees and admin for work managing.

#### 5.2.1.3 Search customer data in future

**Introduction:**

A customer who have registered and have relevant data in the system (like digital photo) can be searched in the future along with business details and his/her data

**Input:**

Admin will provide a search condition to the system like customerId.

**Processing:**

The **PSMS** will search the database of the customers using the provided input.

**Output:**

The **PSMS** will display positive and negative search results according to the availability of relevant data.

#### 5.2.1.4 Edit customer photos

**Introduction:**

Employees can edit photos of customers as per customer requirement in the PSMS photo editor.

**Input:**

Employees will modify customer photos in the editor and ask the system to save both the copies.

**Processing:**

The **PSMS** will store both the copies in the database.

**Output:**

The **PSMS** engine will fetch the original and modified copy for future use.

#### 5.2.1.5 Save a digital copy of the photos printed

**Introduction:**

System must automatically store a digital copy of all the printed photos along with time and date.

**Input:**

Employees uses the print photo option.

**Processing:**

The **PSMS** will save them in a private/ public folder.

**Output:**

The **PSMS** will show the data with details only to the persons with proper aauthority.

#### 5.2.1.6 Add a new trainee

**Introduction:**

**PSMS** will manage trainees coming to the studio.

**Input:**

Trainee provided data like name, address, contact details and other details like payment status etc. will be sent to the system by employees/ admin.

**Processing:**

PSMS will save the entire details in the trainee database.

**Output:**

Details could be shown to the employees/ admin in future whenever is necessary.

#### 5.2.1.7 Mobile entry and query

**Introduction:**

**PSMS** data can be entered and queried from a java mobile device where the mobile app version of the PSMS is preinstalled.

**Input:**

Admin will sync the data stored in the database with a cloud database.

**Processing:**

PSMS data will be saved to the cloud based database and will be available for mobile database if user syncs from the mobile with an active internet connection.

**Output:**

After syncing, admin can see relevant data like work status etc. from his mobile device.

### 5.2.2 Technical Specification

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

**IDE:** Visual Studio 2010

**Framework:** Microsoft .NET 4.0

**Database:** MySQL

**Database Tool:** MySQL workbench CE

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

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

## 5.3 Planning and Scheduling

### 5.3.1 Gantt chart

****

### 5.3.2 Tracking Gantt

****

### 5.3.3 Pert Chart



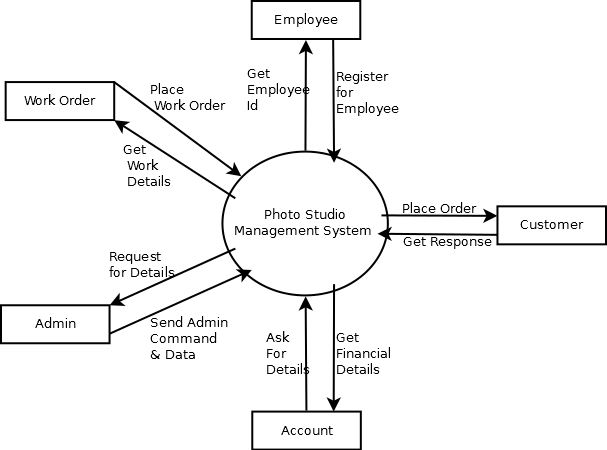
# 6. Scope of the Solution

The main features of Photo Studio Management System are:

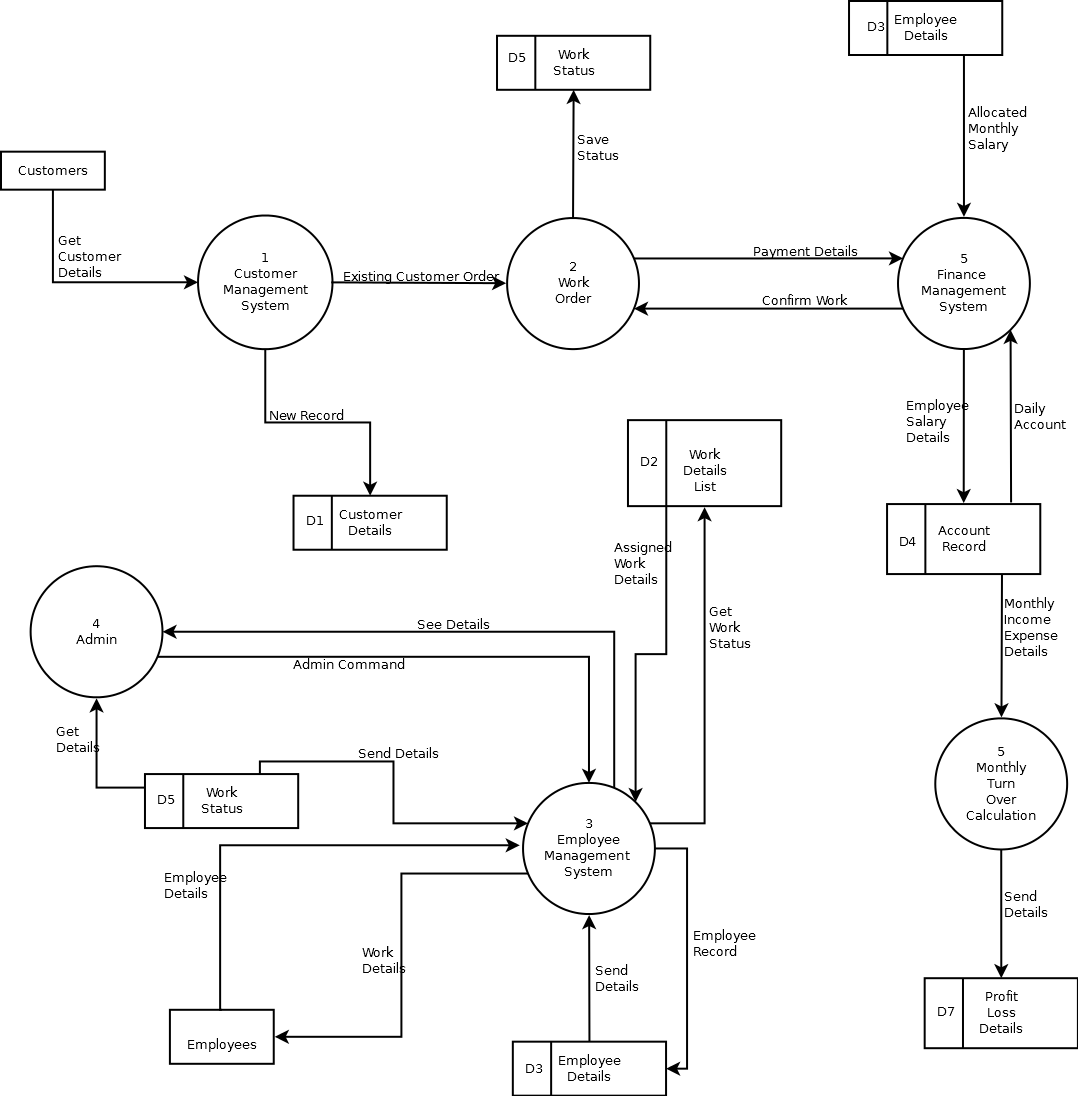
1. Secure user login and data access
2. Computerized records of employee details and customer requirements
3. Query & update customer work progress and delivery date.
4. Edit customer photos and store them in database long with original photos.
5. Search by customer name/ id to find photos/ edited photos etc. of a customer.
6. Java/ android based mobile app for the owner to track status update.

# Analysis

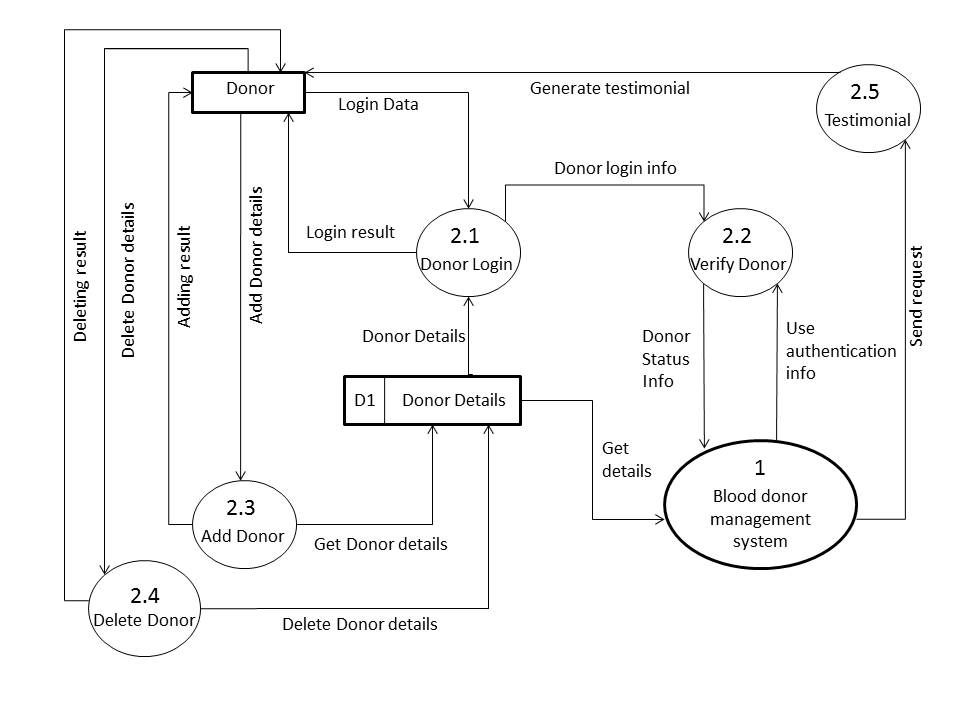
## Context Diagram

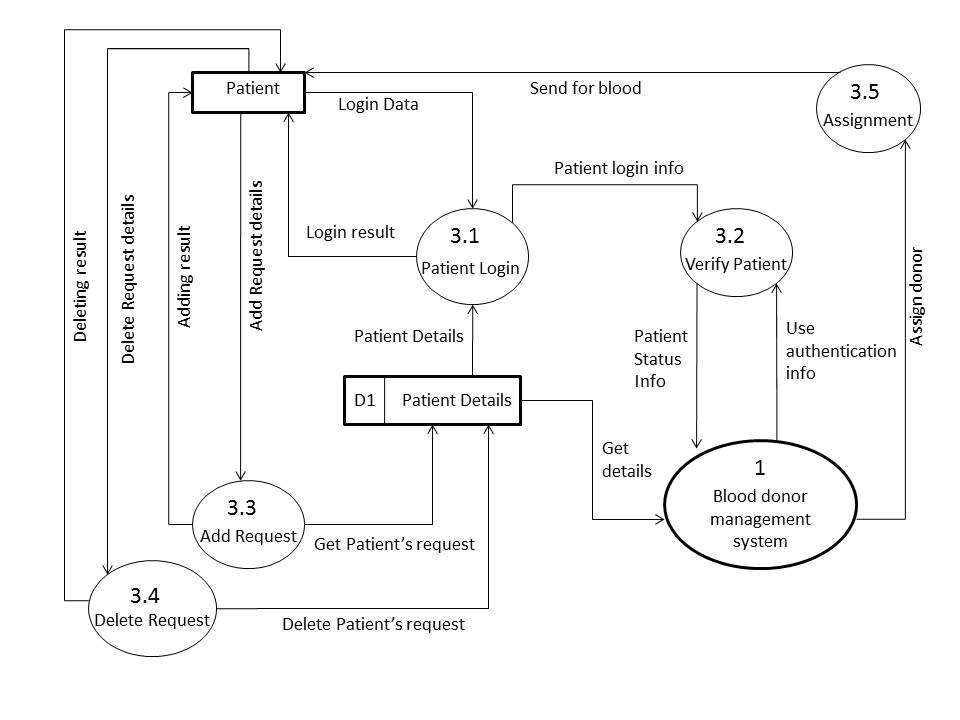


## 0-Level DFD

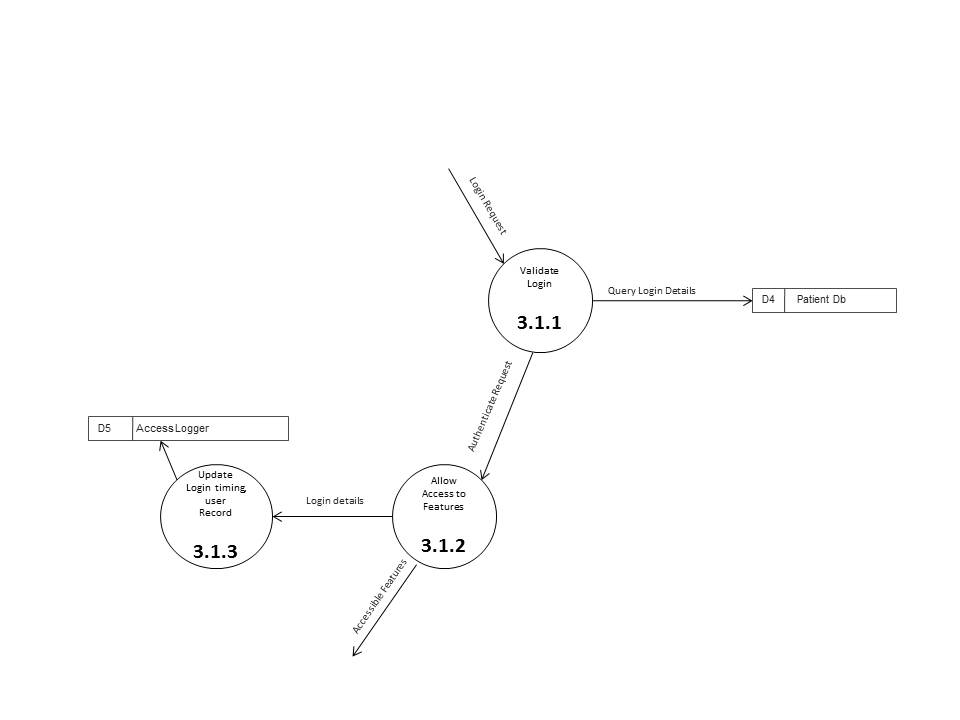


## 1-Level DFD





## 2-Level DFD



## E-R Diagram

We will design a RDBMS for Photo Studio Management System. The entities and their attributes are listed below. Attributes in Bold letter is the unique key.

|  |  |
| --- | --- |
| **Entities** | **Attributes** |
| customers | **customerId,** customerName**,** customerAddress, customerEmail, customerContactNumber |
| employees | **employeeId**, employeeName, employeeAddress, employeeJoinDate, employeeQualification, employeeContactNumber, employeeEmail, employeeContractDetails, employeeDOB |
| trainees | **traineeId,** traineeName, traineeEmail, traineeQualification, traineeAddress, traineeContactNumber, traineePaymentStatus |
| workDetails | **workId,** customerId, employeeId, workDeadline, workDescription |
| accounts | **amountId,** amountDescription, amountType, accountBalance, amountalue |
| workStatus | **workId,** workProgress, workLeft |
| lossProfitDetails | **transactionId**, transactionAmount, transactionType, transactionDetails, monthlyIncome |
| editedData | **editedDataId**, workId, customerId |

**Relationship between Entities:**

* Photo Studio Management System has employees  1 : N
* Photo Studio Management System has customers 1 : N
* Photo Studio Management System manages trainees 1 : N
* Photo Studio Management System manages accounts 1 : 1
* Customers pay to accounts N:1
* Employees receive from accounts N:1
* Photo Studio Management System monitors work status 1 : N
* Photo Studio Management System monitors lossProfitDetails 1 : 1
* Accounts affects lossProfitDetails 1 : 1
* Employees report to work status N:N
* Photo Studio Management System adds workDetails 1 : N
* Employees check workDetails N : N
* Employees edit editedData N : N
* Employees train trainees N : N
* Admin organizes events→1:1



## Class Diagram



# 8. Database & Table Details

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

# 9. Complete Structure

## 9.1 Module Description



The main components of Blood Donation Management system are:

* **PSMS Server**

This is the backbone of PSMS. This dedicated server stores all the data and handles the business logic of the software. It also handles interaction with clients and data syncing. This is divided into two major blocks.

* + **PSMS Engine**: This is the controller of PSMS. This takes decision based on the business logic and employs other components.
  + **PSMS Database**: This is the centralized storage of all PSMS data.
* **PSMS Desktop Client**

This is the full functionality client of PSMS which will be deployed on desktop computers and laptops.

* **PSMS Web Client**

This is the limited functionality client which can be accessed using any web browser.

* **PSMS Mobile Client**

This is the limited functionality client which can be accessed using any mobile devices such as Mobile phone, tabs & pads.

* **PSMS Sync Manager**

This component handles the data syncing between server and web & mobile clients.

## 9.2 estimation



## 9.2 Data Structure

|  |
| --- |
| **DonorInfo** |
| public class DonorInfo  {  public string id { get; set; }  public string name { get; set; }  public string bloodGroup { get; set; }  public string phone { get; set; }  public string address { get; set; }  public DateTime lastDonateDate { get; set; }  public DateTime dob { get; set; }  } |

|  |
| --- |
| **EventInfo** |
| public class EventInfo  {  public string id { get; set; }  public string eventTitle { get; set; }  public DateTime eventDoe { get; set; }  public string eventVenue { get; set; }  public string eventGoal { get; set; }  } |

|  |
| --- |
| **ExpenseInfo** |
| public class ExpenseInfo  {  public string id { get; set; }  public string purpose { get; set; }  public DateTime doe { get; set; }  public string expensed\_by { get; set; }  public double amount { get; set; }  } |

|  |
| --- |
| **FundInfo** |
| public class FundInfo  {  public string id { get; set; }  public string wellwisher\_name { get; set; }  public string contact { get; set; }  public DateTime dod { get; set; }  public string received\_by { get; set; }  public double amount { get; set; }  } |

|  |
| --- |
| **MemberInfo** |
| public class MemberInfo  {  public string id { get; set; }  public string name { get; set; }  public DateTime doj { get; set; }  public string address { get; set; }  public string phone { get; set; }  } |

|  |
| --- |
| **PatientInfo** |
| public class PatientInfo  {  public string id { get; set; }  public string name { get; set; }  public string bloodGroup { get; set; }  public int age { get; set; }  public string address { get; set; }  public string phone { get; set; }  public string admittedAddress { get; set; }  public DateTime expectedDate { get; set; }  public string assignedDonor { get; set; }  public string donorContact { get; set; }  } |

|  |
| --- |
| **TodoInfo** |
| public class TodoInfo  {  public string id { get; set; }  public DateTime date { get; set; }  public string details { get; set; }  } |

|  |
| --- |
| **WellWisherInfo** |
| public class WellWisherInfo  {  public string id { get; set; }  public string name { get; set; }  public string address { get; set; }  public DateTime doj { get; set; }  public string phone { get; set; }  public string remarks { get; set; }  } |

## 9.4 Implementation Methodology

* Object Oriented Programming methodology will be adopted
* User interface development will be done in MVC architecture using Windows Presentation Framework.
* Relational DBMS MySQL will be used to implement & execute SQL query to database.
* Agile Software Development model will be used while developing this software.

## 9.5 List of Reports

List of reports that are likely to be generated in this software are given below:

* Photo Studios will be given a certificate as a token of appreciation
* List of donors can be generated
* List of patients can be generated
* List of members can be generated
* Event details can be generated
* Fund details can be generated
* Yearly donation report can be generated
* Case study report can be created

# 11. Implementation of Security Mechanism at Various Levels

* This software requires a valid password to login and then it allows using any of its features.
* The login password will be saved in encrypted format in database.
* This software will use Google open-id authentication for web interface.

# 12. Future Scope & Further Enhancement of the Project

This application will be developed for Windows operating system (Win7, Win XP) only; in future we are planning to make it runnable under LINUX, MAC operating system also.

In case of mobile client we would develop it for java supported mobiles only; in future we would extend it to make it runnable under other mobile operating systems like Android, iOS or Windows Mobile OS.

Our web client will be developed using Google App Framework& Google Doc interface. Web client & Mobile client will not be synced automatically; it will require a manual sync with the server. It is under a continuous process of development and we are working hard to make it perfect and error free project

# 13. Bibliography

* 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
* www.mysql.org