|  |
| --- |
|  |
| Electric Supply Customer Management System |
| Synopsis |
|  |
| **SANGITA MONDAL(105140503)** |
|  |

|  |
| --- |
|  |

Table of Contents

[1. Introduction 2](#_Toc320368080)

[1.1 Background 2](#_Toc320368081)

[1.2 Objective 2](#_Toc320368082)

[1.3 Purpose and Scope 3](#_Toc320368083)

[1.3.1 Purpose 3](#_Toc320368084)

[1.3.2 Scope 4](#_Toc320368085)

[2. SURVEY OF TECHNOLOGY 4](#_Toc320368086)

[Programming FRAMEWORK (.NET 4) 5](#_Toc320368087)

[Programming Language (C#) 5](#_Toc320368088)

[Database - MySQL 5](#_Toc320368089)

[3. REQUIREMENTS AND ANALYSIS 6](#_Toc320368090)

[3.1 Problem Definition 6](#_Toc320368091)

[3.1.1 Existing System 6](#_Toc320368092)

[3.1.2 Documents maintained 7](#_Toc320368093)

[3.1.3 Work To Be Done 8](#_Toc320368094)

[3.2 Requirements Specification 8](#_Toc320368095)

[3.2.1 Functional Requirements 8](#_Toc320368096)

[3.2.2 Non-Functional Requirements 11](#_Toc320368097)

[3.3 Planning and Scheduling 11](#_Toc320368098)

[Gantt chart 11](#_Toc320368099)

[Tracking Gantt 12](#_Toc320368100)

[Pert chart (Network Diagram) 12](#_Toc320368101)

[3.4 Hardware and Software Requirements 13](#_Toc320368102)

[3.4.1 Hardware Requirements 13](#_Toc320368103)

[3.4.2 Software Requirements 13](#_Toc320368104)

[3.5 PRELIMINARY PRODUCT DESCRIPTION 14](#_Toc320368105)

[3.6 CONCEPTUAL MODELS 15](#_Toc320368106)

[3.6.1 E-R Diagram 15](#_Toc320368107)

[3.6.2 Context Diagram 18](#_Toc320368108)

[3.6.3 Data Flow Diagram 18](#_Toc320368109)

[4. REFERENCES 19](#_Toc320368110)

# Introduction& Objective

## Introduction

Electricity is the ultimate need for almost every citizen of our country. With the advent of new technology we are more dependent on electricity now. Currently the electric supply offices use paper books and ledgers to track & manage customer applications, complaints. As a result, it takes longer time and extra effort to serve customers with existing inefficient system. Electric Supply Customer Management System (will be referred as **ESCMS** in this document) is a computerized solution for managing customer needs in Electric Supply offices. Electric Supply Customer Management System will enable electric supply offices to maintain computerized records and manage customer needs more efficiently with help of sophisticated customer management techniques and technologies.

## Objective

The main objective of this project is to automate the process in an Electric Supply office. Electric Supply Customer Management System will be used by Electric Supply Employees to enter various data about the Customers and their needs. The Employees will update the status of customer requests; track the progress of the work & transactions made in Electric Supply Customer Management System. It can generate reports and receipts required to serve customer request and queries. In a nutshell **ESCMS** will be the backbone of an Electric Supply office and it will be a next generation solution for better customer service and customer satisfaction.

# 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 & Modeling.

# REQUIREMENTS AND ANALYSIS

## Problem Definition

*We all know that electricity is one of the most important parts of our life. Each day, number of people using electricity is increasing quick and fast. Managing the entire system is really very complex. From employees to customers, from complaints to new connection applications, from receiving bills to paying salaries, the entire system is really hectic without proper management system. So, very fast, flexible, easy to use software is really needed to keep the entire system under control. That’s where Electric Supply Management System comes in.*

### Existing System

The existing system is traditional paper books and ledger system where several registers are maintained to store user request and to track other details about the request. The flow diagram of how a customer request executed now is shown below:



### Documents maintained

**Application Register:** Application Number, Name, Address with Contact Number, Load, Initial

Deposit Amount, Application Received Date, Quotation Amount, Quotation Sent Date, Amount

Received On, Service Connection Number.



* **Service Connection Register:** Service Connection Number, Application Number, Name,Address with Contact Number, Quotation Amount, Amount Received Date, Work Assigned To (Contractor), Work Completed on Date.
* **Meter Movement Register:** Service Connection Number, Name Address with Contact Number, Meter Number, Seal Number, Meter Issue Date, Work Assigned To (Staff), Date of Connection
* **Estimation Sheet per Application:** contains input Application Number, name, Address with Contact Number, Wire Length Required, Angle Type and calculates Weight of Angle, and Quotation amount

### Work To Be Done

We will incorporate the above mentioned workflow of an Electric Supply Customer Management System in an automatic computerized way.

## Requirements Specification

### Functional Requirements

#### Apply for new connection

**Introduction**

Customer can apply for a new connection.

**Inputs**

Relevant customer data like name, address, contact number, type, payment.

**Processing**

Employee will enter the data in the ESCMS and create a new connection entry.

**Outputs**

ESCMS will generate an application number for future reference and will provide customer a acknowledgement receipt.

#### check connection request status

**Introduction**

Customer can check the new connection status.

**Inputs**

Application number & customer name.

**Processing**

Employee will enter application number & customer name in the ESCMS and it will search the status & display in the screen.

**Outputs**

Customer will get the status information from employee and he may request for a printed status also.

#### Create a vendor task

**Introduction**

Employee will create a task for vendor.

**Inputs**

Application number, customer details, task details.

**Processing**

Employee will enter details in the ESCMS and it will pick a vendor & assign the task.

**Outputs**

Vendor will get a notification about the task and a printed work order.

#### vendor task status update

**Introduction**

Vendor will update the task status to employee and receive partial payment.

**Inputs**

Application number, task details, proof of task status.

**Processing**

Employee will enter details in the ESCMS and update the system. System will approve the payment order.

**Outputs**

Vendor will get a notification about the task update and a printed payment order.

#### Generate report

**Introduction**

Employee will choose the kind report to be printed and system will create the details of the report and print it.

**Inputs**

Report Type, area, time frame.

**Processing**

Employee will enter details in the ESCMS and the system will collate data. System will print the report.

**Outputs**

A printed report will generated.

### Non-Functional Requirements

* **Efficiency**:

It will be efficient as it reduces manual labor and searching.

* **Backup**:

The employees will take regular print out of daily reports and take back up. Digital back up can be taken in a regular interval.

* **Documentation**:

ESCMS will have user manual and help documents.

* **Maintainability**:

It is designed such a way that it can be maintained with minimal effort.

* **Performance**:

The response time of ESCMS will be very fast. So it will be efficient enough to cater the customer.

* **Privacy**:

The data will be encrypted and the user data will not be shared with third party.

* **Security**:

ESCMS will use secure connection and enhanced security measures to protect data.

* **Usability**:

It will be very user friendly and usable by any person with minimal computer knowledge.

### 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

## Planning and Scheduling

### Gantt chart



### Tracking Gantt



### Pert chart (Network Diagram)



# Scope of the Solution

The purpose of the project is to computerize the workflow of an Electric supply office. The below mentioned diagram shows the existing Customer Management System of Electric Supply offices which will be completely computerized with the help of this software.



The main advantages of Electric Supply Customer Management System over existing traditional paper book and ledger system are given below:

1. It is faster as the computer is doing the searching and fetching of data.

2. It is more efficient as it reduces the need of more employees.

3. It is more accurate.

4. It can generate paper report and main ledgers to help transition from older system.

5. It can store the data in a centralized server so that data will available to use anywhere. So the auditing of the data can be done without physically being present to the site.

Currently this software is aimed for a single electric supply office customer management. It can be extended to support networked multiple electric supply office and have a centralized database and to serve wider range of customers of Electric Supply around the country.

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 Electric Billing System right now. But in future we can easily extend to support that.

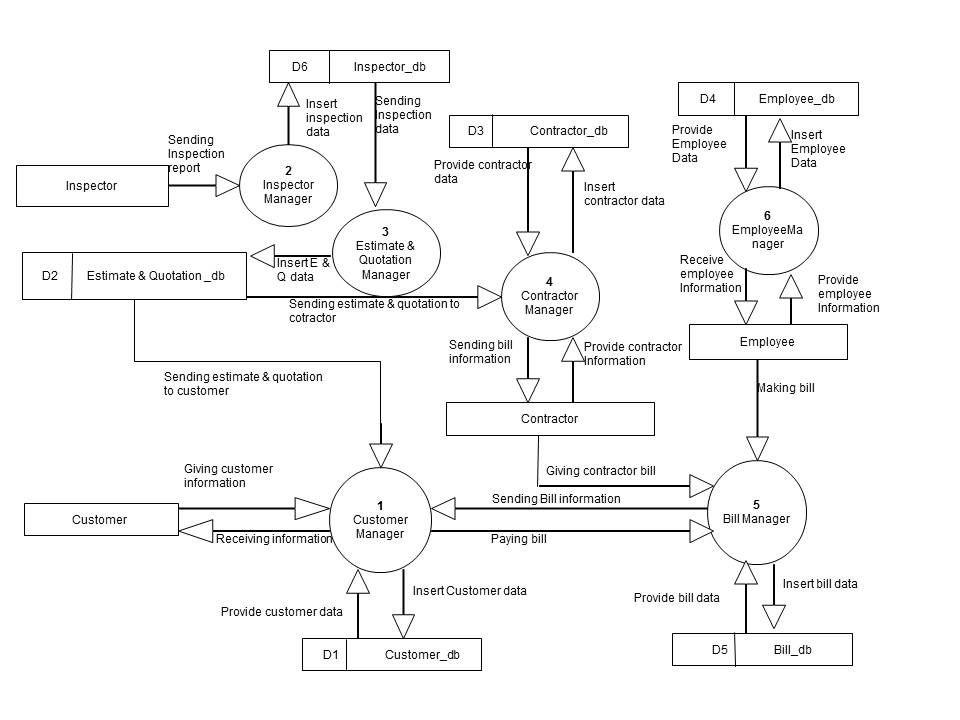
ESCMS will not only solve the complexity of the electric supply management but also speed up the process of a new connection. It is helpful to both users and customers. With its simple to understand GUI, ESCMS helps in managing the entire process more easily, efficiently and accurately for the electric office employees. On the other hand, it helps customers serve better.

# Analysis

## Context Diagram



## 0-Level DFD



## 1-Level DFD



## 2-Level DFD



## E-R Diagram

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



**Relationship between Entities:**

Electric Supply office has Customers1 : N

Electric Supply office has Contractors1 : N

Electric Supply office has Employees1 : N

Customer does Requests 1 : N

Electric Supply serves Requests 1 : N

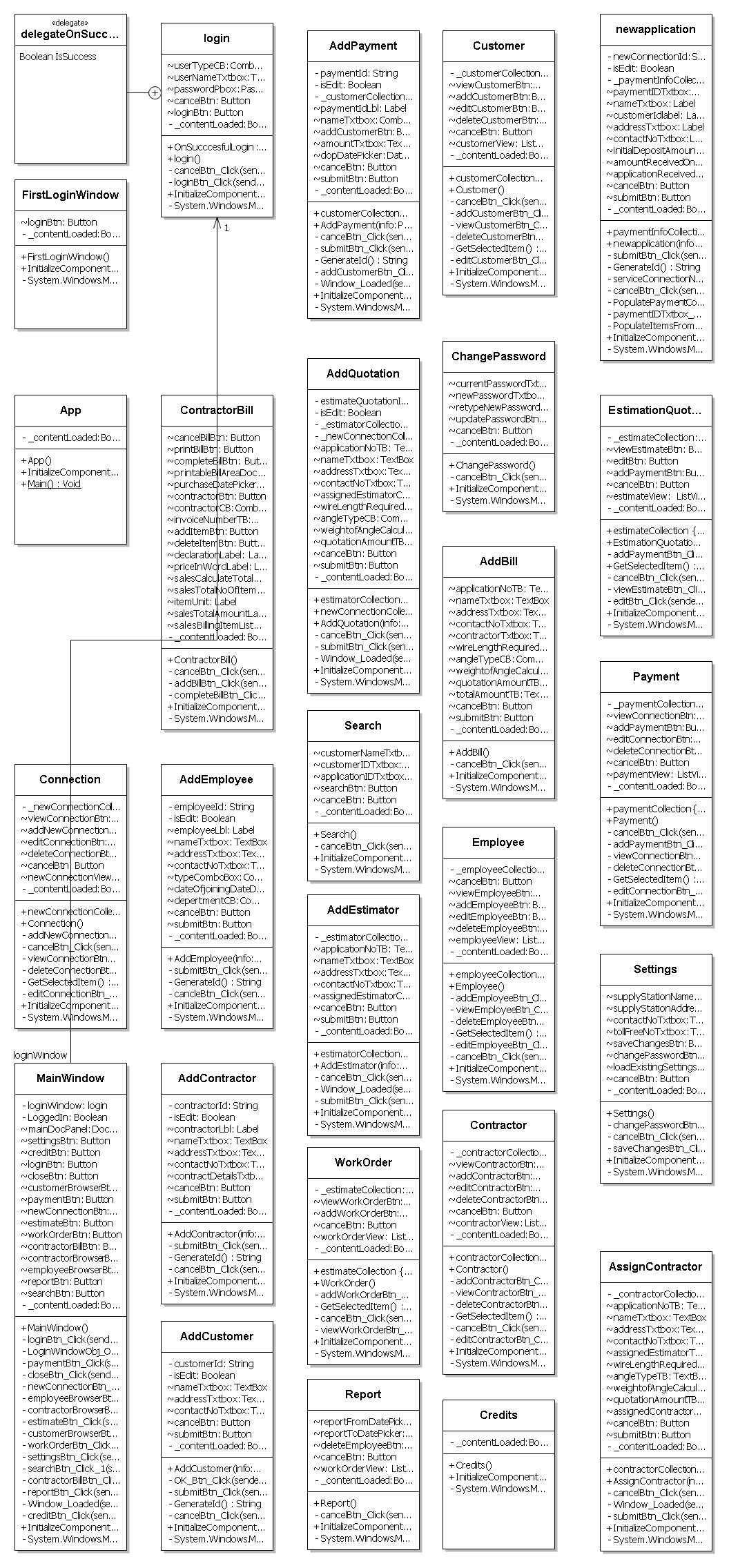
User uses Service Connection 1 : N

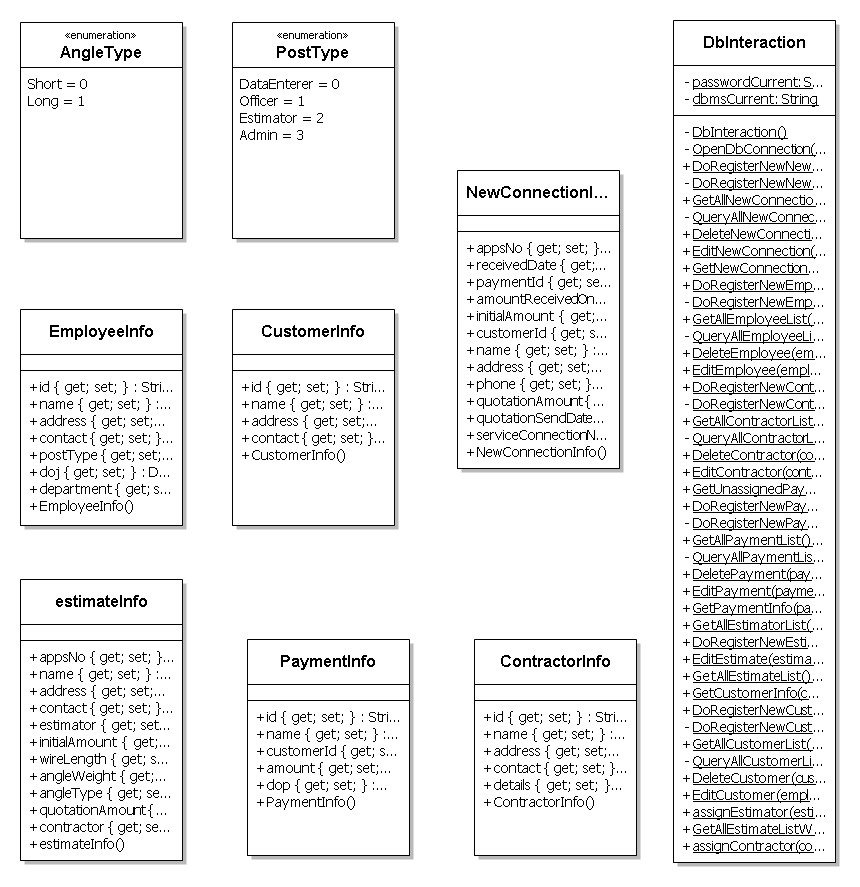
Employees provides Estimates M : N





## Class Diagram





# Database & Table Details

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

|  |  |  |
| --- | --- | --- |
| **APPLICATION\_REGISTER** | |  |
| **Column Name** | **Datatype** | **Default** |
| apps\_no | string | NOT NULL |
| payment\_id | string | NULL |
| customerId | string | NULL |
| received\_date | DATETIME | NULL |
| estimateId | DOUBLE | NULL |
| service\_connection\_no | string | NULL |
|  |  |  |
|  |  |  |
| **CONTRUCTOR** |  |  |
| **Column Name** | **Datatype** | **Default** |
| Id | string | NOT NULL |
| Name | string | NULL |
| Address | string | NULL |
| Contact | string | NULL |
| contract\_details | string | NULL |
|  |  |  |
|  |  |  |
| **CUSTOMER** |  |  |
| **Column Name** | **Datatype** | **Default** |
| Id | string | NOT NULL |
| Name | string | NULL |
| Address | string | NULL |
| Contact | string | NULL |
|  |  |  |
|  |  |  |
| **EMPLOYEE** |  |  |
| **Column Name** | **Datatype** | **default** |
| Id | string | NOT NULL |
| Name | string | NULL |
| Address | string | NULL |
| Contact | string | NULL |
| post\_type | string | NULL |
| Doj | DATETIME | NULL |
| Department | string | NULL |
|  |  |  |
|  |  |  |
| **ESTIMATE** |  |  |
| **Column Name** | **Datatype** | **Default** |
| appNo | string | NOT NULL |
| Estimator | string | TBA |
| wireLength | DOUBLE | 0 |
| angleType | string | SHORT |
| angleWeight | DOUBLE | 0 |
| amountQuotation | DOUBLE | 0 |
| Contractor | string | TBA |
|  |  |  |
|  |  |  |
| **METER\_GER** |  |  |
| **Column Name** | **Datatype** | **Default** |
| Id | string | NOT NULL |
| connection\_no | string | NULL |
| connection\_name | string | NULL |
| Address | string | NULL |
| Contact | string | NULL |
| meter\_no | string | NULL |
| seal\_no | string | NULL |
| issue\_date | DATETIME | NULL |
| work\_assign\_to | string | NULL |
|  |  |  |
|  |  |  |
| **METER\_REGISTER** |  |  |
| **Column Name** | **Datatype** | **Default** |
| Id | string | NOT NULL |
| connection\_no | string | NULL |
| connection\_name | string | NULL |
| Address | string | NULL |
| Contact | string | NULL |
| meter\_no | string | NULL |
| seal\_no | string | NULL |
| issue\_date | DATETIME | NULL |
| work\_assign\_to | string | NULL |
|  |  |  |
|  |  |  |
| **PAYMENT** |  |  |
| **Column Name** | **Datatype** | **Default** |
| Id | string | NOT NULL |
| customerId | string | NULL |
| Amount | DOUBLE | NULL |
| Dop | DATETIME | NULL |
|  |  |  |
|  |  |  |
| **SERVICE\_CONNECTION\_REGISTER** | |  |
| **Column Name** | **Datatype** | **Default** |
| Id | string | NOT NULL |
| apps\_no | string | NULL |
| Name | string | NULL |
| Address | string | NULL |
| Contact | string | NULL |
| wire\_length | string | NULL |
| initial\_deposite\_amount | DOUBLE | NULL |
| angle\_type | string | NULL |
| angle\_weight | string | NULL |
| quotation\_amount | DOUBLE | NULL |

# Complete Structure

## Module Description

Electric Supply Customer Management System will upgrade the existing system, so it needs to have good support for existing system as well. It will collect the data from customers & employees and populate records which will match existing paper book registers. So that employees can take printout and maintain similar records. Electric Supply Customer Management System consists of three main modules:

* ESCMS GUI
* ESCMS Engine
* ESCMS Database



*ECMS contains following main modules:*

* ECMS GUI: all the codes containing WPF GUI designing are written in this module. It contains many sub modules such as:

1. Connection: all the codes of Connection window and its sub windows are written in this module.
2. Contractor: all the codes of contractor window and its sub windows are written in this module.
3. Contractor Bill: This module contains the designing codes of the contractor bill.
4. Customer: Contains codes related to customer window such as Customer details window, adding new customer, updating and deleting them.
5. Employee: All the codes of the employee browser are separated in this module.

* ECMS Controller: all the logics of the entire application are written in this module. The ECMS Controller module controls the logical data flow direction of the entire application, such as what is going to take place when we click on the ‘view’ button in inside the employee browser or how the application assigns a new contractor for a new connection etc.
* ECMS Storage: All the logics related to data storage are written in this module. This module is controlled by ECMS Controller for various database related actions.
* ECMS Style: This module contains all the style definitions of various GUI tool such as buttons, textboxes etc. The style definitions written in this module are used by various sub modules of ECMS GUI whenever the style for a tool is to be defined.

## estimation

## Data Structure

|  |
| --- |
| **ContractorInfo** |
| public class ContractorInfo  {  public string id { get; set; }  public string name { get; set; }  public string address { get; set; }  public string contact { get; set; }  public string details { get; set; }  } |

|  |
| --- |
| **CustomerInfo** |
| public class CustomerInfo  {  public string id { get; set; }  public string name { get; set; }  public string address { get; set; }  public string contact { get; set; }  } |

|  |
| --- |
| **EmployeeInfo** |
| public class EmployeeInfo  {  public string id { get; set; }  public string name { get; set; }  public string address { get; set; }  public string contact { get; set; }  public PostType postType { get; set; }  public DateTime doj { get; set; }  public string department { get; set; }  } |

|  |
| --- |
| **PostType** |
| public enum PostType  {  DataEnterer,  Officer,  Estimator,  Admin  } |

|  |
| --- |
| **AngleType** |
| public enum AngleType  {  Short,  Long  } |

|  |
| --- |
| **estimateInfo** |
| public class estimateInfo  {  public string appsNo { get; set; }  public string name { get; set; }  public string address { get; set; }  public string contact { get; set; }  public string estimator { get; set; }  public double initialAmount { get; set; }  public double wireLength { get; set; }  public double angleWeight { get; set; }  public AngleType angleType { get; set; }  public double quotationAmount { get; set; }  public string contractor { get; set; }  } |

|  |
| --- |
| **NewConnectionInfo** |
| public class NewConnectionInfo  {  public string appsNo { get; set; }  public DateTime receivedDate { get; set; }  public string paymentId { get; set; }  public DateTime amountReceivedOn { get; set; }  public double initialAmount { get; set; }  public string customerId { get; set; }  public string name { get; set; }  public string address { get; set; }  public string phone { get; set; }    public double quotationAmount { get; set; }  public DateTime quotationSendDate { get; set; }  public string serviceConnectionNo { get; set; }  } |

|  |
| --- |
| **PaymentInfo** |
| public class PaymentInfo  {  public string id { get; set; }  public string name { get; set; }  public string customerId { get; set; }  public double amount { get; set; }  public DateTime dop { get; set; }  } |

## Implementation Methodology

* Object Oriented Programming methodology will be adopted and C# 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 WPF (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.

## List of Reports

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

* Locality wise connection report can be generated
* List of New connections can be generated
* List of Customers can be generated
* List of Vendors can be generated
* Connection details can be generated
* Fund details can be generated
* Yearly report can be generated
* Salary slips can be created

# Implementation of Security Mechanism at Various Levels

* This software requires a valid password to login and then it allows using any of its features. It allows admin to create various types of accounts with different permission levels like clerk, officer, contractor etc. so that user can see relevant data only.
* The login password will be saved in encrypted format in database.
* This software will use Google open-id authentication for web interface.

# Future Scope & Further Enhancement of the Project

It does not allow a user to pay bill via automated transaction machines. In the modern age this feature might rally be useful for the convenience of the customers.

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

# 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](http://www.w3schools.com/)
* [www.mysql.org](http://www.mysql.org)
* Electric Supply Professionals
* **Programming C#** - E. R. Balaguruswamy