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

**Prithvijit Francis Dey ( )**

|  |
| --- |
|  |

Table of Contents

[Introduction& Objective 3](#_Toc351368845)

[Introduction 3](#_Toc351368846)

[Objective 3](#_Toc351368847)

[System Analysis 4](#_Toc351368848)

[Identification of Need 4](#_Toc351368849)

[Preliminary Investigation 5](#_Toc351368850)

[Feasibility Study 5](#_Toc351368851)

[Project Planning 5](#_Toc351368852)

[Project Scheduling (PERT Chart and Gantt Chart both) 5](#_Toc351368853)

[Gantt chart 5](#_Toc351368854)

[Tracking Gantt 6](#_Toc351368855)

[Pert Chart 7](#_Toc351368856)

[Software requirement specifications (SRS) 7](#_Toc351368857)

[Functional Requirement 7](#_Toc351368858)

[Non-functional Requirements 11](#_Toc351368859)

[Software Engineering Paradigm applied 11](#_Toc351368860)

[Data models (like DFD) 11](#_Toc351368861)

[Context Diagram 12](#_Toc351368862)

[0-Level DFD 12](#_Toc351368863)

[1-Level DFD 14](#_Toc351368864)

[2-Level DFD 15](#_Toc351368865)

[Control Flow diagrams, 16](#_Toc351368866)

[State Diagrams/Sequence diagrams, 16](#_Toc351368867)

[Entity Relationship Model, 16](#_Toc351368868)

[Class Diagrams/ 19](#_Toc351368869)

[CRC Models/Collaboration Diagrams/Use-case Diagrams/Activity Diagrams depending upon your project requirements 20](#_Toc351368870)

[System Design 20](#_Toc351368871)

[Modularisation details 20](#_Toc351368872)

[Data integrity and constraints 20](#_Toc351368873)

[Database design, Procedural Design/Object Oriented Design 20](#_Toc351368874)

[User Interface Design 20](#_Toc351368875)

[Test Cases (Unit Test Cases and System Test Cases) 20](#_Toc351368876)

[Coding 20](#_Toc351368877)

[Complete Project Coding 20](#_Toc351368878)

[Comments and Description of Coding segments 20](#_Toc351368879)

[Standardization of the coding 20](#_Toc351368880)

[Code Efficiency 20](#_Toc351368881)

[Error handling 20](#_Toc351368882)

[Parameters calling/passing 20](#_Toc351368883)

[Validation checks 20](#_Toc351368884)

[Testing 21](#_Toc351368885)

[Testing techniques and Testing strategies used 21](#_Toc351368886)

[Testing Plan used 21](#_Toc351368887)

[Test reports for Unit Test Cases and System Test Cases 21](#_Toc351368888)

[Debugging and Code improvement 21](#_Toc351368889)

[System Security measures (Implementation of security for the project developed) 21](#_Toc351368890)

[Database/data security 21](#_Toc351368891)

[Creation of User profiles and access rights 21](#_Toc351368892)

[Cost Estimation of the Project along with Cost Estimation Model 21](#_Toc351368893)

[Reports (sample layouts should be placed) 22](#_Toc351368894)

[Future scope and further enhancement of the Project 22](#_Toc351368895)

[Bibliography 22](#_Toc351368896)

[Appendices (if any) 23](#_Toc351368897)

[Glossary. 23](#_Toc351368898)

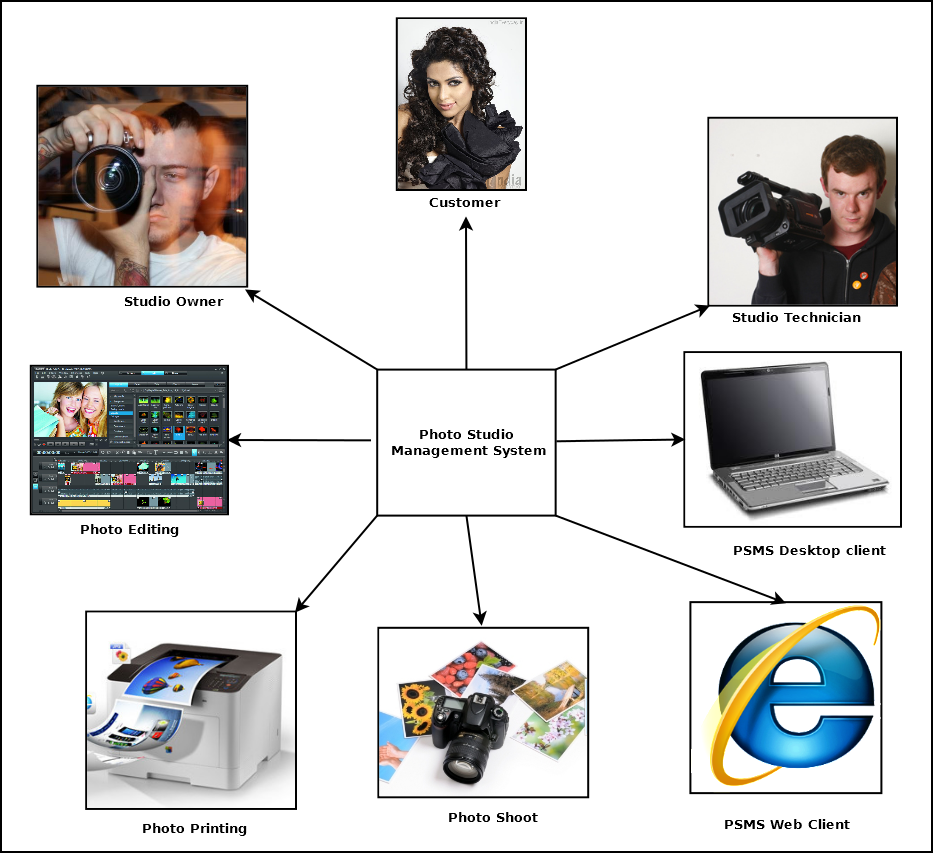
# Introduction& Objective

## 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 managehis 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.

# System Analysis

## Identification of Need

We frequently visit a local photo studio for photo related needs and they ask for an ex envelop through which the photo was delivered or their own generated number. This problem is often come out at the time of reprint any photo. To maintain the envelop or the number is very difficult to a customer. On the other hand various studio related function that the owner of studio maintains manually like account, task, manually bill generation etc. though he has a computer in his studio. The studio owner also faces a problem when he has to maintain more than one studio with other helping hands. To maintain the technicians information or monitor the work flow is very difficult. These are the main things, which inspire me to develop this management system to minimize the manual effort.

## Preliminary Investigation

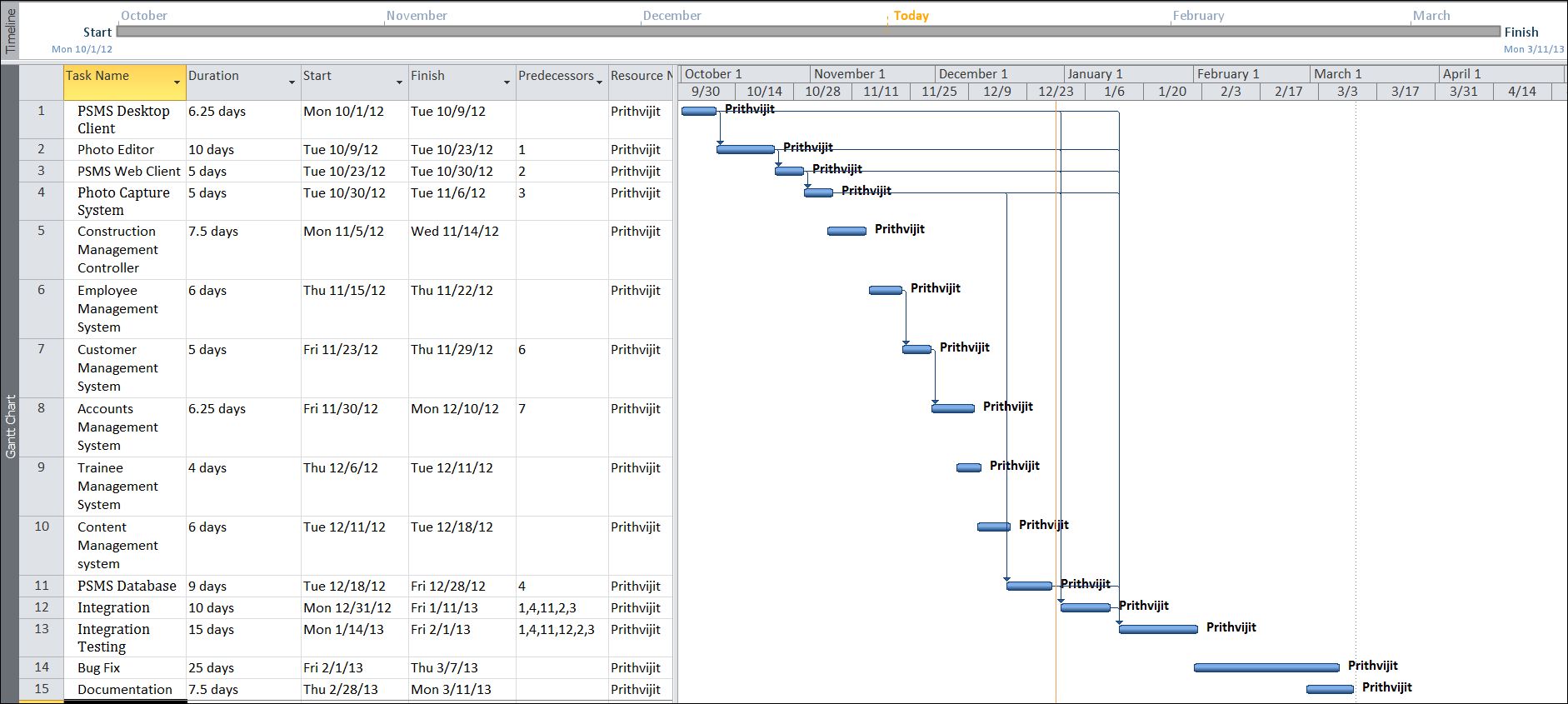
Form the very beginning I visited some studio in our locality and discussed with the studio owners; tried to list down the problems. Then I talked to software experts and took their opinion to develop the management system.

## Feasibility Study

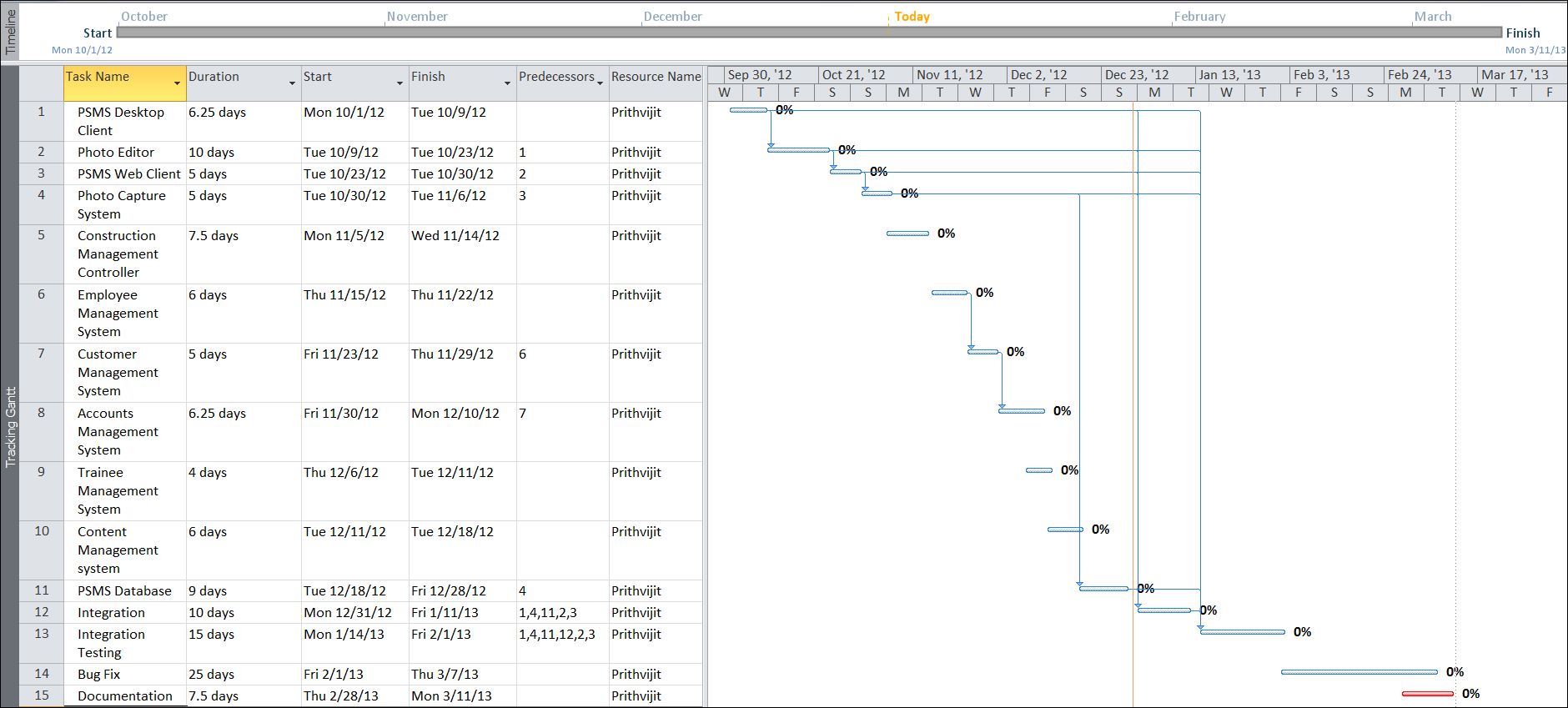
Photo studio business is growing up day by day. The owners can easily afford to buy various software to manage the studio. If we keep the cost in comfortable label with the basic feature, which help to maintain the studio, we can sell the application. And according to their need we can modify the applications and customize it for their help.

## Project Planning & Scheduling (PERT Chart and Gantt Chart both)

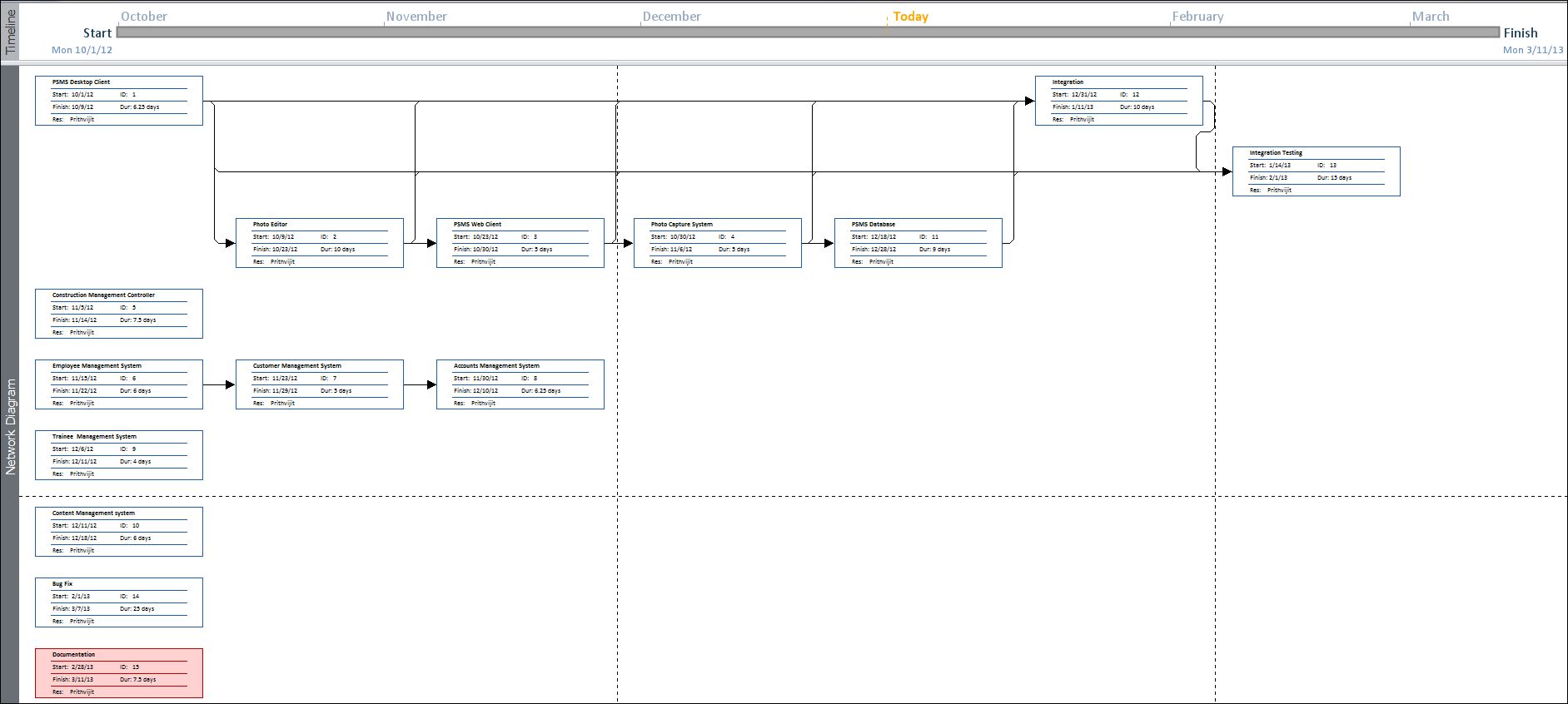
### Gantt chart

****

### Tracking Gantt

****

### Pert Chart



## Software requirement specifications (SRS)

### Functional Requirement

#### 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 detailsI for future reference.

#### Customer Requirement storing and tracking

**Introduction:**

Admin can add customer requirement in database and track them.

**Input:**

Admin will enter data provided by customer likework 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.

#### 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.

#### Edit customer photos (Brightness and contrast)

**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.

#### Edit customer photos (Rotate)

#### Bill generation

#### Daily transaction report

#### Import photo from devices

#### Show work order status

#### Generate Employee status by emp\_id

#### Accounts Status yearly

#### Add new training course

#### 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 use 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.

#### 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.

#### 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.

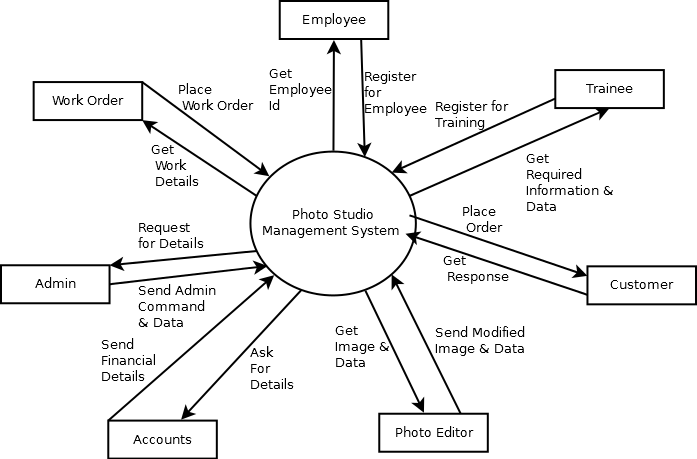
### Non-functional Requirements

* The application will be self-dependent and no dependency on other parties required.
* There will be a digital backup and restore system.
* There will be more opportunity to extend the application in future.
* The response time will be low and the system will response fast.
* It will be very user friendly and usable by any person with minimal computer knowledge.
* In terms of security unauthorized access will be denied and register user will be able to change as necessary.
* It will be efficient as it reduces manual labor and searching.
* DNBSN will have user manual and help documents.
* It is designed such a way that it can be maintained with minimal effort.

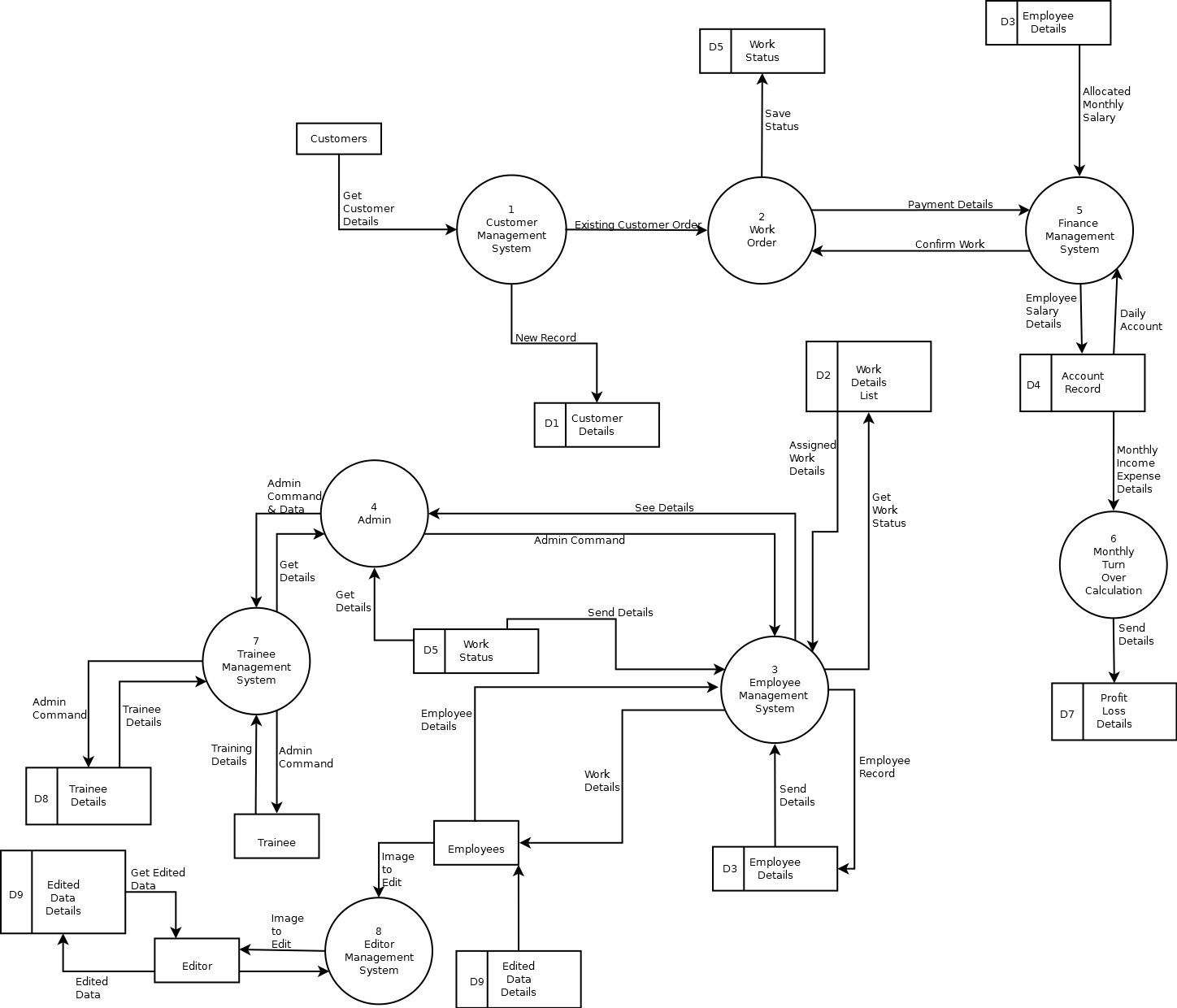
## Software Engineering Paradigm applied

## Data models

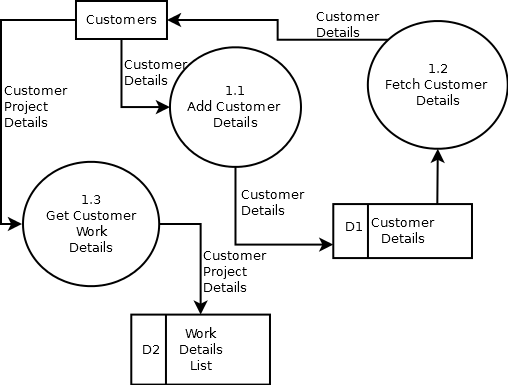
## Context Diagram

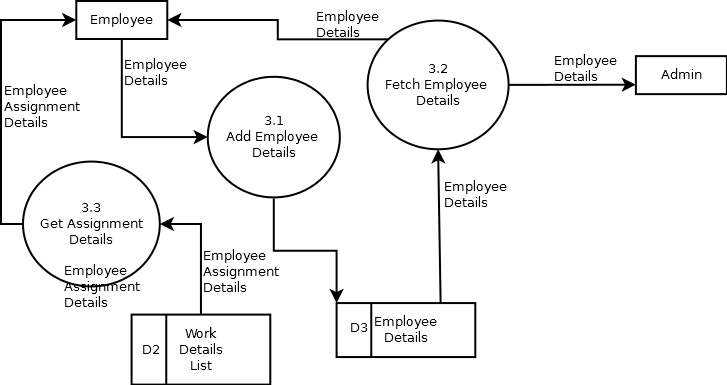


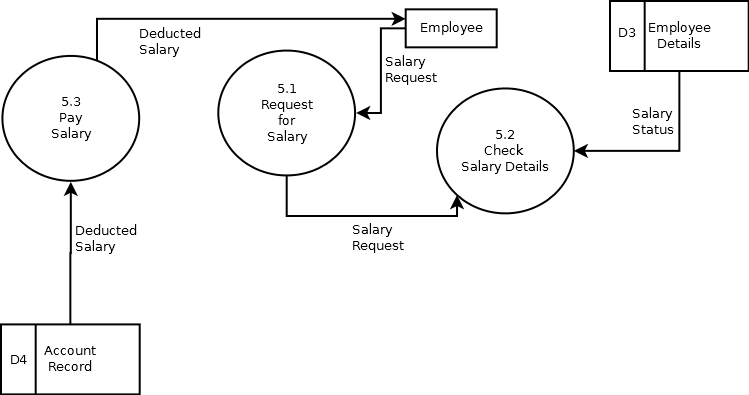
## 0-Level DFD



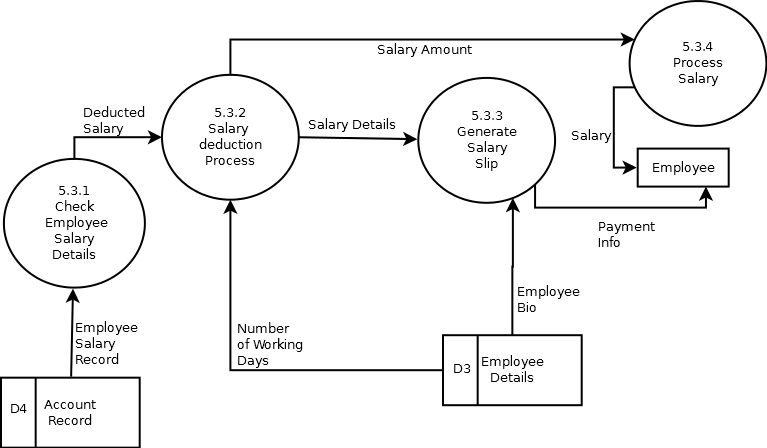
## 1-Level DFD







## 2-Level DFD



## Control Flow diagrams,

## State Diagrams/Sequence diagrams,

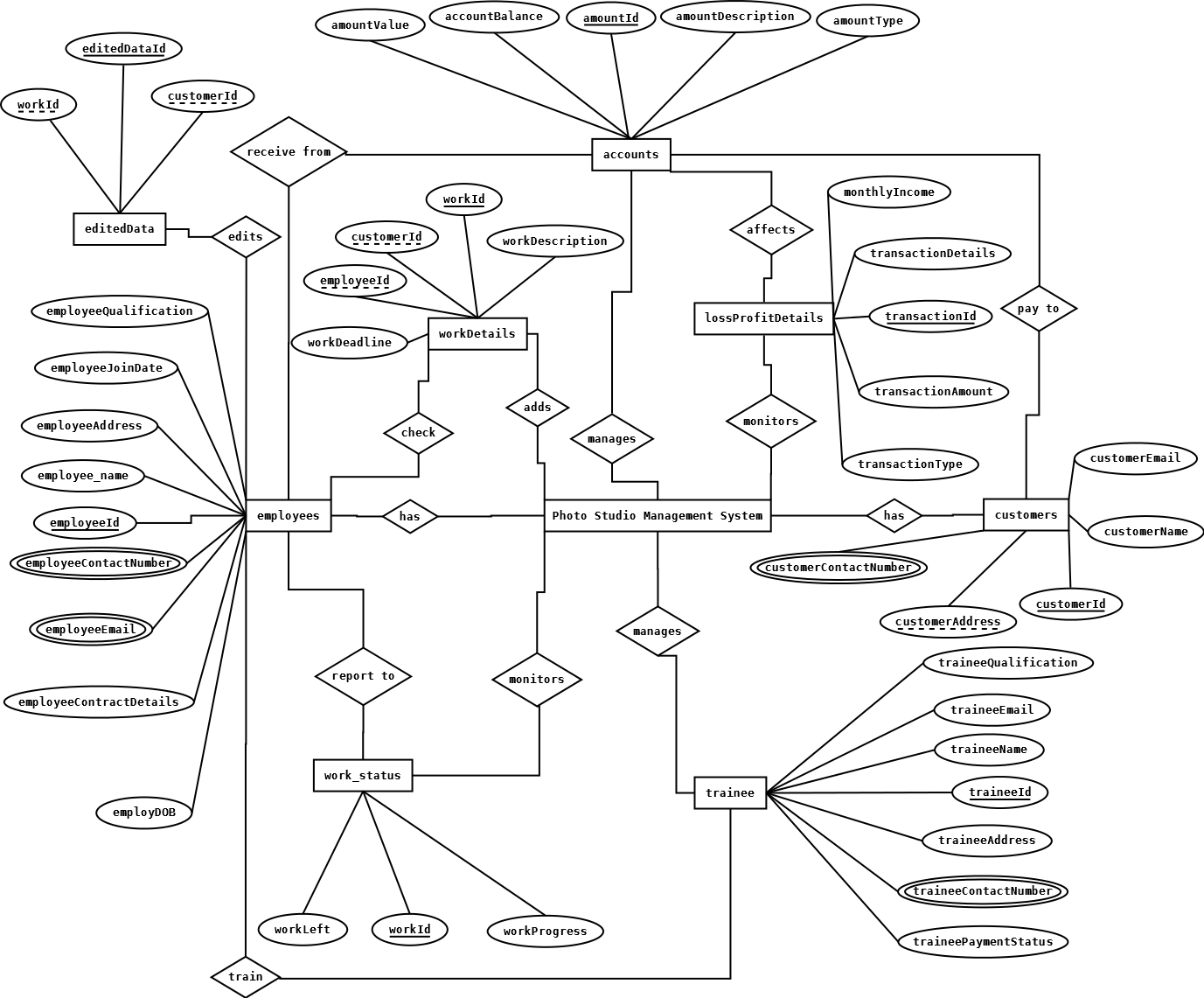
## Entity Relationship Model,

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.

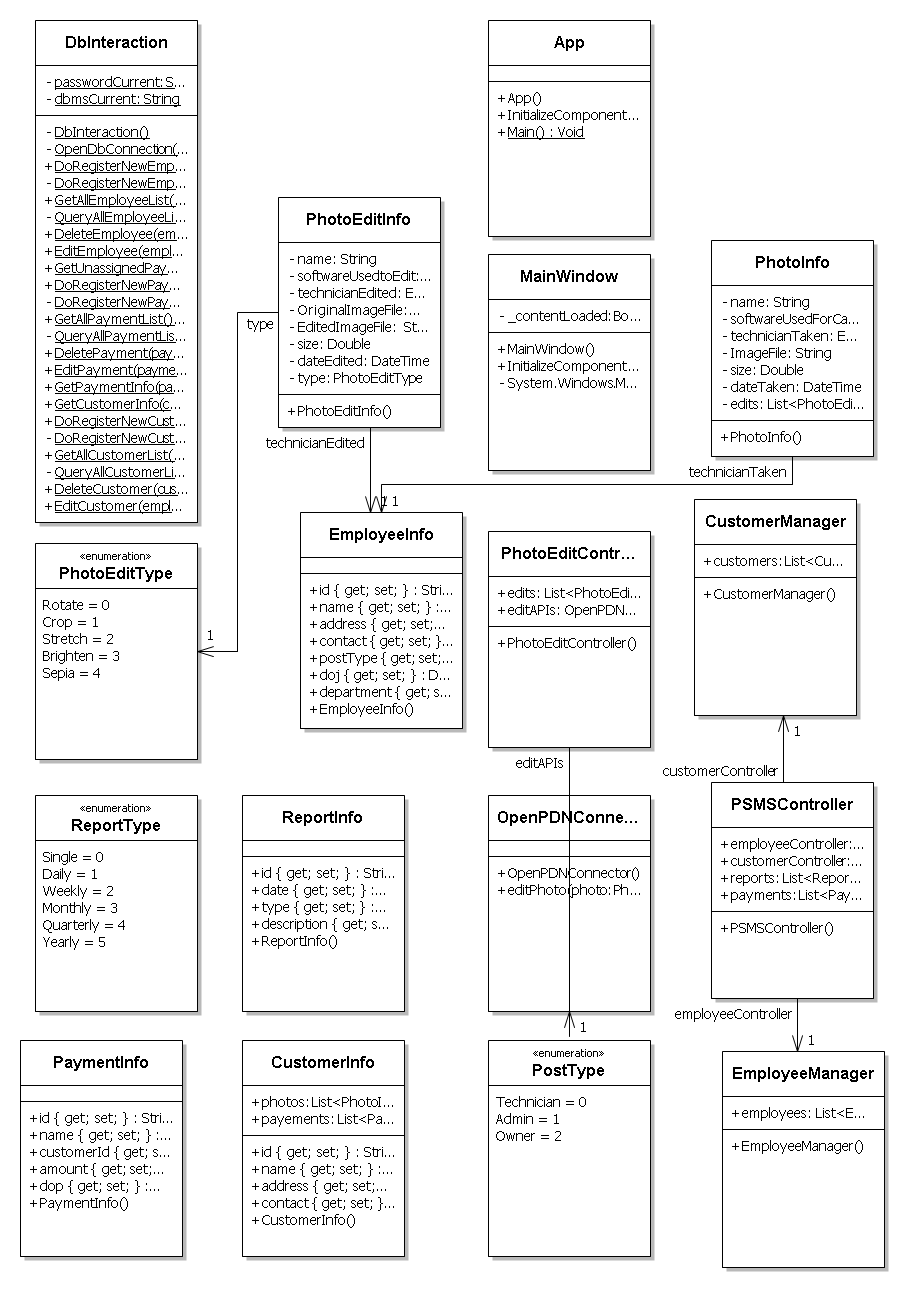
|  |  |
| --- | --- |
| **Entities** | **Attributes** |
| Customers | **customerId,** customerName**,**customerAddress, customerEmail, customerContactNumber |
| employees | **employeeId**, employeeName, employeeAddress, employeeJoinDate, employeeQualification, employeeContactNumber, employeeEmail, employeeContractDetails, 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 Diagrams/



## CRC Models/Collaboration Diagrams/Use-case Diagrams/Activity Diagrams depending upon your project requirements

# System Design

## Modularization details

## Data integrity and constraints

## Database design

The schema or database used for this application has been named ‘**psmsdb**’. Tables and corresponding keys are shown below in table form along with the screenshot of each table. Primary keys are shown in bold.

Screenshots of table structures:

Table Name: customer

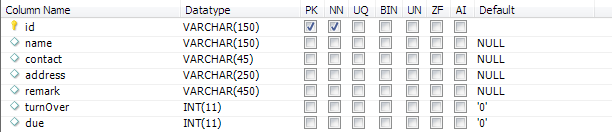


Table Name: studioinfo

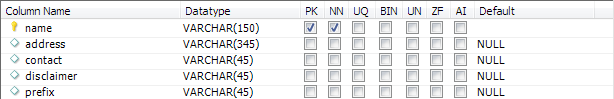


Table Name: technician

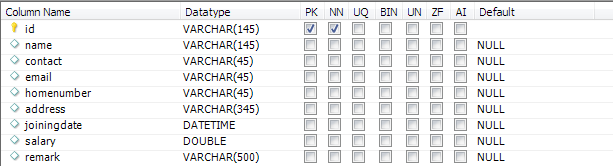


Table Name: todo

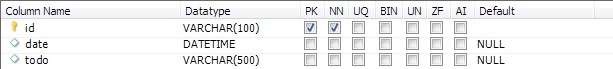
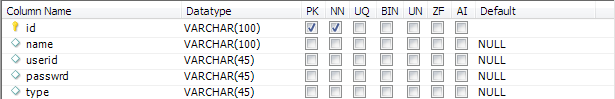


Table Name: user



|  |  |
| --- | --- |
| **Tables** | **Keys** |
| 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 |

## User Interface Design

## Test Cases (Unit Test Cases and System Test Cases)

# Coding

## Complete Project Coding

## Comments and Description of Coding segments

## Standardization of the coding

12

## Code Efficiency

## Error handling

## Parameters calling/passing

## Validation checks

# Testing

## Testing techniques and testing strategies used

## Testing Plan used

## Test reports for Unit Test Cases and System Test Cases

## Debugging and Code improvement

# System Security measures (Implementation of security for the project developed)

## Database/data security

* 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.
* The printed photos will be backed up in an encrypted format in a secure place to track and check unwanted photo printing by technicians.
* The customer photos will be kept in encrypted format so that unauthorized persons cannot access the photos.

## Creation of User profiles and access rights

The software requires a predefined username and password to login.

It allows a guest login as well which lets a guest user this application with very limited access to the user data.

# Cost Estimation of the Project along with Cost Estimation Model

The basic COCOMO estimation formula for DNBSN semidetached software:

Our Effort = 3.0 \* (3.2)1.12 PM

= 11 PM

Normal Development time = 2.5 \* (11)0.35 months

=6 months

Cost required to develop the product = Rs. 6 \* 20000

= Rs. 120,000

# Reports (sample layouts should be placed)

* Customers will be provided with a receipt for photo taken.
* List of customers can be generated
* List of employees can be generated
* List of trainee can be generated
* Photo printing details can be generated
* Account details can be generated
* Monthly/Yearly business report can be generated

# Future scope and 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.
* Integration with other photo editing software like Adobe Photoshop, Photo-Magic, GIMP will be done in Future.

# 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

# Appendices (if any)

# Glossary.