**2013**

**SUSMITA PODDER**

**1/19/2013**



SCHOOL MANAGEMENT SYSTEM

Table of Contents

[INTRODUCTION AND OBJECTIVE 4](#_Toc346626317)

[INTRODUCTION 4](#_Toc346626318)

[OBJECTIVE 4](#_Toc346626319)

[SYSTEM ANALYSIS 4](#_Toc346626320)

[IDENTIFICATION OF NEED 5](#_Toc346626321)

[PRELIMINARY INVESTIGATION 5](#_Toc346626322)

[FEASIBILITY STUDY 5](#_Toc346626323)

[PROJECT PLANNING 5](#_Toc346626324)

[TRACKING GANTT 5](#_Toc346626325)

[PROJECT SCHEDULING 5](#_Toc346626326)

[PERT CHART (NETWORK DIAGRAM) 5](#_Toc346626327)

[GANTT CHART 6](#_Toc346626328)

[SOFTWARE REQUIREMENT AND SPECIFICATION 7](#_Toc346626329)

[FUNCTIONAL REQUIREMENTS 7](#_Toc346626330)

[TECHNICAL SPECIFICATION 10](#_Toc346626331)

[SOFTWARE ENGINEERING PARADIGM APPLIED 10](#_Toc346626332)

[DATA MODELS 10](#_Toc346626333)

[CONTEXT DIAGRAM 10](#_Toc346626334)

[DATA FLOW DIAGRAM (DFD) 11](#_Toc346626335)

[CONTROL FLOW DIAGRAM 15](#_Toc346626336)

[STATE DIAGRAM / SEQUENCE DIAGRAM 15](#_Toc346626337)

[ENTITY RELATIONSHIP MODEL 15](#_Toc346626338)

[CLASS DIAGRAM / CRC MODEL /COLLABORATION DIAGRAM / USE-CASE DIAGRAM / ACTIVITY DIAGRAM 17](#_Toc346626339)

[SYSTEM DESIGN 18](#_Toc346626340)

[MODULARISATION DETAILS 18](#_Toc346626341)

[SCHOOL MANAGEMENT SERVER 19](#_Toc346626342)

[SCHOOL MANAGEMENT CLIENT 19](#_Toc346626343)

[SCHOOL MANAGEMENT DATABASE 20](#_Toc346626344)

[DATA INTEGRITY AND CONSTRAINTS 20](#_Toc346626345)

[DATABASE AND TABLE DESIGN 20](#_Toc346626346)

[PROCEDURAL DESIGN / OBJECT ORIENTED DESIGN 21](#_Toc346626347)

[USER INTERFACE DESIGN 22](#_Toc346626348)

[TEST CASES 22](#_Toc346626349)

[UNIT TEST CASES 22](#_Toc346626350)

[SYSTEM TEST CASES 22](#_Toc346626351)

[CODING 23](#_Toc346626352)

[COMPLETE PROJECT CODING 23](#_Toc346626353)

[COMMENTS AND DESCRIPTION OF CODING SEGMENTS 23](#_Toc346626354)

[STANDARDIZATION OF THE CODING 23](#_Toc346626355)

[CODE EFFICIENCY 23](#_Toc346626356)

[ERROR HANDLING 23](#_Toc346626357)

[PARAMETERS CALLING / PASSING 23](#_Toc346626358)

[VALIDATION CHECKS 23](#_Toc346626359)

[TESTING 23](#_Toc346626360)

[TESTING TECHNIQUES AND TESTING STRATEGIES USED 23](#_Toc346626361)

[TESTING PLAN USED 23](#_Toc346626362)

[TESTING REPORTS 23](#_Toc346626363)

[UNIT TEST CASES 23](#_Toc346626364)

[SYSTEM TEST CASES 24](#_Toc346626365)

[DEBUGGING AND CODE IMPROVEMENT 24](#_Toc346626366)

[SYSTEM SECURITY MEASURES 24](#_Toc346626367)

[DATABASE / DATA SECURITY 24](#_Toc346626368)

[CREATION OF USER PROFILES AND ACCESS RIGHTS 24](#_Toc346626369)

[COST ESTIMATION 24](#_Toc346626370)

[COST ESTIMATION MODEL 24](#_Toc346626371)

[REPORTS 24](#_Toc346626372)

[FUTURE SCOPE AND FURTHER ENHANCEMENT 25](#_Toc346626373)

[BIBLIOGRAPHY 25](#_Toc346626374)

[APPENDICES 25](#_Toc346626375)

[GLOSSARY 25](#_Toc346626376)

# INTRODUCTION AND 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.

# SYSTEM ANALYSIS

## IDENTIFICATION OF NEED

When I was a kid I saw my parents coming to the school for submitting the remuneration and other reasons. The entire data was written using pen and papers so undoubtedly it took lots of time completing even minor queries. Nowadays I see my relatives going to the school to admit their kids and they face huge problems as no proper procedure is followed even by larger and more reputed institutes. I have always felt that there is a need of a proper procedure which could not only save lots of time of both the school authority and the guardians, also easy to maintain the entire process. So I really feel that the modernization of technology could be used to minimize these problems. That is why a proper application or software is required to ease the task of the school administration and save the time of the parents as well.

## PRELIMINARY INVESTIGATION

I started talking to my relatives and neighbors who have recently admitted their child to the schools or they are in the process of admitting them. Naturally, they are the ones who know the problems very well. I also spoke with some of school administrations in my locality about the problems they faced and what exactly would they want if they are given software like School Management System just to get an idea on what could the points be if we want to develop a new application to minimize school management related problems and difficulties. I obviously then consulted with some IT professionals and software engineers and expressed my view to them. I was really amazed to see their positive response on this matter and I got helps in many ways from them. After completing all this process I decided to develop this application for sure.

## FEASIBILITY STUDY

We all know that the numbers of school is growing as fast as Information Technology. Developing an application is very easy these days so people would love to use technology ease their tasks. There is need and there is solution so undoubtedly this software is going to be appreciated by the market.

## PROJECT PLANNING

### TRACKING GANTT



## PROJECT SCHEDULING

### PERT CHART (NETWORK DIAGRAM)



### GANTT CHART



## SOFTWARE REQUIREMENT AND 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

## SOFTWARE ENGINEERING PARADIGM APPLIED

Project Category

## DATA MODELS

### CONTEXT DIAGRAM



### DATA FLOW DIAGRAM (DFD)

#### LEVEL 0 DFD



#### LEVEL 1 DFD







#### LEVEL 2 DFD



### CONTROL FLOW DIAGRAM







### STATE DIAGRAM / SEQUENCE DIAGRAM

### ENTITY RELATIONSHIP MODEL

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 SystemhasStudents 1 : N

School Management System has Faculties1 : N

School Management System has Admin 1 : 1

Studenthas Attendance1 :1

AdminChecksAttendance1 : 1

StudentsreadsBooksM : N

StudentspaysAccount1 : 1

AdminControlsAccount1 : 1

Students Studies in Course N:1



### CLASS DIAGRAM / CRC MODEL /COLLABORATION DIAGRAM / USE-CASE DIAGRAM / ACTIVITY DIAGRAM



# SYSTEM DESIGN

## MODULARISATION DETAILS



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.

## DATA INTEGRITY AND CONSTRAINTS

?????????? database related issues and limitations

## DATABASE AND TABLE DESIGN

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.



## PROCEDURAL DESIGN / OBJECT ORIENTED DESIGN

## USER INTERFACE DESIGN

## TEST CASES

### UNIT TEST CASES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID** | **ITEM** | **DESCRIPTION** | **ACTUAL RESULT** | **TESTED BY** |
| SMS – 001 | Login | Enter User ID and Password for Login. | Successfully Logged in. | Susmita |
| SMS – 002 | Cancel | Select Cancel to close Login window. | Successfully Canceled. | Susmita |

### SYSTEM TEST CASES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID** | **ITEM** | **DESCRIPTION** | **ACTUAL RESULT** | **TESTED BY** |
| SMS – 001 | Login | Enter User ID and Password for Login. | Successfully Logged in. | Susmita |
| SMS – 002 | Cancel | Select Cancel to close Login window. | Successfully Canceled. | Susmita |
| SMS – 003 | Admission | To add a new student enter the Student\_ID, Student\_DOB, Student\_Name, Student\_Parent \_Name, Student\_Address, Student\_Admission\_ Date,Student\_course \_Name, Student\_Contact | New Student is added to the School Management System. | Susmita |
| SMS – 004 | ViewStudentStatus | Enter Student\_ID, Student\_Name, Student\_Course\_Name of the Student. | Show the Student Details. | Susmita |
| SMS – 005 | EditStudentStatus | Select the Student and Click the Edit option. Now edit the Student Details and submit the Details. | Student Details successfully updated. | Susmita |

# CODING

## COMPLETE PROJECT CODING

## COMMENTS AND DESCRIPTION OF CODING SEGMENTS

## STANDARDIZATION OF THE CODING

## CODE EFFICIENCY

## ERROR HANDLING

## PARAMETERS CALLING / PASSING

## VALIDATION CHECKS

# TESTING

## TESTING TECHNIQUES AND TESTING STRATEGIES USED

We have vigorously tested the application to make it error free and smooth. To achieve our goal we tested the modules differently inside the codes and then tested the entire application as a whole to mark its drawbacks.

## TESTING PLAN USED

## TESTING REPORTS

### UNIT TEST CASES

|  |  |  |
| --- | --- | --- |
| Test Case Id | Comments | Status |
| SMS – 001  SMS – 001 | NA  NA | PASS  PASS |

### SYSTEM TEST CASES

|  |  |  |
| --- | --- | --- |
| Test Case Id | Comments | Status |
| SMS – 001 | NA | PASS |
| SMS – 002  SMS – 003  SMS – 004  SMS – 005  SMS – 006  SMS – 007  SMS – 008  SMS – 009  SMS – 010  SMS – 011  SMS – 012  SMS – 013  SMS – 014  SMS – 015  SMS – 016  SMS – 017  SMS – 018  SMS – 019  SMS – 020  SMS – 021  SMS – 022  SMS – 023  SMS – 024  SMS – 025  SMS – 026  SMS – 027  SMS – 028  SMS – 029  SMS – 030 | NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA | PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS  PASS |

## DEBUGGING AND CODE IMPROVEMENT

# SYSTEM SECURITY MEASURES

* 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.

## DATABASE / DATA SECURITY

## CREATION OF USER PROFILES AND ACCESS RIGHTS

# COST ESTIMATION

## COST ESTIMATION MODEL

??????????????????????????



# 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

# FUTURE SCOPE AND FURTHER ENHANCEMENT

* 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

# APPENDICES

# GLOSSARY