

STUDENT RESULT MANAGEMENT SYSTEM

Mini Project Report

Submitted to

Visvesvaraya Technological University

Belagavi-590018

By

NITHIN THOMAS

USN: 4SU18CS052

NIVIL SHIBU

USN: 4SU18CS053

Under the guidance of

Mrs. Suchetha NV

Assistant Professor

in partial fulfilment of the requirements for the award of

V Semester Database Management System Lab



Department of Computer Science and Engineering

SDM INSTITUTE OF TECHNOLOGY

UJIRE- 574 240

2020-21

S. D. M. INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)

UJIRE-574240

Department of Computer Science and Engineering

CERTIFICATE

Certified that the project work entitled “**STUDENT RESULT MANAGEMENT SYSTEM**” is carried out by **Nithin Thomas** bearing USN **4SU18CS052** and **Nivil Shibu** bearing USN **4SU18CS053** in partial fulfilment for requirements for **V SEMESTER DBMS Laboratory with Mini Project** of the Visvesvaraya Technological University, Belagavi during the year 2020-21. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Database Management system lab.

Signature of the Guide

(Mrs. Suchetha NV)

Signature of the H.O.D

(Dr. Thayagaraju G.S)

EXTERNAL VIVA

Name of the Examiners:

1. _____

2. _____

Acknowledgement

We express our deepest gratitude to our guide **Suchetha NV**, Asst. Professor Department of Computer Science and Engineering, for her valuable guidance and encouragement while doing this project work.

We are obligated to **Dr. Thayagaraju G. S.** Head of the Department, and **Dr. Ashok Kumar T** Principal, for their advice and suggestions at various stages of the work. We also extend our heartfelt gratitude to **Saritha M**, Asst. Professor Department of Computer Science and Engineering, for her assistance.

We also extend our thanks to the management of SDM Institute of Technology, Ujire, for providing an excellent study environment, reference materials and laboratories facilities. We remain grateful to the co-operation and help rendered by the teaching and non-teaching staff to the department.

Nithin Thomas (4SU18CS052)

Nivil Shibu (4SU18CS053)

Abstract

Databases have been in use since the earliest days of electronic computing. Unlike modern systems which can be applied to widely different databases and needs, the vast majority of older systems were tightly linked to the custom databases in order to gain speed at the expense of flexibility.

This specification document describes the capabilities that will be provided by the software application STUDENT RESULT MANAGEMENT SYSTEM. It also states the various constraints by which the system will abide. The intended audience for this document is the development team, testing team and end users of the product.

The application will manage the information about various students enrolled in this course in different years, the subjects offered during different semesters of the course, the marks obtained by the various students in various subjects in different semesters. The application will greatly simplify and speed up the result preparation and management process.

Table of contents

	Page No.
Acknowledgement	i
Abstract	ii
Table of Contents	iii
Chapter 1 Introduction	1
Chapter 2 Literature Review and Problem Statement	2
2.1 Introduction to DBMS	
2.2 What is MY SQL	
Chapter 3 Problem Formulation	7
3.1 Problem statement	
3.2 Aim of the work	
3.3 Objectives	
Chapter 4 Requirement Specifications	9
4.1 Hardware requirements	
4.2 Software requirements	
Chapter 5 Database Design	10
5.1 Entity relationship diagram	
5.2 Conversion from ER diagram to schema	
5.3 Normalization	
5.4 Scheme diagram	

Chapter 6	Implementation	16
6.1	Tools Description	
6.2	Stored Procedures	
6.3	Trigger	
6.4	Scripts	
Chapter 7	Results and Discussion	44
Chapter 8	Conclusions and Scope for future work	48
Bibliography		49
Personal Profile		50

Introduction

This document aims at defining overall software requirement for STUDENT RESULT MANAGEMENT SYSTEM. Efforts have been made to define the requirements exhaustively and accurately. The final product will be having only features/functionalities mentioned in this document and assumptions for any additional functionality/feature should not be made by any of the parties involved in developing/testing/implementing /using this product.

Student Result Management System divided in two modules–

- Student
- Admin

Admin Features-

- Admin Dashboard
- Admin can add/update/ Class
- Admin can add/update/ Subjects
- Admin can add/update/ Active/Inactive Subject combination with class
- Admin can register new student and also edit info of the student
- admin can declare/ edit result of a student.
- Admin can change own password

Students features-

- Student can search their result using valid roll-id.

Literature Review

2.1 Introduction to Database Management System:

A database management system (DBMS) refers to the technology for creating and managing databases. DBMS is a software tool to organize (create, retrieve, update and manage) data in a database.

The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. A Database is a collection of logically related tables which are stored in a file or set of files. A database is a structured object that consists of data and metadata. A database field can be described as a part of a record which is reserved for a specific type of data. A record is the collection of fields where each field contains specific information.

A database table is a collection of records of a specific type. A Database Management System is a general-purpose software system which allows the users to construct the database, analyse the data, maintain the data in the database, share the database among various users and application and much more. In short, DBMS allows us to create, define and manipulate the database. there are 4 components: Queries, DBMS, Operating System and Database.

The queries are the user's or application's request to get some data from the database or modify some data in the database. The queries are not directly communicated to the database. DBMS acts as an intermediate agent which communicates the queries to the database via the operating system. A DBMS allow us to maintain a centralized repository of data, the data is integrated. In case of traditional file management system, the data is not that integrated. The data may be stored in Hierarchical Files.

The management of data is much easier and more convenient in the DBMS.

The data can be accessed across multiple locations at the same time and concurrency conflicts can be effectively handled in DBMS.

2.1.1 Applications of Database

The development of computer graphics has been driven both by the needs of the user community and by the advances in hardware and software. The applications of database are many and varied; it can be divided into four major areas:

1. Hierarchical
2. Flexibility with relational database
3. Object oriented application.
4. Interchanging the data on the web for ecommerce.
5. Extending database capabilities for applications.

2.1.2 Hierarchical and Network System:

Many early database applications-maintained records in large organization such as corporations, universities, hospitals, and banks. In many of these applications, there were large numbers of records of similar structure. For example, in a university application, information would be kept for each student, each course, and each grade record and so on. There were also many interrelationships among them. Another short coming of early systems was that they provided only programming language interfaces. This made it time consuming and expensive to implement a new queries and transactions. Since new program had to be written, tested and debugged.

2.1.3 Flexibility with relational database:

Relational database was originally proposed to separate the physical storage of data from its conceptual representation and to provide a mathematical foundation for content storage. The relational data model also introduced high-level query languages that provided an alternative to programming language interfaces; hence it was a lot quicker to write new queries. Relational systems were initially targeted to the same applications as earlier systems. But were meant to provide flexibility to develop new queries quickly and to recognize the database as requirements change.

2.1.4 Object oriented application:

The object-oriented databases were considered a competitor to relational databases, since they provide a more general data structure. They also incorporated many of the useful object-oriented paradigms, such as abstract data types, encapsulation of operations, inheritance, and object identity. However, the complexity of the model and the lack of an early standard contributed to their limited use. They are now mainly used in specialized applications, such as engineering design, multimedia publishing, and manufacturing systems. Despite expectations that they will make a big impact, their overall penetrations into the database product market remains under 5% today.

2.1.5 Interchanging the data on the web for e-commerce:

The World Wide Web provides a large network of interconnected computers. Users can create documents using a Web publishing language, such as Hypertext Markup Language (HTML) and stores these documents on the web server who other users (clients)

can access them. Documents can be linked through hyperlinks, which points to other documents. It quickly became apparent that parts of the information on e-commerce web pages were often dynamically extracted data from DBMSs. A variety of technique uses were developed to interchange the data on the web.

2.1.6 Extending Databases Capabilities for New Applications:

The success of database systems in traditional applications encouraged developers of other types of applications to attempt to use them. Such applications traditionally used their own specialized file and data structures. Scientific applications that store large amount of data resulting from scientific experiments in areas such as high energy physics or the mapping of the human genome. Storage and retrieval of images, from scanned news or personal photographs to satellite photograph images and images from medical procedure such as X rays and MRI.

2.2 What is MySQL?

MySQL is multithreaded, multi user SQL database management system (DBMS). The basic program run as server providing multi user access to a number of databases. The project's source code is available under terms of the GNU General Public License, as well as under a variety of property arguments. MySQL is a database. The data in a MySQL is stored in a Database objects called tables. A table is a collection of related data entries and it consists of columns and rows. The databases are useful when storing information categorically.

MySQL is a central components of the LAMP open source web application software stack (and other "AMP" stacks). LAMP is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. Application that use the MySQL database include TYP03, MODx,

Joomla, WordPress, phpBB, MyBB and Drupal. MySQL is also used large scale web sites, including Google (Though not for the searches).

2.2.1 MySQL Command Syntax

As you might have observed from the simple program in the previous section, MySQL uses mainly six commands in which SELECT is used to retrieve rows selected from one or more tables. FROM refers to the table from which we need to select the attributes. WHERE clause, if given, indicates condition or conditions that rows must satisfy to be selected. Where condition is expression that evaluates to true for each row to be selected. This statement selects all rows if there is nowhere clause. GROUP BY clause used to group the values of the attributes provided that values must be same. HAVING clause is applied nearly last, just before items are sent to the client, with no optimization. If the HAVING clause refers to a column that is ambiguous, warning occurs. ORDER BY clause is used for the purpose of sorting the values of the attributes in a result. If you use GROUP BY, output rows are sorted according to GROUP BY columns as if you had an ORDER BY for the same column.

2.2.2 MySQL-related Libraries

The MySQL Servlet extensions are lightweight wrappers on top of a C client library. The extensions can either use the MySQL and library or libmysqlclientlibrary. Choosing a library is a compile time decision. The MySQL and library are part of the Servlet distribution since 5.3.0. It offers features like lazy connections and query caching, features that are not available with the libmysqlclient, so using the built-in library is highly recommended. It is recommended to use the MySQL and library instead of the MySQL client server. Both libraries are supported and constantly being improved.

Problem Formulation

3.1 Problem Statement

The project mainly aims at minimising the manual process of student result management system, here student can login and view his/her result on his rest place easily. Admin can add/manage student and his results. Front and back end are implemented using PHP and MySQL respectively.

The current manual system has a lot of paper work. To maintain the records of students manually is the time-consuming task. With the increase in database, it will become a massive task to maintain the database. Requires large quantities of file cabinets, which are huge and require quite a bit of space in the office, which can be used for storing records of previous details. The retrieval of records of previously registered students will be a tedious task. Lack of security for the records, anyone disarrange the records of your system. If someone want to check the details of the available students the previous system does not provide any necessary detail of this type.

To solve this problem, we created a web application of Student Result Management System. This application completely computerized eliminating paper work and speeding up the maintenance of the information.

3.2 Aim of the work

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of existing system. The system provides proper security and reduces manual work. The proposed Student Result Management System application is created to manage user data in detail. This reduces dependency on manual system and can easily help in tracking individual user records for any references.

3.3 Objectives

- The main objective of the project is to provide the examination result to the student in a simple way.
- By a result analyser with subject status and marks is an application tool for displaying the results in secure way.
- The system is intended for the student. And the privileges that are provided to student are to read and execute his/her result by providing username and password for secure login and in case of new student the registration is done by the admin.
- The whole result analyser will be under the control of the administrator and the admin as the full privileges to read, write and execute the result.
- The student can download his/her result.

Requirement Specification

Requirements specification is a specification of software requirements and hardware requirements required to do the project. Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analysing, documenting, validating and managing software or system requirements.

4.1 Hardware Configuration

Sl No.	Hardware/Equipment	Specifications
1	Processor	Pentium IV
2	Clock Speed	2.2 GHz
3	Monitor	1024*768 resolution, colour
4	RAM	512 MB

4.2 Software Requirements

Sl No.	Software	Specifications
1	Web Browser	Google Chrome/ Microsoft Edge
2	Operating System	Windows 10
3	Scripting	HTML, CSS
4	Database	MySQL
5	Tool used	PHP

System Design and Database Design

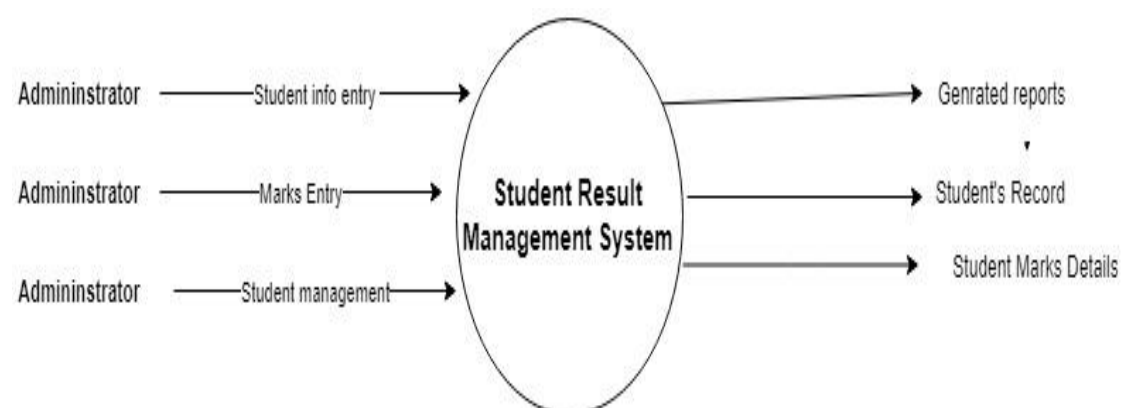
System Design

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

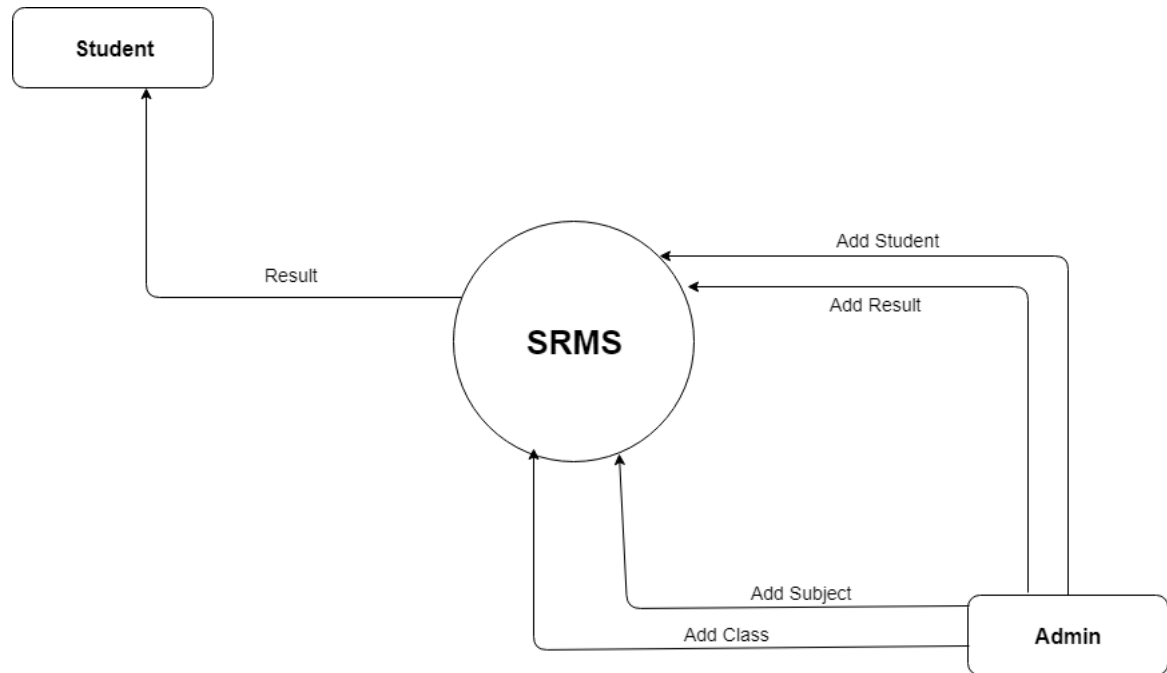
The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

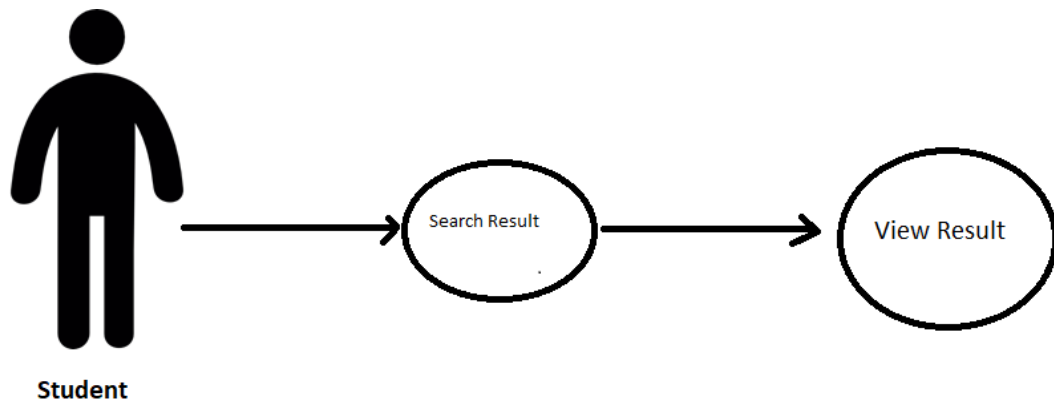
Context Diagram 0



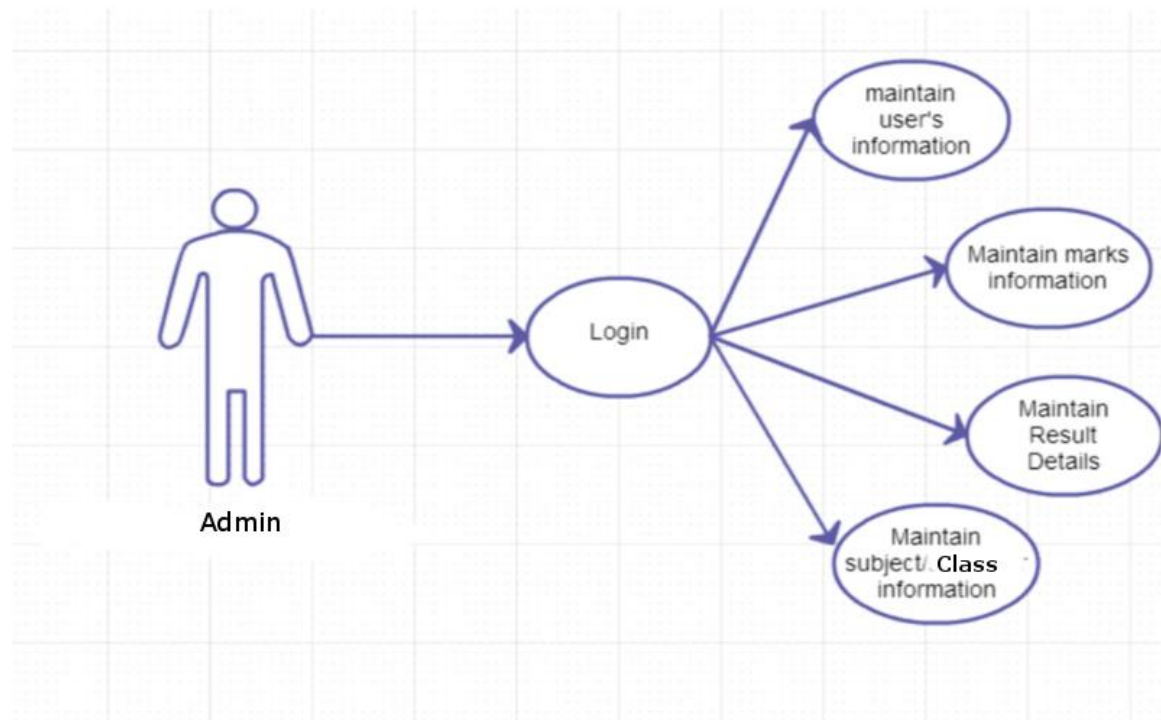
Context Diagram 1



Student case Diagram



Admin Case Diagram



Database Design

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MySQL Access database has been chosen for developing the relevant databases.

5.1 Entity relationship diagram

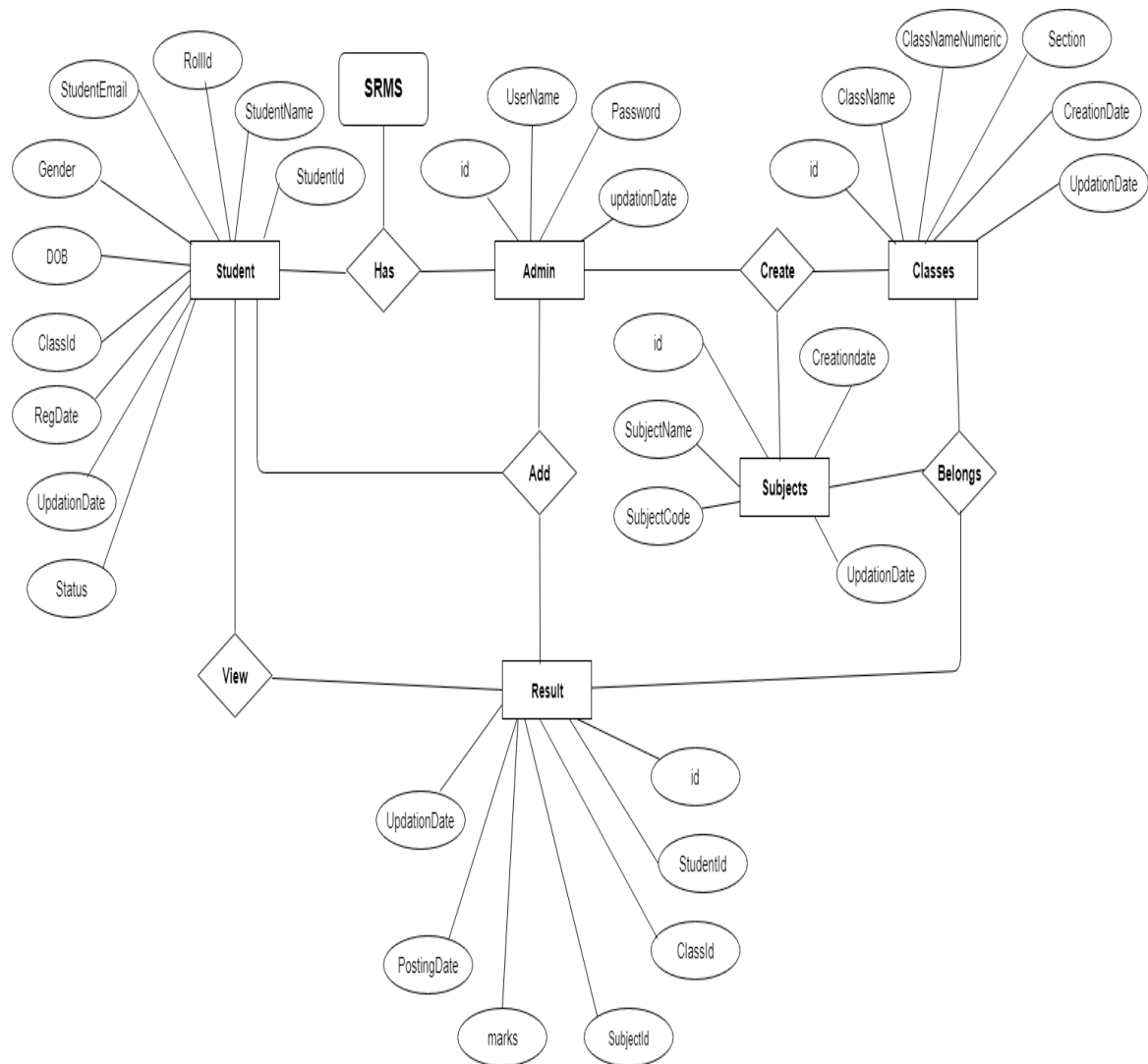


Figure 5.1 Entity relationship diagram for Hospital management system.

5.2 Conversion from ER diagram to schema

In order to convert from Entity-Relationship diagram to schema Diagram need to apply the following methods such as relational mapping algorithm, normalization and must follow the application development process.

5.3 Normalization

Normalization is the process of decomposing larger relations into smaller relations by analysis the given relation schema, based on their functional dependencies and primary keys to achieve the desirable properties of

- (i) Minimizing redundancy.
- (ii) Minimizing insertion, deletion and updating anomalies.

There are mainly five types of Normalization, but in this particular project we are mainly deals with the first three types of Normalization.

- **First Normal Form**

A table is said to be in 1NF if all the attributes of a table are atomic(simple, indivisible) and single valued.

- **Second Normal Form**

A table is said to be in 2 NF if

- (i) The table should be in 2NF.
- (ii) There should not be any transitive dependency in the table.

- **Third Normal Form**

A table is said to be in 3NF if the table should be in 2NF and there should not be any transitive dependency in the table.

5.4 Schema Diagram

admin

<u>id</u>	UserName	Password	updationDate
-----------	----------	----------	--------------

tblclasses

<u>id</u>	className	ClassnameNumeric	Section	CreationDate	UpdationDate
-----------	-----------	------------------	---------	--------------	--------------

tblstudents

<u>StudentId</u>	StudentName	RollId	StudentEmail	Gender	DOB	ClassId	RegDate	UpdationDate	Status
------------------	-------------	--------	--------------	--------	-----	---------	---------	--------------	--------

tblresult

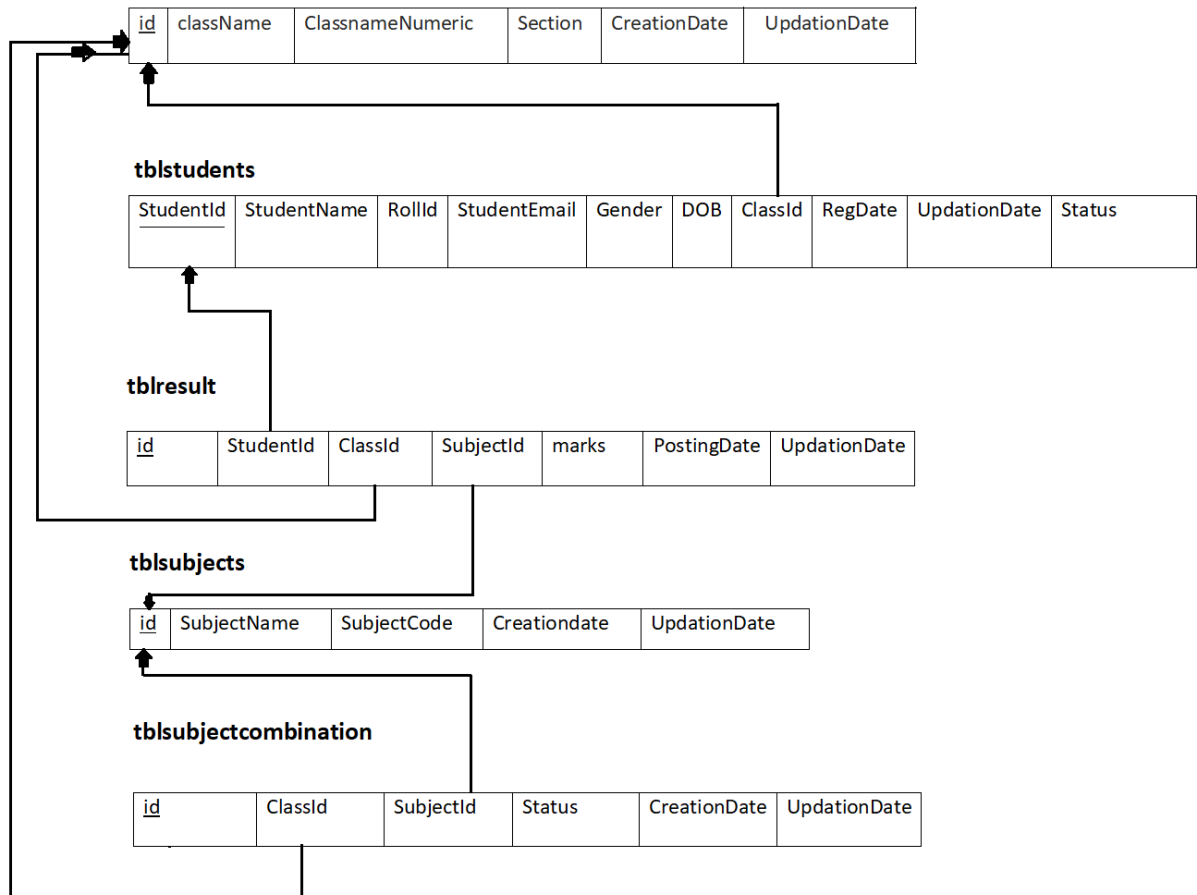
<u>id</u>	StudentId	ClassId	SubjectId	marks	PostingDate	UpdationDate
-----------	-----------	---------	-----------	-------	-------------	--------------

tblsubjects

<u>id</u>	SubjectName	SubjectCode	Creationdate	UpdationDate
-----------	-------------	-------------	--------------	--------------

tblsubjectcombination

<u>id</u>	ClassId	SubjectId	Status	CreationDate	UpdationDate
-----------	---------	-----------	--------	--------------	--------------



Implementation

The project is implemented using MySQL database along with Servlet and HTML. In this project, we use the below mentioned concepts for implementing different type of effects like HTML pages to demonstrate on a web page as a stand-alone or web-based application. It is a three-tier architecture project. A three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms. Three-tier architecture is a software design pattern and a well - established software architecture.

6.1 Tools Description

The user interface (UI), in the industrial design field of human-computer interaction in the space where interactions between humans and machines occur. The goal of this interactions is to allow effective operations and control of the machine from the human end, whilst the machine simultaneously feeds back information that aids the operator's decision-making process. In this particular project, we used HTML as user interface tool. The main goal of this HTML user interface design is to produce a user interface which makes it easy (self-explanatory) efficient, and enjoyable (user friendly) to a machine in the way which produces the desired result. This generally means that the operator needs to provide minimal input to achieve the desired output, and also that the machine minimizes undesired outputs to the human.

6.2 Stored Procedures:

A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name, which are stored in a relational database management system as a group, so it can be reused and shared by multiple programs. Stored procedure can

access or modify data in a database, but it is not tied to a specific database or object, which offers a number of advantages.

6.3 Triggers

A database trigger is a procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database.

6.4 Scripts:

6.4.1 Index Page

```
<?php

session_start();

error_reporting(0);

include('includes/config.php');

if ($_SESSION['alogin'] != '') {

$_SESSION['alogin'] = '';

}

if (isset($_POST['login'])) {

$username = $_POST['username'];

$password = md5($_POST['password']);

$sql = "SELECT UserName,Password FROM admin WHERE UserName=:username and
Password=:password";

$query = $dbh->prepare($sql);
```

STUDENT RESULT MANAGEMENT SYSTEM

```
$query->bindParam(':uname', $uname, PDO::PARAM_STR);

$query->bindParam(':password', $password, PDO::PARAM_STR);

$query->execute();

$results = $query->fetchAll(PDO::FETCH_OBJ);

if ($query->rowCount() > 0) {

    $_SESSION['alogin'] = $_POST['username'];

    echo "<script type='text/javascript'> document.location = 'dashboard.php'; </script>";

} else {

    echo "<script>if (!window.Notification) {

        console.log('Browser does not support notifications.');"

    } else {

        // check if permission is already granted

        if (Notification.permission === 'granted') {

            // show notification here

            var notify = new Notification('OOPS!', {

                body: 'Invalid Password',

                icon: 'https://thumbs.dreamstime.com/z/invalid-red-stamp-white-background-finger-

                hand-pointing-52829258.jpg',

            });

        } else {

            // request permission from user

            Notification.requestPermission().then(function(p) {

                if(p === 'granted') {

                    // show notification here
```


STUDENT RESULT MANAGEMENT SYSTEM

```
} else {  
  
console.log('User blocked notifications.');
```



```
}  
  
}).catch(function(err) {  
  
console.error(err);  
  
});  
  
}  
  
}</script>";  
  
}  
  
}  
  
?>  
  
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
<meta charset="utf-8">  
  
<meta http-equiv="X-UA-Compatible" content="IE=edge">  
  
<meta name="viewport" content="width=device-width, initial-scale=1">  
  
<title>Admin Login</title>  
  
<link rel="stylesheet" href="css/bootstrap.min.css" media="screen">  
  
<link rel="stylesheet" href="css/font-awesome.min.css" media="screen">  
  
<link rel="stylesheet" href="css/animate-css/animate.min.css" media="screen">  
  
<link rel="stylesheet" href="css/prism/prism.css" media="screen"> <!-- USED FOR DEMO  
HELP - YOU CAN REMOVE IT -->  
  
<link rel="stylesheet" href="css/main.css" media="screen">
```

STUDENT RESULT MANAGEMENT SYSTEM

```
<script src="js/modernizr/modernizr.min.js"></script>

</head>

<body class="">

<div class="main-wrapper">

<div class="">

<div class="row">

<h1 align="center">Student Result Management System</h1>

<div class="col-lg-6 visible-lg-block">

<section class="section">

<div class="row mt-40">

<div class="col-md-10 col-md-offset-1 pt-50">

<div class="row mt-30 ">

<div class="col-md-11">

<div class="panel">

<div class="panel-heading">

<div class="panel-title text-center">

<h4>For Students</h4>

</div>

</div>

<div class="panel-body p-20">

<div class="section-title">

<p class="sub-title">Student Result Management System</p>

</div>

<form class="form-horizontal" method="post">
```

STUDENT RESULT MANAGEMENT SYSTEM

```
<div class="form-group">

<label for="inputEmail3" class="col-sm-6 control-label">Search your result</label>

<div class="col-sm-6">

<a href="find-result.php">click here</a>

</div>

</div>

</form>

</div>

</div>

<!-- /.panel -->

</div>

<!-- /.col-md-11 -->

</div>

<!-- /.row -->

</div>

<!-- /.col-md-12 -->

</div>

<!-- /.row -->

</section>

</div>

<div class="col-lg-6">

<section class="section">

<div class="row mt-40">

<div class="col-md-10 col-md-offset-1 pt-50">
```

STUDENT RESULT MANAGEMENT SYSTEM

```
<div class="row mt-30 ">

<div class="col-md-11">

<div class="panel">

<div class="panel-heading">

<div class="panel-title text-center">

<h4>Admin Login</h4>

</div>

</div>

<div class="panel-body p-20">

<div class="section-title">

<p class="sub-title">Student Result Management System</p>

</div>

<form class="form-horizontal" method="post">

<div class="form-group">

<label for="inputEmail3" class="col-sm-2 control-label">Email</label>

<div class="col-sm-10">

<input type="text" name="username" class="form-control" id="inputEmail3"
placeholder="UserName">

</div>

</div>

<div class="form-group">

<label for="inputPassword3" class="col-sm-2 control-label">Password</label>

<div class="col-sm-10">

<input type="password" name="password" class="form-control" id="inputPassword3"
placeholder="Password">


```

```
</div>

</div>

<div class="form-group mt-20">

<div class="col-sm-offset-2 col-sm-10">

<button type="submit" name="login" class="btn btn-success btn-labeled pull-right">Sign
in<span class="btn-label btn-label-right"><i class="fa fa-check"></i></span></button>

</div>

</div>

</form>

</div>

</div>

<!-- /.panel -->

<p class="text-muted text-center"><small>Copyright © Nithin & Nivil</small></p>

</div>

<!-- /.col-md-11 -->

</div>

<!-- /.row -->

</div>

<!-- /.col-md-12 -->

</div>

<!-- /.row -->

</section>

</div>

<!-- /.col-md-6 -->
```

STUDENT RESULT MANAGEMENT SYSTEM

```
</div>

<!-- /.row -->

</div>

<!-- /. -->

</div>

<!-- /.main-wrapper -->


<!-- ===== COMMON JS FILES ===== -->

<script src="js/jquery/jquery-2.2.4.min.js"></script>

<script src="js/jquery-ui/jquery-ui.min.js"></script>

<script src="js/bootstrap/bootstrap.min.js"></script>

<script src="js/pace/pace.min.js"></script>

<script src="js/lobipanel/lobipanel.min.js"></script>

<script src="js/iscroll/iscroll.js"></script>


<script src="js/main.js"></script>

<script>

$(function() {

});

</script>

</body>

</html>
```

6.4.2 Dashboard

```
<?php

session_start();

error_reporting(0);

include('includes/config.php');

if(strlen($_SESSION['alogin'])=="")

{

header("Location: index.php");

}

else{

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Student Result Management System | Dashboard</title>

<link rel="stylesheet" href="css/bootstrap.min.css" media="screen" >

<link rel="stylesheet" href="css/font-awesome.min.css" media="screen" >

<link rel="stylesheet" href="css/animate-css/animate.min.css" media="screen" >

<link rel="stylesheet" href="css/lobipanel/lobipanel.min.css" media="screen" >

<link rel="stylesheet" href="css/toastr/toastr.min.css" media="screen" >

<link rel="stylesheet" href="css/icheck/skins/line/blue.css" >
```

STUDENT RESULT MANAGEMENT SYSTEM

```
<link rel="stylesheet" href="css/icheck/skins/line/red.css" >

<link rel="stylesheet" href="css/icheck/skins/line/green.css" >

<link rel="stylesheet" href="css/main.css" media="screen" >

<script src="js/modernizr/modernizr.min.js"></script>

</head>

<body class="top-navbar-fixed">

<div class="main-wrapper">

<?php include('includes/topbar.php');?>

<div class="content-wrapper">

<div class="content-container">

<?php include('includes/leftbar.php');?>

<div class="main-page">

<div class="container-fluid">

<div class="row page-title-div">

<div class="col-sm-6">

<h2 class="title">Dashboard</h2>

</div>

<!-- /.col-sm-6 -->

</div>

<!-- /.row -->

</div>

<!-- /.container-fluid -->

<section class="section">

<div class="container-fluid">
```


STUDENT RESULT MANAGEMENT SYSTEM

```
<div class="row">

<div class="col-lg-3 col-md-3 col-sm-6 col-xs-12">

<a class="dashboard-stat bg-primary" href="manage-students.php">

<?php

$sql1 ="SELECT StudentId from tblstudents ";

$query1 = $dbh -> prepare($sql1);

$query1->execute();

$results1=$query1->fetchAll(PDO::FETCH_OBJ);

$totalstudents=$query1->rowCount();

?>

<span class="number counter"><?php echo htmlentities($totalstudents);?></span>

<span class="name">Regd Users</span>

<span class="bg-icon"><i class="fa fa-users"></i></span>

</a>

<!-- /.dashboard-stat -->

</div>

<!-- /.col-lg-3 col-md-3 col-sm-6 col-xs-12 -->

<div class="col-lg-3 col-md-3 col-sm-6 col-xs-12">

<a class="dashboard-stat bg-danger" href="manage-subjects.php">

<?php

$sql ="SELECT id from tblsubjects ";

$query = $dbh -> prepare($sql);

$query->execute();

$results=$query->fetchAll(PDO::FETCH_OBJ);
```

STUDENT RESULT MANAGEMENT SYSTEM

```
$totalsubjects=$query->rowCount();

?>

<span class="number counter"><?php echo htmlentities($totalsubjects);?></span>

<span class="name">Subjects Listed</span>

<span class="bg-icon"><i class="fa fa-ticket"></i></span>

</a>

<!-- /.dashboard-stat -->

</div>

<!-- /.col-lg-3 col-md-3 col-sm-6 col-xs-12 -->

<div class="col-lg-3 col-md-3 col-sm-6 col-xs-12">

<a class="dashboard-stat bg-warning" href="manage-classes.php">

<?php

$sql2 ="SELECT id from tblclasses ";

$query2 = $dbh -> prepare($sql2);

$query2->execute();

$results2=$query2->fetchAll(PDO::FETCH_OBJ);

$totalclasses=$query2->rowCount();

?>

<span class="number counter"><?php echo htmlentities($totalclasses);?></span>

<span class="name">Total classes listed</span>

<span class="bg-icon"><i class="fa fa-bank"></i></span>

</a>

<!-- /.dashboard-stat -->

</div>
```

STUDENT RESULT MANAGEMENT SYSTEM

```
<!-- /.col-lg-3 col-md-3 col-sm-6 col-xs-12 -->

<div class="col-lg-3 col-md-3 col-sm-6 col-xs-12">

<a class="dashboard-stat bg-success" href="manage-results.php">

<?php

$sql3="SELECT distinct StudentId from tblresult ";

$query3 = $dbh -> prepare($sql3);

$query3->execute();

$results3=$query3->fetchAll(PDO::FETCH_OBJ);

$totalresults=$query3->rowCount();

?>

<span class="number counter"><?php echo htmlentities($totalresults);?></span>

<span class="name">Results Declared</span>

<span class="bg-icon"><i class="fa fa-file-text"></i></span>

</a>

<!-- /.dashboard-stat -->

</div>

<!-- /.col-lg-3 col-md-3 col-sm-6 col-xs-12 -->

</div>

<!-- /.row -->

</div>

<!-- /.container-fluid -->

</section>

<!-- /.section -->
```

```
</div>

<!-- /.main-page -->

</div>

<!-- /.content-container -->

</div>

<!-- /.content-wrapper -->

</div>

<!-- /.main-wrapper -->

<!-- ===== COMMON JS FILES ===== -->

<script src="js/jquery/jquery-2.2.4.min.js"></script>

<script src="js/jquery-ui/jquery-ui.min.js"></script>

<script src="js/bootstrap/bootstrap.min.js"></script>

<script src="js/pace/pace.min.js"></script>

<script src="js/lobipanel/lobipanel.min.js"></script>

<script src="js/iscroll/iscroll.js"></script>

<!-- ===== PAGE JS FILES ===== -->

<script src="js/prism/prism.js"></script>

<script src="js/waypoint/waypoints.min.js"></script>

<script src="js/counterUp/jquery.counterup.min.js"></script>

<script src="js/amcharts/amcharts.js"></script>

<script src="js/amcharts/serial.js"></script>

<script src="js/amcharts/plugins/export/export.min.js"></script>
```

STUDENT RESULT MANAGEMENT SYSTEM

```
<link rel="stylesheet" href="js/amcharts/plugins/export/export.css" type="text/css"
media="all" />
```

```
<script src="js/amcharts/themes/light.js"></script>
```

```
<script src="js/toastr/toastr.min.js"></script>
```

```
<script src="js/icheck/icheck.min.js"></script>
```

```
<!-- ===== THEME JS ===== -->
```

```
<script src="js/main.js"></script>
```

```
<script src="js/production-chart.js"></script>
```

```
<script src="js/traffic-chart.js"></script>
```

```
<script src="js/task-list.js"></script>
```

```
<script>
```

```
$(function(){
```

```
// Counter for dashboard stats
```

```
$('.counter').counterUp({
```

```
delay: 10,
```

```
time: 1000
```

```
});
```

```
// Welcome notification
```

```
toastr.options = {
```

```
"closeButton": true,
```

```
"debug": false,
```

```
"newestOnTop": false,
```

```
"progressBar": false,
```

STUDENT RESULT MANAGEMENT SYSTEM

```
"positionClass": "toast-top-right",

"preventDuplicates": false,

"onclick": null,

"showDuration": "300",

"hideDuration": "1000",

"timeOut": "5000",

"extendedTimeOut": "1000",

"showEasing": "swing",

"hideEasing": "linear",

"showMethod": "fadeIn",

"hideMethod": "fadeOut"

}

toastr["success"]( "Welcome to student Result Management System!");

});

</script>

</body>

</html>

<?php } ?>
```

6.4.3 Result

```
<?php

session_start();

error_reporting(0);

include('includes/config.php');
```

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Result Management System</title>

<link rel="stylesheet" href="css/bootstrap.min.css" media="screen">

<link rel="stylesheet" href="css/font-awesome.min.css" media="screen">

<link rel="stylesheet" href="css/animate-css/animate.min.css" media="screen">

<link rel="stylesheet" href="css/lobipanel/lobipanel.min.css" media="screen">

<link rel="stylesheet" href="css/prism/prism.css" media="screen">

<link rel="stylesheet" href="css/main.css" media="screen">

<script src="js/modernizr/modernizr.min.js"></script>

</head>

<body>

<div class="main-wrapper">

<div class="content-wrapper">

<div class="content-container">

<!-- /.left-sidebar -->

<div class="main-page">

STUDENT RESULT MANAGEMENT SYSTEM

```
<div class="container-fluid">

<div class="row page-title-div">

<div class="col-md-12">

<h2 class="title" align="center">Result Management System</h2>

</div>

</div>

<!-- /.row -->

<!-- /.row -->

</div>

<!-- /.container-fluid -->

<section class="section" id="exampl">

<div class="container-fluid">

<div class="row">

<div class="col-md-8 col-md-offset-2">

<div class="panel">

<div class="panel-heading">

<div class="panel-title">

<h3 align="center">Student Result Details</h3>

<hr />

<?php

// code Student Data

$rollid = $_POST['rollid'];

$classid = $_POST['class'];

$_SESSION['rollid'] = $rollid;
```


STUDENT RESULT MANAGEMENT SYSTEM

```
$_SESSION['classid'] = $classid;
```

```
$query = "SELECT
```

```
tblstudents.StudentName,tblstudents.RollId,tblstudents.RegDate,tblstudents.StudentId,  
tblstudents.Status,tblclasses.ClassName,tblclasses.Section from tblstudents join  
tblclasses on tblclasses.id=tblstudents.ClassId where tblstudents.RollId=:rollid and  
tblstudents.ClassId=:classid ";
```

```
$stmt = $dbh->prepare($query);
```

```
$stmt->bindParam(':rollid', $rollid, PDO::PARAM_STR);
```

```
$stmt->bindParam(':classid', $classid, PDO::PARAM_STR);
```

```
$stmt->execute();
```

```
$resultss = $stmt->fetchAll(PDO::FETCH_OBJ);
```

```
$cnt = 1;
```

```
if ($stmt->rowCount() > 0) {
```

```
foreach ($resultss as $row) { ?>
```

```
<p><b>Student Name :</b> <?php echo htmlentities($row->StudentName); ?></p>
```

```
<p><b>Student Roll Id :</b> <?php echo htmlentities($row->RollId); ?>
```

```
<p><b>Student Class:</b> <?php echo htmlentities($row->ClassName); ?><?php echo  
htmlentities($row->Section); ?>
```

```
<?php }
```

```
?>
```

```
</div>
```

```
<div class="panel-body p-20">
```

```
table class="table table-hover table-bordered" border="1" width="100%">
```

```
<thead>
```

```
<tr style="text-align: center">
```

```
<th style="text-align: center">#</th>
```

STUDENT RESULT MANAGEMENT SYSTEM

```
<th style="text-align: center"> Subject</th>

<th style="text-align: center">Marks</th>

</tr>

</thead>

<tbody>

<?php

// Code for result

$query = "select
t.StudentName,t.RollId,t.ClassId,t.marks,SubjectId,tblsubjects.SubjectName from (select
sts.StudentName,sts.RollId,sts.ClassId,tr.marks,SubjectId from tblstudents as sts join
tblresult as tr on tr.StudentId=sts.StudentId) as t join tblsubjects on
tblsubjects.id=t.SubjectId where (t.RollId=:rollid and t.ClassId=:classid)";

$query = $dbh->prepare($query);

$query->bindParam(':rollid', $rollid, PDO::PARAM_STR);

$query->bindParam(':classid', $classid, PDO::PARAM_STR);

$query->execute();

$results = $query->fetchAll(PDO::FETCH_OBJ);

$cnt = 1;

if ($countrow = $query->rowCount() > 0) {

foreach ($results as $result) {

?>

<tr>

<th scope="row" style="text-align: center"><?php echo htmlentities($cnt); ?></th>

<td style="text-align: center"><?php echo htmlentities($result->SubjectName); ?></td>
```

STUDENT RESULT MANAGEMENT SYSTEM

```
<td style="text-align: center"><?php echo htmlentities($totalmarks = $result->marks);
?></td>

</tr>

<?php
$totlcount += $totalmarks;

$cnt++;
}

?>

<tr>

<th scope="row" colspan="2" style="text-align: center">Total Marks</th>

<td style="text-align: center"><b><?php echo htmlentities($totlcount); ?></b> out of
<b><?php echo htmlentities($outof = ($cnt - 1) * 100); ?></b></td>

</tr>

<tr>

<th scope="row" colspan="2" style="text-align: center">Percentage</th>

<td style="text-align: center"><b><?php echo htmlentities($totlcount * (100) / $outof);
?> %</b></td>

</tr>

<tr>

<td colspan="3" align="center"><i class="fa fa-print fa-2x" aria-hidden="true"
style="cursor:pointer" OnClick="CallPrint(this.value)"></i></td>

</tr>

<!-- && $result->status !=0 -->

<?php } else {

?>
```

STUDENT RESULT MANAGEMENT SYSTEM

```
<div class="alert alert-warning left-icon-alert" role="alert">
```

```
<strong>Notice!</strong> Your result not declare yet
```

```
<?php
```

```
}
```

```
?>
```

```
</div>
```

```
<?php
```

```
} else { ?>
```

```
<div class="alert alert-danger left-icon-alert" role="alert">
```

```
strong>Oh snap!</strong>
```

```
<?php
```

```
echo htmlentities("Invalid Roll Id");
```

```
echo "<script>if (!window.Notification) {
```

```
console.log('Browser does not support notifications.');
```

```
} else {
```

```
// check if permission is already granted
```

```
if (Notification.permission === 'granted') {
```

```
// show notification here
```

```
var notify = new Notification('OOPS!', {
```

```
body: 'Invalid Roll Id',
```

```
icon: 'https://thumbs.dreamstime.com/z/invalid-red-stamp-white-background-finger-  
hand-pointing-52829258.jpg',});
```

```
} else {
```

```
// request permission from user
```

STUDENT RESULT MANAGEMENT SYSTEM

```
Notification.requestPermission().then(function(p) {

if(p === 'granted') {

// show notification here

} else {

console.log('User blocked notifications.');
```



```
}

}).catch(function(err) {

console.error(err);

});

}

}</script>";

}

?>

</div>

</tbody>

</table>

</div>

</div>

<!-- /.panel -->

</div>

<!-- /.col-md-6 -->

<div class="form-group">

<div class="col-sm-6">

<a href="index.php">Back to Home</a>
```

```
</div>

</div>

</div>

<!-- /.row -->

</div>

<!-- /.container-fluid -->

</section>

<!-- /.section -->

</div>

<!-- /.main-page -->

</div>

<!-- /.content-container -->

</div>

<!-- /.content-wrapper -->


</div>

<!-- /.main-wrapper -->


<!-- ===== COMMON JS FILES ===== -->

<script src="js/jquery/jquery-2.2.4.min.js"></script>

<script src="js/bootstrap/bootstrap.min.js"></script>

<script src="js/pace/pace.min.js"></script>

<script src="js/lobipanel/lobipanel.min.js"></script>

<script src="js/iscroll/iscroll.js"></script>
```

```
<!-- ===== PAGE JS FILES ===== -->
```

```
<script src="js/prism/prism.js"></script>
```

```
<!-- ===== THEME JS ===== -->
```

```
<script src="js/main.js"></script>
```

```
<script>
```

```
$(function($) {
```

```
});
```

```
function CallPrint(strid) {
```

```
var prtContent = document.getElementById("exampI");
```

```
var WinPrint = window.open("", "
```

```
'left=0,top=0,width=800,height=900,toolbar=0,scrollbars=0,status=0');
```

```
WinPrint.document.write(prtContent.innerHTML);
```

```
WinPrint.document.close();
```

```
WinPrint.focus();
```

```
WinPrint.print();
```

```
WinPrint.close();
```

```
}
```

```
</script>
```

```
</script>
```

```
</body>
```

```
</html>
```

6.2 Stored Procedures

```

<!DOCTYPE html>

<html>

<head>

<title>PHP MySQL stored procedure demo</title>

<link rel="stylesheet" href="css/table.css" type="text/css" />

</head>

<body bgcolor='429456'>

<?php

$host = 'localhost';

$dbname = 'result';

$username = 'admin';

$password = "";

$con=mysql_connect("localhost","root","");

If(!$con)

{

die('Could not connect: '.mysql_error());

}mysql_select_db("result",$con);

print "<h2 align='center'>Creation of Stored Procedure that return a result set in
PHP</h2>";

print "<h2 align='center'><font color='985643'>PROCEDURE:</font><br/><font
color='yellow'>CREATE PROCEDURE Getsol()<br/>BEGIN <br/>SELECT sname,
result</font></h2>

$sql1 = "CREATE PROCEDURE Getsol() BEGIN SELECT sname, resultFROM student;

END;";

```


STUDENT RESULT MANAGEMENT SYSTEM

```
mysql_query($sql1,$con);
```

```
</body>
```

```
</html>
```

6.3 Trigger

```
CREATE TRIGGER 'deleteLog'
```

```
BEFORE DELETE ON 'tblstudents'
```

```
FOR EACH ROW
```

```
INSERT INTO logs
```

```
VALUES(null, OLD.RollId,OLD.ClassId, "Deleted", NOW());
```

Results

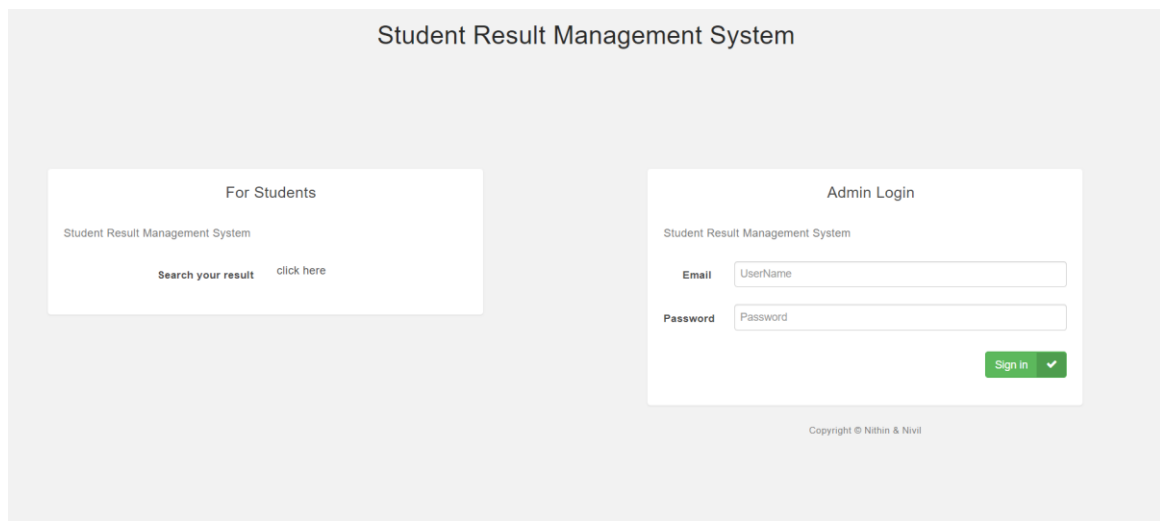


Figure 7.1 Index Page

This is the index page of Student Result Management System where in Admin and students can login.

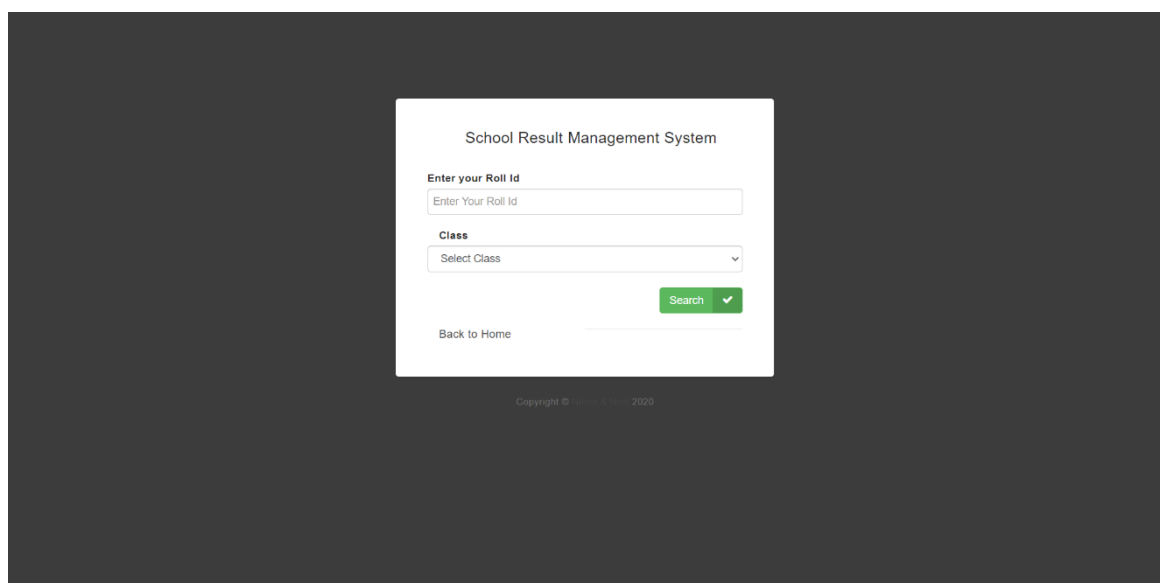


Figure 7.2 Student Login

This is the student login page where in the student can enter the credentials and view the result.

STUDENT RESULT MANAGEMENT SYSTEM

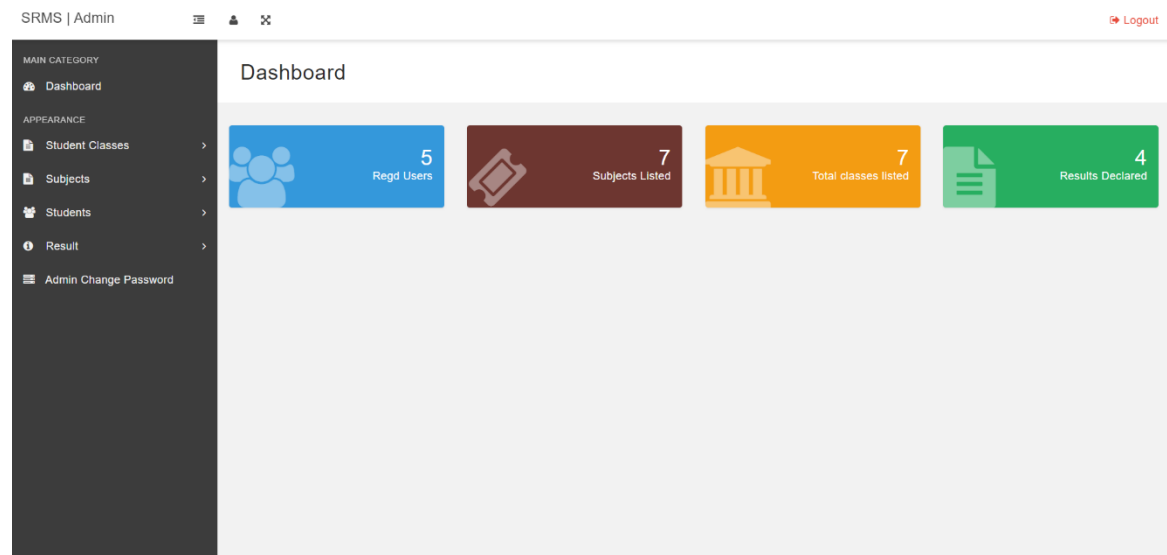


Figure 7.3 Dashboard

This is the dashboard page shown-up after the admin login, where he/she can alter the student details and results.

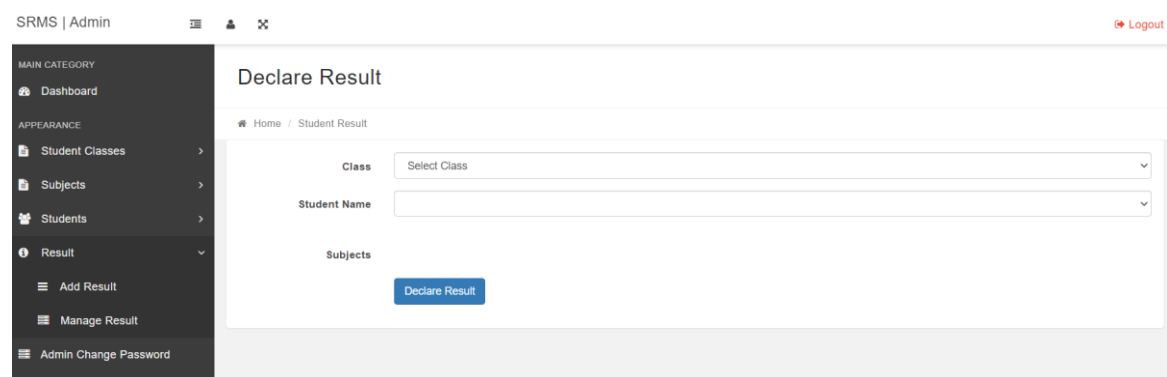


Figure 7.4 Result Declare page

Admin can update the results of the students using this page.

STUDENT RESULT MANAGEMENT SYSTEM

The screenshot shows the 'Create Student Class' page in the SRMS Admin interface. The page has a dark sidebar on the left with a 'MAIN CATEGORY' section containing 'Dashboard' and an 'APPEARANCE' section containing 'Student Classes', 'Subjects', 'Students', 'Result', and 'Admin Change Password'. The main content area is titled 'Create Student Class' and includes a breadcrumb trail 'Home / Classes / Create Class'. The form contains three input fields: 'Class Name' with a hint 'Eg- Third, Fourth, Sixth etc', 'Class Name in Numeric' with a hint 'Eg- 1,2,4,5 etc', and 'Section' with a hint 'Eg- A,B,C etc'. A green 'Submit' button with a checkmark is at the bottom of the form.

Figure 7.5 Create Student Class page

This page allows the admin to create a new class for the students.

The screenshot shows the 'Subject Creation' page in the SRMS Admin interface. The page has a dark sidebar on the left with a 'MAIN CATEGORY' section containing 'Dashboard' and an 'APPEARANCE' section containing 'Student Classes', 'Subjects', 'Students', 'Result', and 'Admin Change Password'. The main content area is titled 'Subject Creation' and includes a breadcrumb trail 'Home / Subjects / Create Subject'. The form contains two input fields: 'Subject Name' and 'Subject Code'. A blue 'Submit' button is at the bottom of the form.

Figure 7.6 Subject creation page

This page allows the admin to create new subject.

STUDENT RESULT MANAGEMENT SYSTEM

SRMS | Admin

Logout

MAIN CATEGORY

- Dashboard

APPEARANCE

- Student Classes
- Subjects
- Students
- Result
- Admin Change Password

Admin Change Password

Home / Admin change password

Admin Change Password

Current Password

New Password

Confirm Password

Change

Figure 7.7 Admin change password page

This page allows the admin to change the password by confirming the current password.

phpMyAdmin

Server: 127.0.0.1 - Database: sms

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking More

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
admin	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 K	-
tblclasses	Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	16.0 K	-
tblresult	Browse Structure Search Insert Empty Drop	16	InnoDB	latin1_swedish_ci	16.0 K	-
tblstudents	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16.0 K	-
tblsubjectcombination	Browse Structure Search Insert Empty Drop	22	InnoDB	latin1_swedish_ci	16.0 K	-
tblsubjects	Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	16.0 K	-
6 tables	Sum	58	InnoDB	utf8mb4_general_ci	96.0 K	0.0

Check all With selected:

Print Data dictionary

Figure 7.8 Database Tables

This is the Database of Student Result Management System.

CONCLUSION

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

- Automation of the entire system improves the efficiency
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications.
- Updating of information becomes so easier.
- System security, data security and reliability are the striking features.
- The System has adequate scope for modification in future if it is necessary.

It has been wonderful experience for us throughout the preparation of this mini project. By implementing the project, we have got the brief ideas about the three tier architecture. We hope that it really helps us in future projects. Any School/Colleges can make use of it for saving Student, class and results in the database. This application can easily implement under various situation. We can add new features as and when we require. Reusability of this application is also possible.

BIBLIOGRAPHY

The following books were referred during the analysis and execution phase of the project

- PHP and MySQL
Web Development Book by Luke Welling
- Head First PHP & MySQL
Book by Lynn Beighley and Michael Morrison
- PHP & MySQL for Dummies
Book by Janet Valade

WEBSITES:

- www.google.com
- www.w3schools.com
- www.tutorialspoint.php
- <http://stackoverflow.com>

Personal Profile

Mrs Suchetha NV,

Asst. Prof.

Project Guide

	<p>Name: Nithin Thomas</p> <p>USN: 4SU18CS052</p> <p>Address: Poovathingal(H), Nidigal post, Belthangady TQ, DK Dist</p> <p>Email: Nithinthomaspoovathingal@gmail.com</p> <p>Contact No.: 9900412213</p>
	<p>Name: Nivil Shibu</p> <p>USN: 4SU18CS053</p> <p>Address: Maliackel(H), Ujire post, Belthangady TQ, DK Dist</p> <p>Email: nivilshibu@gmail.com</p> <p>Contact No.: 9481903998</p>