|  |
| --- |
|  |
| School Management System |
| Synopsis |
|  |
| **Susmita Podder( 105140695 )** |
|  |

|  |
| --- |
|  |

Table of Contents

[Introduction & OBJECTIve 3](#_Toc344538733)

[INTRODUCTION 3](#_Toc344538734)

[Objective 3](#_Toc344538735)

[PROJECT CATEGORY 3](#_Toc344538736)

[Hardware and Software Specification 3](#_Toc344538737)

[Hardware Requirement 4](#_Toc344538738)

[Software Requirement 4](#_Toc344538739)

[REQUIREMENTS AND ANALYSIS 4](#_Toc344538740)

[Problem Definition 4](#_Toc344538741)

[Existing System 4](#_Toc344538742)

[Documents maintained 6](#_Toc344538743)

[Work To Be Done 7](#_Toc344538744)

[Requirements Specification 7](#_Toc344538745)

[Functional Requirements 7](#_Toc344538746)

[technical specification 10](#_Toc344538747)

[Planning and Scheduling 10](#_Toc344538748)

[Gantt chart 10](#_Toc344538749)

[Tracking Gantt 11](#_Toc344538750)

[Pert chart (Network Diagram) 12](#_Toc344538751)

[Scope of the Solution 12](#_Toc344538752)

[Analysis 13](#_Toc344538753)

[Context Diagram 13](#_Toc344538754)

[Data Flow Diagram 13](#_Toc344538755)

[Level 0 DFD 13](#_Toc344538756)

[Level 1 DFD 14](#_Toc344538757)

[Level 2 DFD 16](#_Toc344538758)

[E-R Diagram 16](#_Toc344538759)

[Class Diagram 20](#_Toc344538760)

[Database & Table Details 20](#_Toc344538761)

[COMPLETE DATA STRUCTURE 22](#_Toc344538762)

[Module Description 22](#_Toc344538763)

[School Management Server 22](#_Toc344538764)

[School Management Client 23](#_Toc344538765)

[School Management Database 23](#_Toc344538766)

[estimation 23](#_Toc344538767)

[9.2 Data Structure 24](#_Toc344538768)

[9.4 Implementation Methodology 27](#_Toc344538769)

[9.5 List of Reports 27](#_Toc344538770)

[sECURITY MECHANISM 27](#_Toc344538771)

[FUTURE SCOPE AND FURTHER REQUIREMENTS 28](#_Toc344538772)

[1. bIBLIOGRAPHY 28](#_Toc344538773)

# Introduction & OBJECTIve

## INTRODUCTION

The population of our country is increasing rapidly, but the resources for providing proper education to the children are limited. So we need to utilize our existing school management processes properly with the help of digital technologies.

Most of the schools are managed by a single authority like head master or a group of people like governing body. But in this fast paced world people tend to switch jobs. If the key people leave the school then the quality of education and school management deteriorates. There are so many dependencies on the school management personnel. We can eradicate this dependency by deploying a computerized solution for managing school.

Nowadays both of the parents are employed for most kids and they don’t have enough time to interact with teachers frequently. We need a computerized system to manage interaction between parents and teachers.

We will implement a school management system to address these issues and bring up an efficient system to manage activities of a school using single software.

## Objective

School Management System is versatile and complete end-to-end school management software .School Management System is used to enhance the administrative efficiency of educational institutions. It is an interactive platform for all entities viz. Students, Teachers, Management, Parents. It is a simple yet powerful one point integrated platform that connects all the departments of an institution namely office, fee counter, library, hostel, stores, academics, activity center and so on.

# PROJECT CATEGORY

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

OOP Language: Java

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)

* Microsoft .NET 4.0

# 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 student and teacher. The flow diagram of how a student takes admission in Schools is shown below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Student must take a registration form from the school with a small amount as a cost of registration form. |  |  | | --- | | Fill up the form by the student and submit to the School. |  |  | | --- | | According to the submitted registration form, an interview will be arranged. |  |  | | --- | | If the student passed the interview then he/she can take admission in the school with a admission fees. |  |  | | --- | | He/ She will get study materials from the School. |  |  | | --- | | He / She must pay the overall Tuition fees over the Year. |  |  | | --- | | He / She can join the Class. | |

The flow diagram of how a Student is promoted to next class according to his / her Results is shown below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Checks the attendance of the Student whether he/she has 75% attendance. |  |  | | --- | | If he/she has , an admit card will be issued |  |  | | --- | | He / She can give the exam. |  |  | | --- | | Records are stored in the system. |  |  | | --- | | Checks the getting numbers in each subject whether it is more than 40%. |  |  | | --- | | If yes, then He / She will be promoted to the next class. |  |  | | --- | | Mark sheet given. | |

### Documents maintained

* **Admission Register**: Form Number, Student Name, Address with Contact Number, Mother’s Name, Father’s Name, Parent’s Income per annum, Parent’s qualification, Initial amount for registration, Form Submission Date.
* **Enrolment Register:** Form Number, Student Name, Address with Contact Number, Parent’s name, Deposit Amount, Amount Received Date, Student Assigned to (Class).
* **Examination Register:** Enrolment Number, Student Name, Address with Contact Number, Attendance, Class Performance, Deposit Amount check, Received Amount date.
* **Grade card generation:** Enrolment Number, Student name, Address with Contact Number, Getting marks, Grand total, Percentage Marks, Grade given, Position given.

### Work To Be Done

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

## Requirements Specification

### Functional Requirements

#### View and Enter new Student and Employee information

**Introduction**

The details of a new student are stored into a student profile. Only School administration department can enter that details of new student but and teachers can only updated the existing student status. Details of employee are also kept into individual employee profile and it will also be updated by School Administration System.

**Input**

Relevant student and employee data like name, address, contact no., applying for which class.

**Processing**

Employee will enter data in SMS and create a new Student enrolment no, as well as a new code number for Employee.

**Output**

SMS will generate Enrolment no. for Student and Code no. for Employee .Details can be viewed later on whenever required.

#### View and Enter new timetable information

**Introduction**

Employee can view the time table and can also update the timetable information.

**Input**

Employee number, department number and Week range

**Processing**

Employee number and department number must be unique, and when entering timetable of any employee or teacher both values must be valid references. Week range must be between 1to 52.

**Output**

Teacher and Student both can see the time table.

#### Security

**Introduction**

Only the high level members of the School and Network manager will have access to the system for securing their important data from others.

**Input**

System username and password

**Processing**

The network operating system in the department will be used to enforce security. Another security level should also be incorporated to make the system more secure.

**Output**

All data are secured and that can be used in future.

#### Changing Password and Username

**Introduction**

Change existing username and password

**Input**

New username and password

**Processing**

Old username and password will be replaced by user provided new username and password after authenticating.

**Output**

Password and Username can be changed according to the Employee requirement whenever they want to change for better security of the System.

#### Mail Notification

**Introduction**

If holiday is declared suddenly, all students, teachers and employees are informed by sending them a mail.

**Input**

Student and Teacher’s name and email id.

**Processing**

Employee will enter the name, email id and reason of holidays in the SMS and it will generate a message.

**Output**

Employee and Students get a message from SMS.

#### Exam Grade Details

**Introduction**

Data sheets are prepared for individual class. And each datasheet is given via email to the student of the corresponding class.

**Input**

Student name, marks in individual subject, attendance, class performance**.**

**Processing**

Grade card can be generated for individual students. Exam administrators would need to be able to view, update, delete, print and add grade details.

**Output**

A printed Grade card can be given to the student as well as an email can be received by the student with Grade card.

#### Fees Details

**Introduction**

Fees details of all Students are kept and they are reminded after every 4 days after last date. Late Fee is also charged after last date.

**Input**

Student name, enrolment no, remaining fees, last date of deposit amount.

**Processing**

SMS automatically generates a message and send it to the student email id.

**Output**

Students are reminded after every 4 days after last date.

### 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

Currently this software is aimed for a single school management. It can be extended to support networked multiple school and have a centralized database and to serve wider range of students of different branches of same School 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 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

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

|  |  |
| --- | --- |
| **Entities** | **Attributes** |
| Student | **Student\_ID**, Student\_DOB, Student\_Name, Student\_Parent\_Name, Student\_Address, Student\_Admission\_Date, Student\_Course\_Name, Student\_Contact |
| Account | **Transaction\_ID**, Transaction\_Amount, Transaction\_Type, Transaction\_Reason,Account\_Balance |
| Admin | **Staff\_ID**, Staff\_Name, Staff\_Permission\_Level, Staff\_Address, Staff\_Admission\_Date, Staff\_Course\_Name, Staff\_Contract\_details, Staff\_Join\_Date, Staff\_Email, Staff\_Role |
| Books | **Book\_ID**, Book\_Name, Book\_Author, Purchase\_Date, Book\_Status, Book\_Description,Purchase\_Amount |
| Faculty | **Faculty\_ID**, Faculty\_Name, Faculty\_Address, Faculty\_Join\_Date, Faculty\_Course\_Under, Faculty\_Contact\_Details, Faculty\_Salary\_Details |
| Course | **Course\_ID**, Course\_Faculty, Course\_Name, Required\_Qualification, Course\_Fees, Course\_Admission\_Date, Students\_Under, Course\_Description |

**Relationship between Entities:**

School Management System has Courses 1 : N

School Management System has Students 1 : N

School Management System has Faculties 1 : N

School Management System has Admin 1 : 1

Student has Attendance 1 : 1

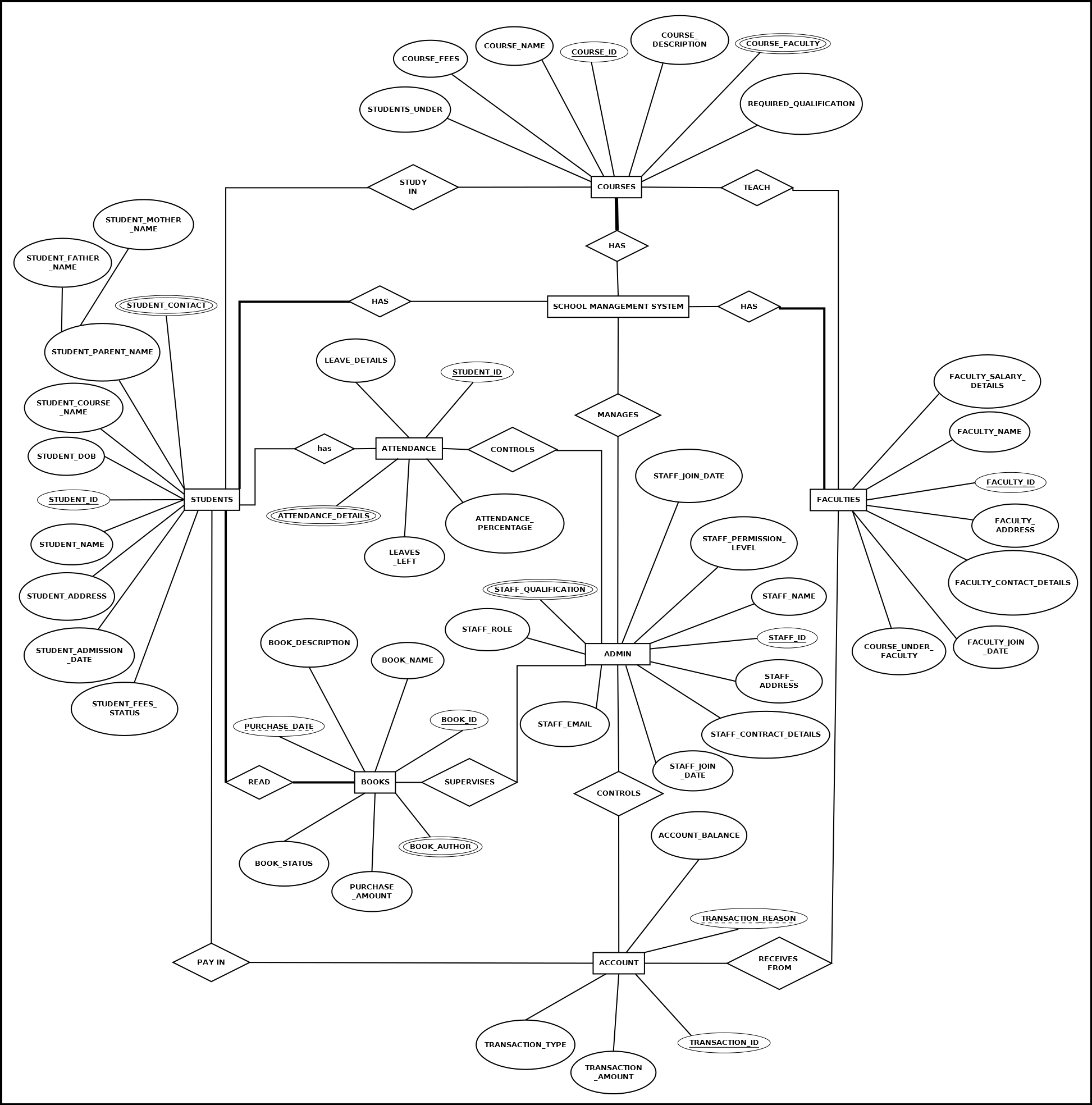
Admin Checks Attendance 1 : 1

Students reads Books M : N

Students pays Account 1 : 1

Admin Controls Account 1 : 1

Students Studies in Course N:1



## Class Diagram



# Database & Table Details

The database used for this software is called **smsdb**. 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



School Management System is divided three main modules such as:

1. School Management Server
2. School Management Client
3. School Management Database

### School Management Server

School Management server is a singleton server designed provide services for school management system. It controls various activities required for the school management system. To manage these activities it has several sub modules such as:

1. Admission Management
2. Student Management
3. Faculty Management
4. Course Management
5. Attendance & Leave Management
6. Library Management
7. Accounts Management
8. Administration Management

### School Management Client

School Management System will provide two different clients for the convenience of the user. Desktop client is for doing bulk activities and faster tasks. Web client will allow instant access from anywhere and anytime.

### School Management Database

School Management 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

|  |
| --- |
| **StudentInfo** |
| public class StudentInfo  {  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 admissionDate { get; set; }  public DateTime dob { get; set; }  } |

|  |
| --- |
| **CourseInfo** |
| public class CourseInfo  {  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; }  } |

|  |
| --- |
| **FacultyInfo** |
| public class FacultyInfo  {  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; }  public List<string> qualifications { get; set; }    public List<string> courses { get; set; }    } |

|  |
| --- |
| **AdmissionInfo** |
| public class AdmissionInfo  {  public string id { get; set; }  public string name { get; set; }  public string purpose { get; set; }  public int ageLimit { get; set; }  public string Venue { get; set; }  public string coordinatorPhone { get; set; }  public string admittedAddress { get; set; }  public DateTime examDate { get; set; }  public DateTime resultDate { get; set; }  public List<string> subjects { get; set; }  public int numberOfSeats { get; set; }  } |

|  |
| --- |
| **BookInfo** |
| public class BookInfo  {  public string id { get; set; }  public DateTime issueDate { get; set; }  public List<string> author { get; set; }  public List<string> tag { get; set; }  } |

|  |
| --- |
| **TransactionInfo** |
| public class TransactionInfo  {  public string id { get; set; }  public string name { get; set; }  public double amount { get; set; }  public string type { get; set; }  public string reason { get; set; }  public string receivedFrom { get; set; }  public string receivedBy { get; set; }  public DateTime date { get; set; }  } |

## 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 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:

* Results & Grade report can be generated
* Class toppers will be given a certificate as a token of appreciation
* List of students can be generated
* List of teachers can be generated
* List of courses can be generated
* Course details can be generated
* Fund details can be generated
* Yearly donation report can be generated
* Salary slips can be created

# sECURITY MECHANISM

* School Management System is password protected software. It will be developed such a way that the admin will have complete control on the school’s data.
* Admin can create account with various permission levels, like clerk, librarian, teacher, admin etc. so that the users can see relevant data only.
* The data of the school 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.

# FUTURE SCOPE AND FURTHER REQUIREMENTS

* Mobile application could be developed for students’ guardians for querying about various details.
* Support for Linux operating system could be added.
* Online result checking and fees payment feature could be added.

# 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)
* School Professionals
* **Programming Java** - E. R. Balaguruswamy