



湖北工业大学
HUBEI UNIVERSITY OF TECHNOLOGY

DBMS

Course Design Report

Design Topic Education Management System

Mentor Prof. JIANXIA CHEN

Profession SOFTWARE ENGINEERING

Class 20LC

Student ID 2011562107

Name RUDRA MD IFTEKHARUL ISLAM

Date 19-12-2021

Objective and significance of the project

Now a day's education plays a great role in development of any country.

Many of education organizations try to increase education quality. One of the aspects of this improvement is managing of school resources.

The main objectives and significance of Education Management System is given here –

Education Management System (EMS) is such a service which provides all services for an educational institute to make your life easier and faster by assuring its performance. Easy User Management System, Easy Admission Process, Easy Attendance System.

Education Management System carried on by any individual or institution engaged in providing a services to students, teachers, guardians and other persons are intermediary that performs one or more of the following functionalities – Student Admission, Employee Registration, Student List, Employee List, Student Attendance, Employee Attendance, Student Routine, Result Management, Payroll & Accounts.

EMS is a system that will provide you a bird's eye view of the functioning of the entire educational institution. It is a management information system helps to manage the different processes in an educational institution like General Administration, Staff Management,

Academics, Student Management, and Accounts etc. The information is made using the latest technologies and help's to make decision making a lot faster, effective and easier than ever before. Also helps to improve the overall quality of education of the institution.

The difference area where we can use this application as:

- Any education institution makes use of it providing class schedule.
- It can be used in offices and modifications can be easily done according to requirements.

The common features of the projects are:

- This is very easy to use for each user.
- Increase Efficiencies and Reduce Costs
- Transform IT for Higher Education
- Easy Solution
- Easy Admission Process
- Secure All Data
- Easy Account Maintenance
- Transaction History
- Easy Attendance Process
- The user of the database can see all information and also can edit if necessary.
- Easy implemented routine for student and teacher's.

System operation environment description

We use database and database technology are having a major impact on the growing use of computers.

We use C# (.net framework) and MSSQL Server 2012.

C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. You can use C# to create Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework.

The .NET Framework (pronounced dot net) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library known as Framework Class Library (FCL) and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for .NET Framework execute in a software environment (as contrasted to hardware environment), known as Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management,

and exception handling. FCL and CLR together constitute .NET Framework.

Software Requirements

- Microsoft windows XP/ Windows 7/ Windows Vista/ Windows 8/ Windows 10/ Windows Server 2003, 2008, 2012.
- Visual Studio 2012 should be installed.
- .Net framework should be installed, Crystal report should be installed for report view (visual studio package not installed in OS).
- MSSQL Server 2012 should be installed.

C# Features

- C# is a simple, modern, object oriented language derived from C++ and Java.
- It aims to combine the high productivity of Visual Basic and the raw power of C++.
- It is a part of Microsoft Visual Studio7.0.
- Visual studio supports Vb,VC++,C++,Vbscript, Jscript. All of these languages provide access to the Microsoft .NET platform.
- .NET includes a Common Execution engine and a rich class library.
- Microsofts JVM eqiv is Common language run time (CLR).
- CLR accommodates more than one languages such as C#, VB.NET, Jscript, ASP.NET,C ++.
- Source code --->Intermediate Language code (IL) ---> (JIT Compiler) Native code.
- The classes and data types are common to all of the .NET languages.
- We may develop Console application, Windows application, and Web application using C #.
- In C# Microsoft has taken care of C++ problems such as Memory

management, pointers etc.

- It supports garbage collection, automatic memory management and a lot.

Database Platform

A database is an organized collection of data. The data is typically organized to model aspects of reality in a way that supports processes requiring information, such as modeling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

Database management systems (DBMS) are computer software applications that interact with the user, other applications, and the database itself to capture and analyze data. A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase and IBM DB2.

SQL Server database Introduction

In computing, Microsoft SQL Server is a relational database management system, currently developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network (including the Internet).

SQL Server database system features

- Supports most administrative tasks for SQL Server.
- A single, integrated environment for SQL Server Database Engine management and authoring.
- Dialogs for managing objects in the SQL Server Database Engine, Analysis Services, and Reporting Services, that allows you to execute your actions immediately, send them to a Code Editor, or script them for later execution.

- Non-modal and resizable dialogs allow access to multiple tools while a dialog is open.
- A common scheduling dialog that allows you to perform action of the management dialogs at a later time.
- Exporting and importing SQL Server Management Studio server registration from one Management Studio environment to another.

System requirements analysis

Systems requirements analysis is a problem solving technique that decomposes a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose. As the software system requirements were predictable, it is decided to follow the classical system development life cycle method. This process demands a systematic, sequential approach to software development that begins at the system level and progress through analysis, design, coding, testing and maintenance. The steps that is applicable to all software engineering paradigms. The program is followed by SDLC (Software Development Life Cycle)

System Analysis

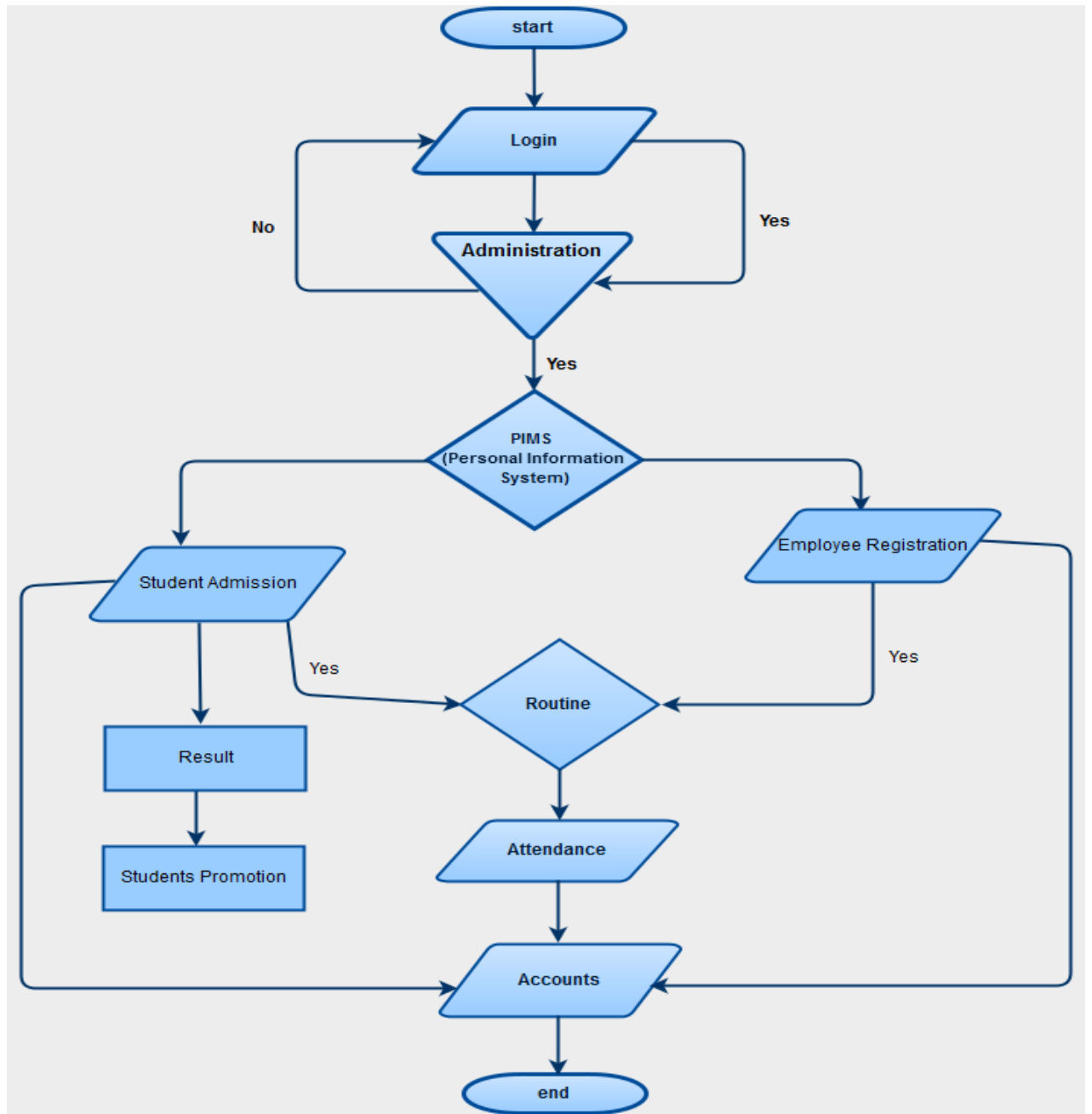
Analysis involves the requirement determination and specification. Systems analysis is a problem solving technique that

decomposes a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose. According to the Merriam-Webster dictionary, systems analysis is the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way. Analysis and synthesis, as scientific methods, always go hand in hand, they complement one another.

Requirement Analysis

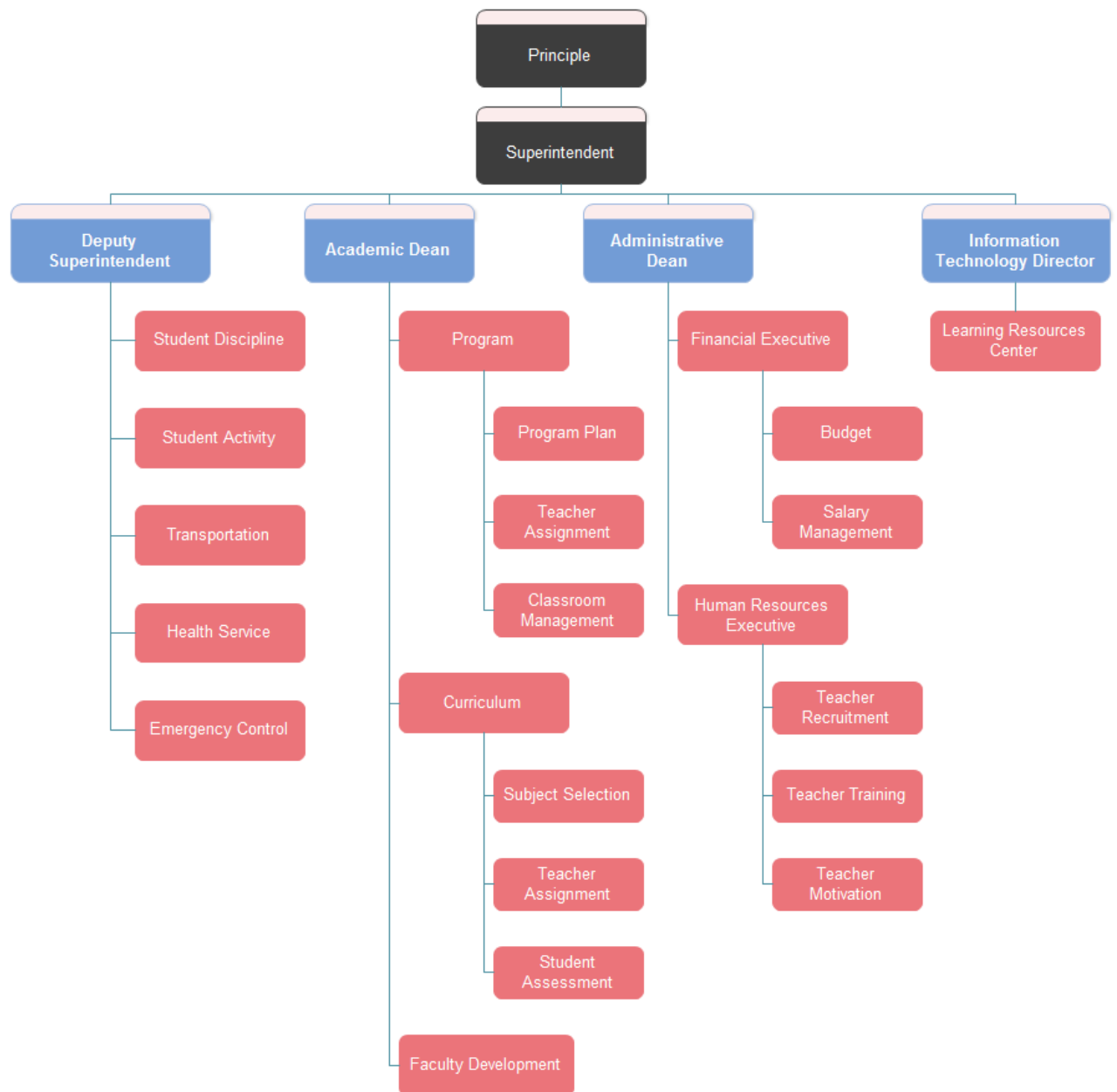
Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.

Requirements analysis is critical to the success of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.



Organization chart

An effective school organizational structure involves better work distribution, efficient management control, and good information processing.

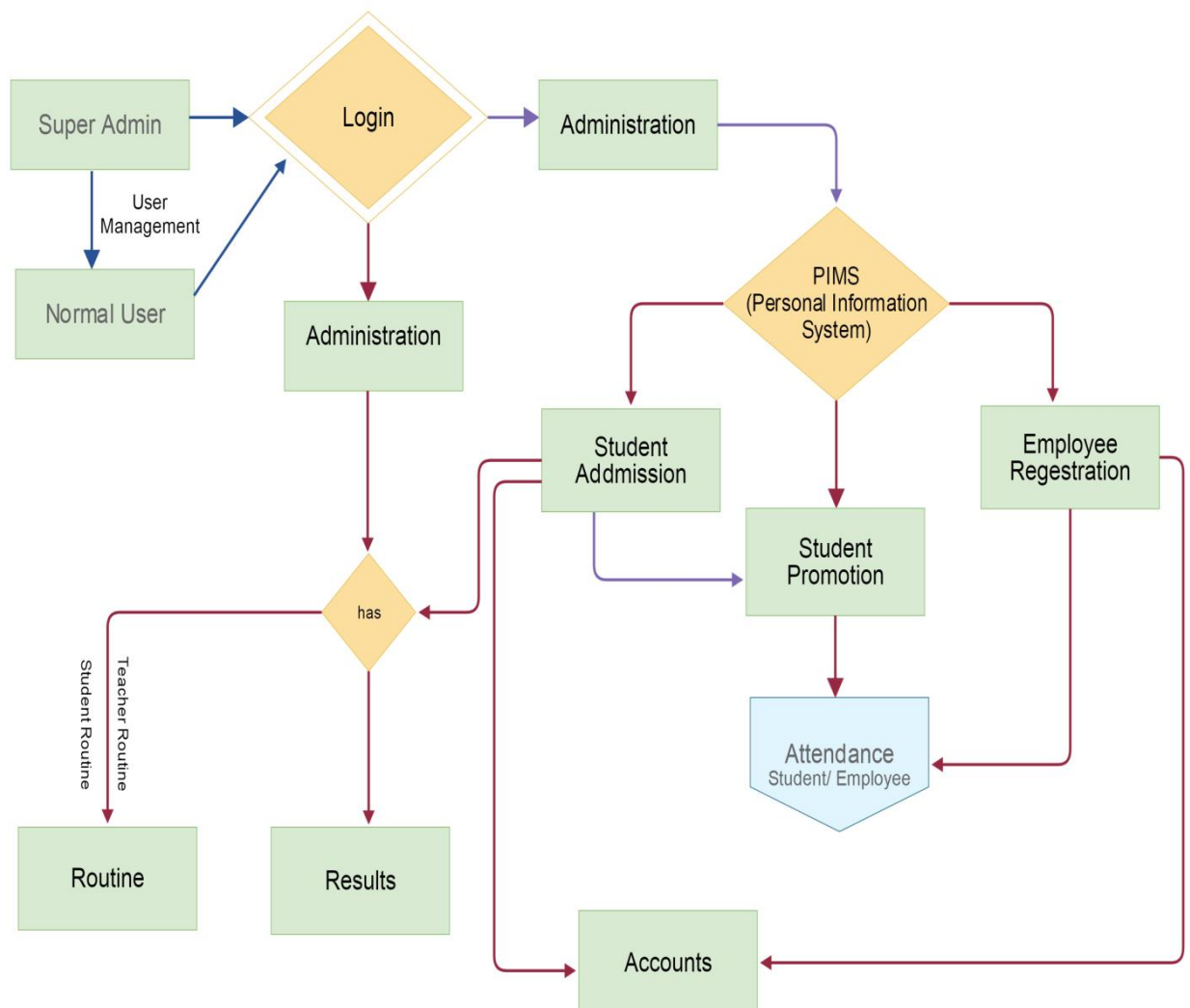


- **Principle:** Involves vision, forward planning, foresight, strategy, school development planning, etc.
- **Superintendent:** Involves with communication, general administration, principle implementation, etc.
- **Deputy Superintendent:** Involves with student discipline, student activity organization, transportation, health care, emergency preparedness, etc.
- **Academic Dean:** Involves program plan development, teachers and teaching resources distribution, classroom management, subject selection, student

assessment, etc.

- **Administrative Dean:** Including two parts – the financial department and human resources department. It involves a budget plan, teacher recruitment, teacher training, and staff motivation.
- **Information and Technology center** involves teaching resources management, purchasing, technical equipment maintenance.

Data flow graph and data dictionary



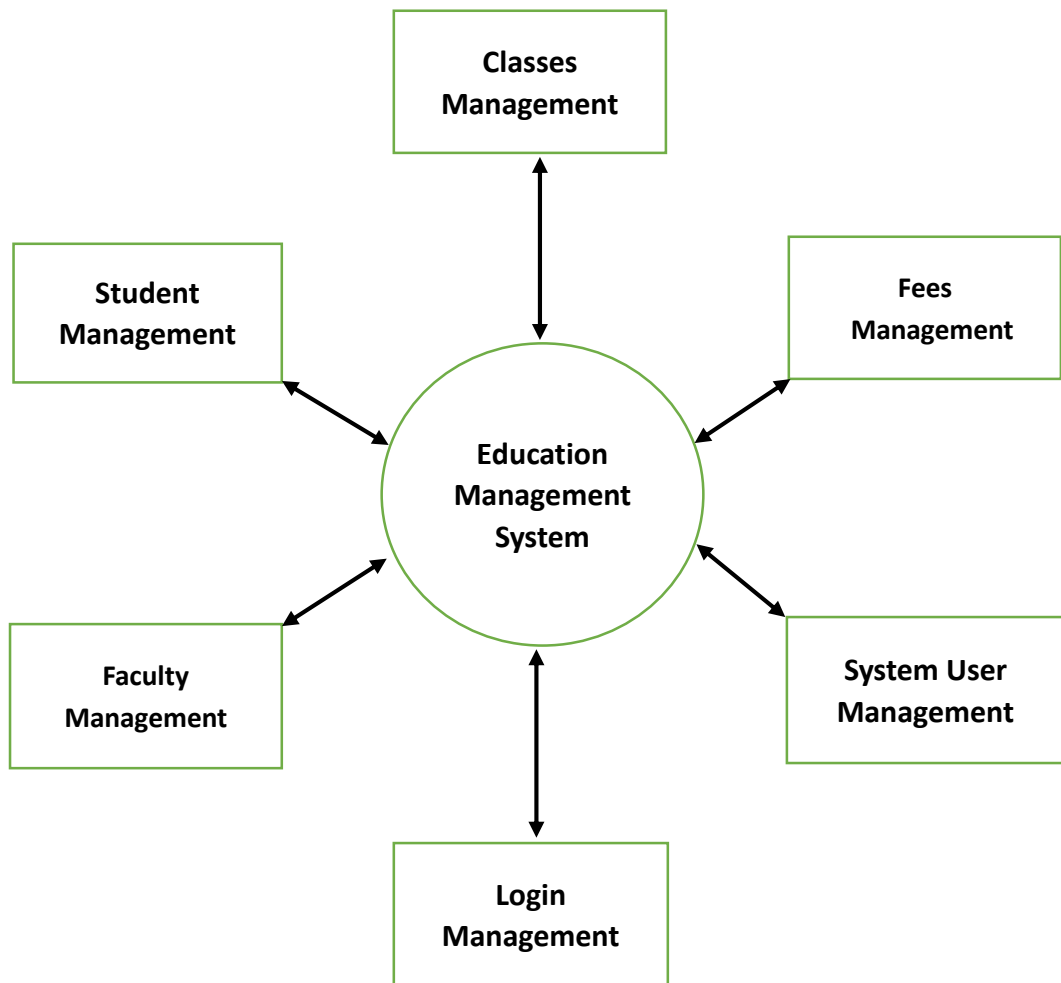
Education Management System Data flow Diagram is often used as a preliminary step to create an overview of the School without going into detail, which can later be elaborated. It normally consists of overall application data flow and processes of the School process. It contains all of the userflow and their entities such all the flow of students, class, Teachers, Timetable, Fees, Attendance, Subjects, All of the bellow diagrams has been used for the visualization of data processing and structured design of the School process and working flow.

Zero Level Dataflow Diagram –

This is the Zero Level DFD of Education/School Management System, where we have elaborated the high level process of school. It's a basic overview of the whole School Management System or process being analyzed or modeled. It's designed to be an at-a-glance view of Fees, Attendance and Subjects showing the system as a single high-level process, with its relationship to external entities of Students, Class and Teachers. It should be easily understood by a wide audience, including Students, Teachers and Fees in zero level DFD of Education Management System, we have described the high level flow of the Education System.

- Managing all the Students
- Managing all the Class
- Managing all the Teachers

- Managing all the Timetable
- Managing all the Fees
- Managing all the Attendance
- Managing all the Subjects

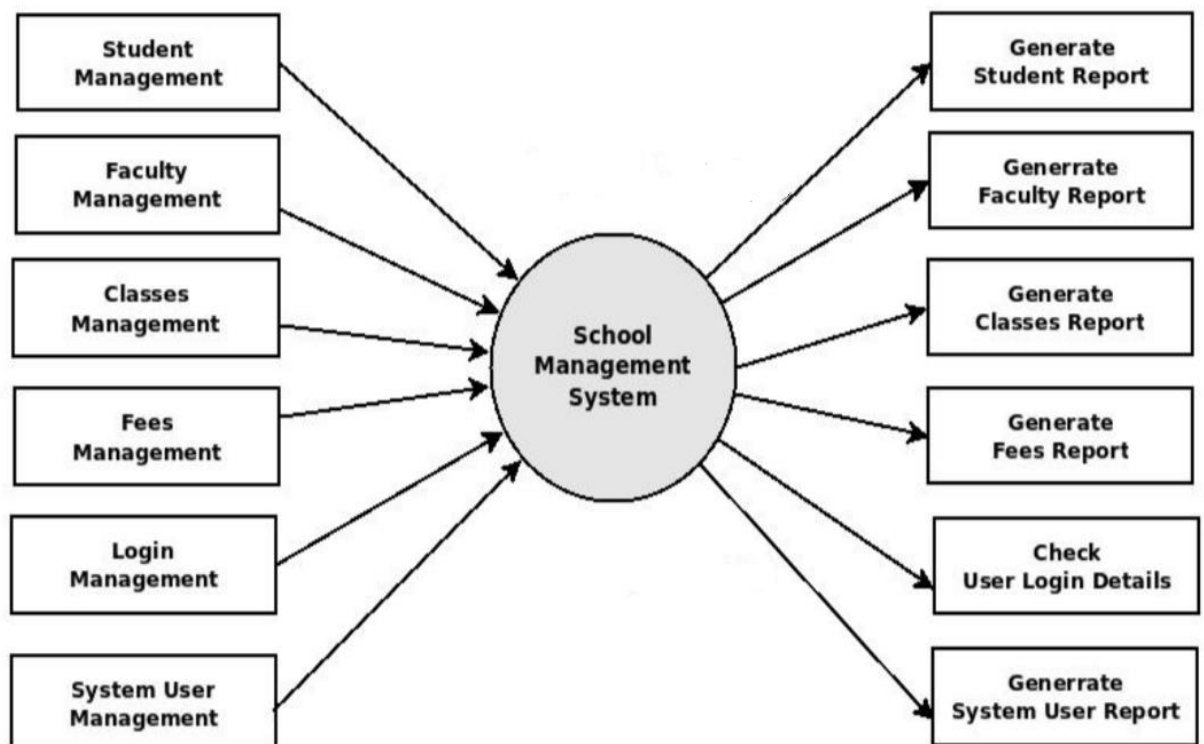


First Level Dataflow Diagram –

First Level DFD of Education Management system shows how the system is divided into sub-systems, each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the Education Management System as a

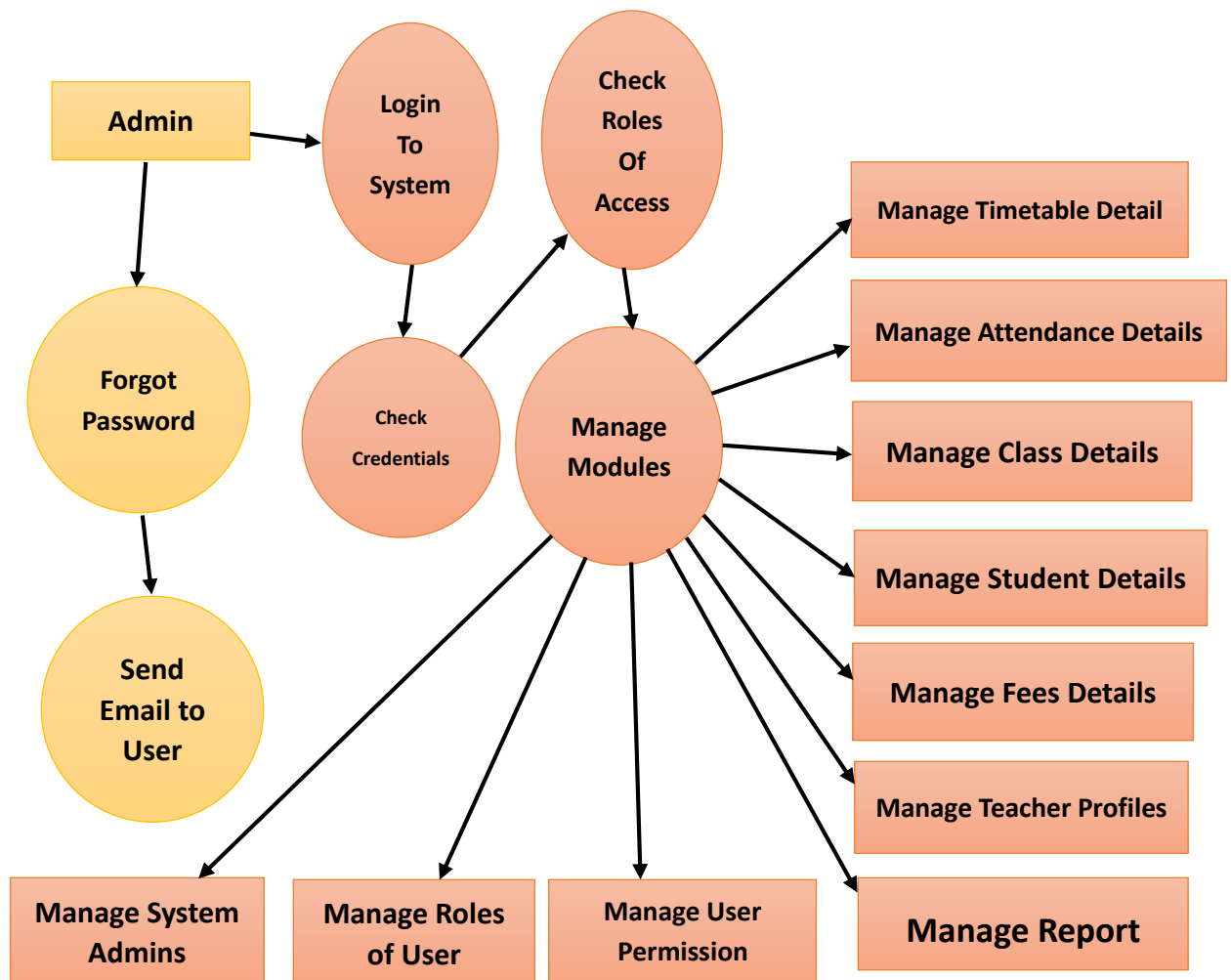
whole. It also identifies internal data stores of Subjects, Attendance, Fees, Timetable, Teachers that must be present in order for the Education System to do its job, and shows the flow of data between the various parts of Students, Teachers, Attendance, Subjects, Fees of the system. DFD Level 1 provides a more detailed breakout of pieces of the 1st level DFD.

- Processing Students records and generate report of all Students
- Processing Class records and generate report of all Class
- Processing Teachers records and generate report of all Teachers
- Processing Timetable records and generate report of all Timetable
- Processing Fees records and generate report of all Fees
- Processing Attendance records and generate report of all Attendance
- Processing Subjects records and generate report of all Subjects



Second Level Dataflow Diagram –

DFD Level 2 then goes one step deeper into parts of Level 1 of School. It may require more functionalities of School to reach the necessary level of detail about the School functioning. First Level DFD of Education Management System shows how the system is divided into sub-systems. The 2nd Level DFD contains more details of Subjects, Attendance, Fees, Timetable, Teachers, Class, Students.



System function structure

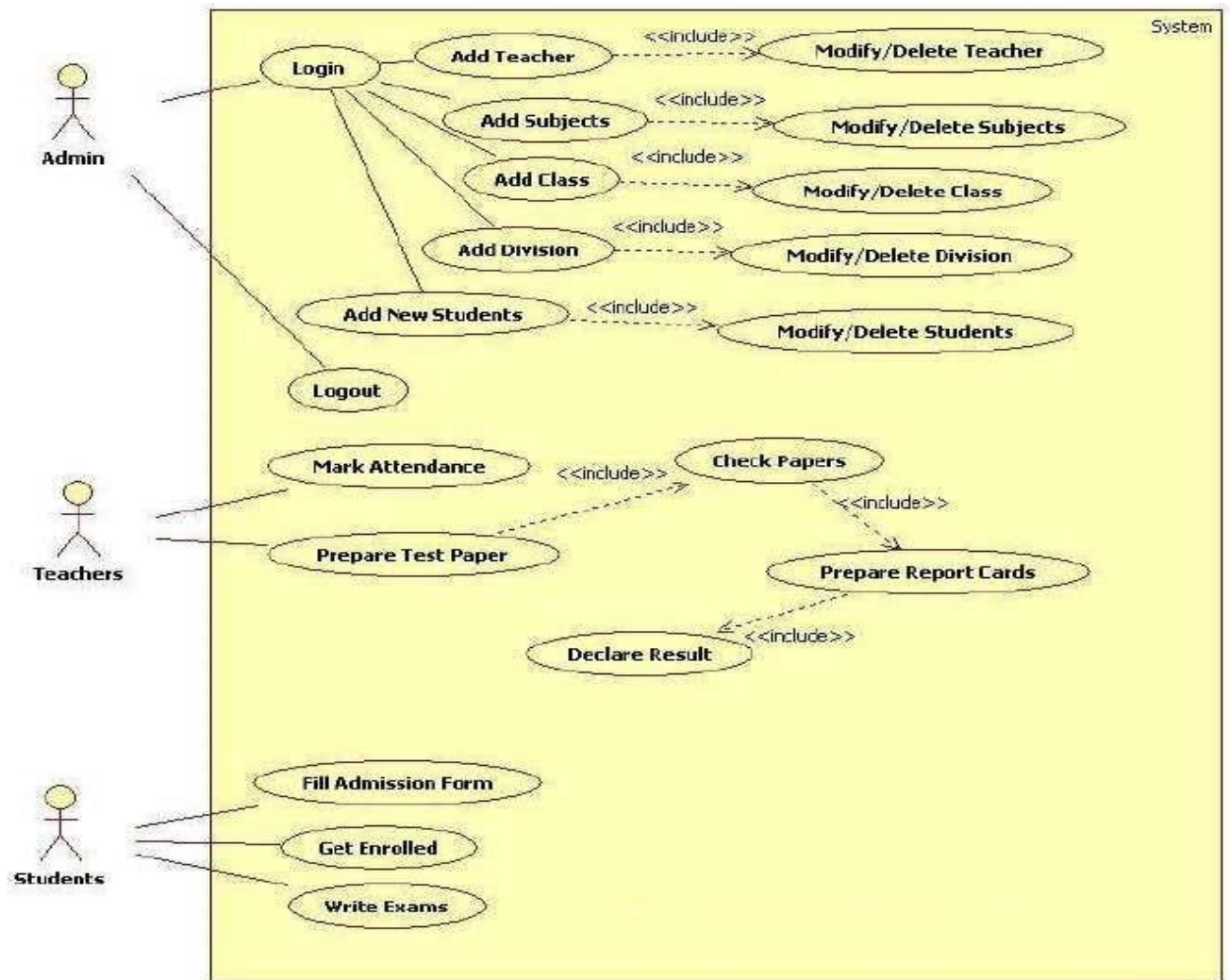
Use case diagram –

UML Use Case diagram for Education/School Management System is shown below. The various participants of the same are detailed below:-

Actors: - Admin, Teacher, Students

The corresponding use cases for these actors are:-

- **Admin:** Login, Logout, Add Teachers, Add Students, Add Class, Add Divisions, Add Subjects, Modify/Delete Teachers, Modify/Delete Students, Modify/Delete Class, Modify/Delete Divisions, Modify/Delete Subjects.
- **Teacher:** Mark Attendance, Prepare Test Paper, Check Papers, Prepare Report Cards, Declare Result.
- **Student:** Fill Admission Form, Get Enrolled, Write Exams The Use Case UML diagram for School Management System is shown below:-



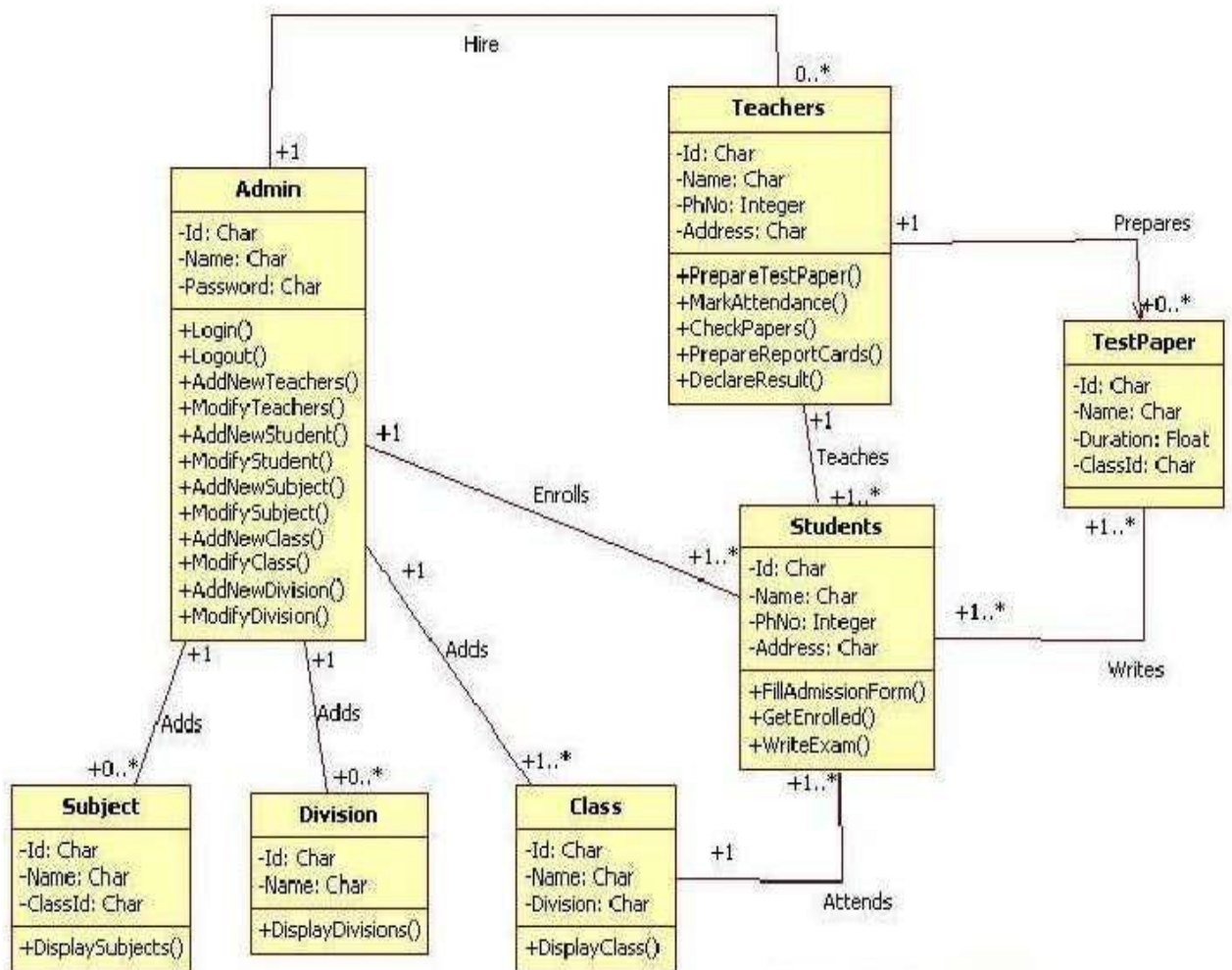
Class diagram –

UML Class diagram for School Management System is shown below.

The various Classes involved in the system are:

Classes: Admin, Students, Teachers, Class, Subjects, Divisions, Test Paper.

The Class Diagram for School Management System is shown below:

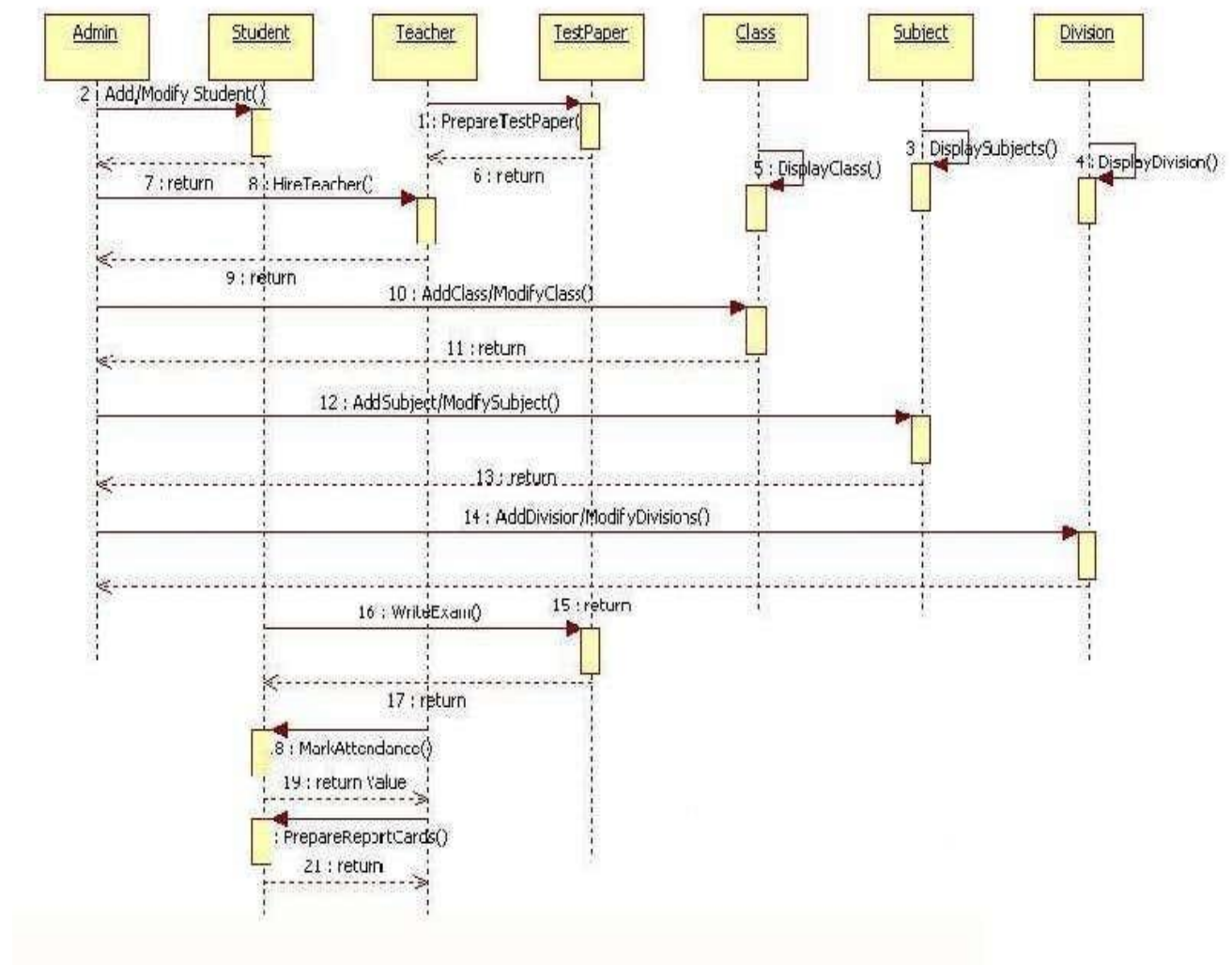


Sequence diagram –

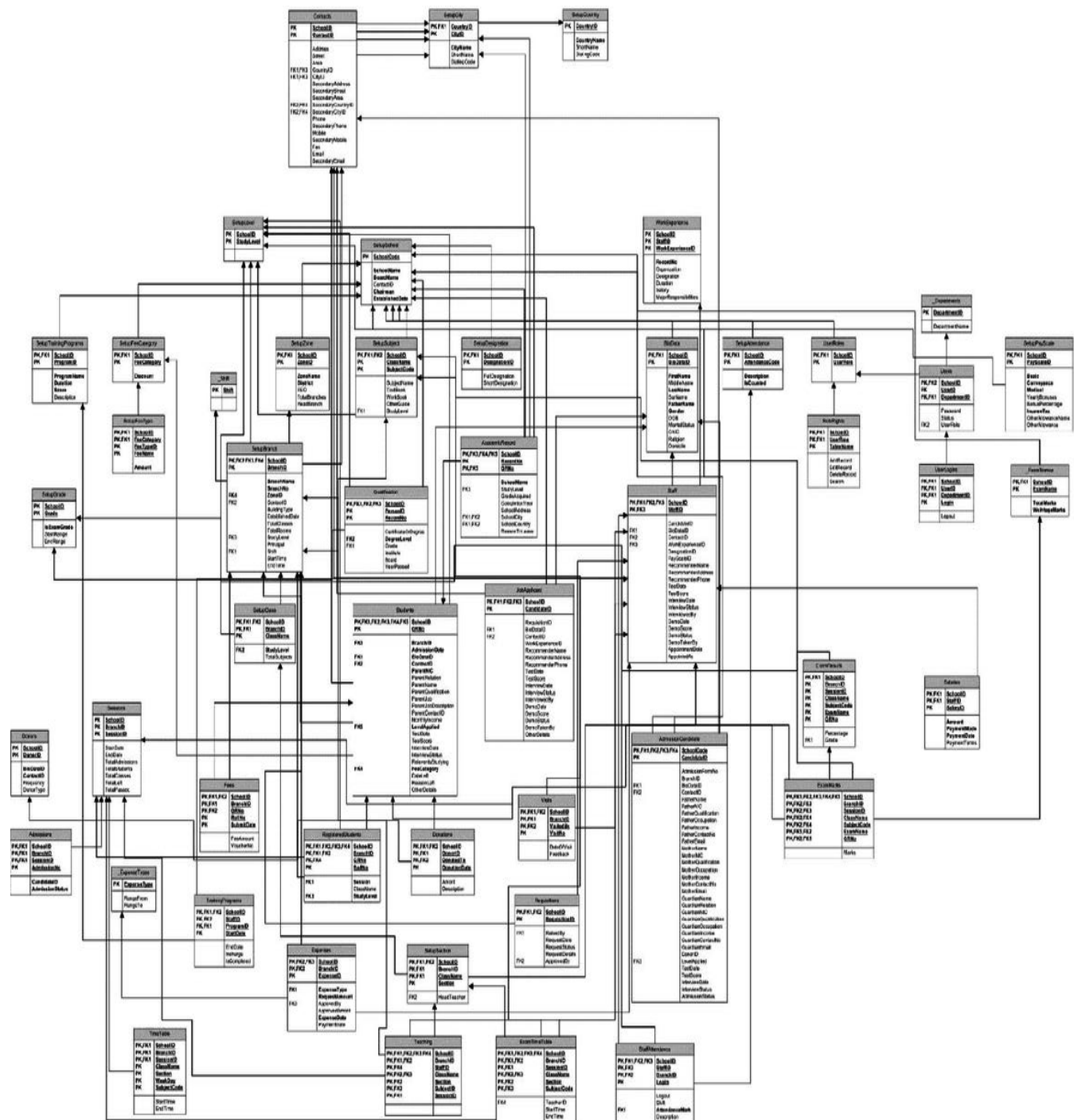
UML Sequence diagram for School Management System is shown below.

The instance of class objects involved in this UML Sequence Diagram are as follows:-

Admin, Teacher, Test Paper. Student, Class, Subject, Division



System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages.



Database design

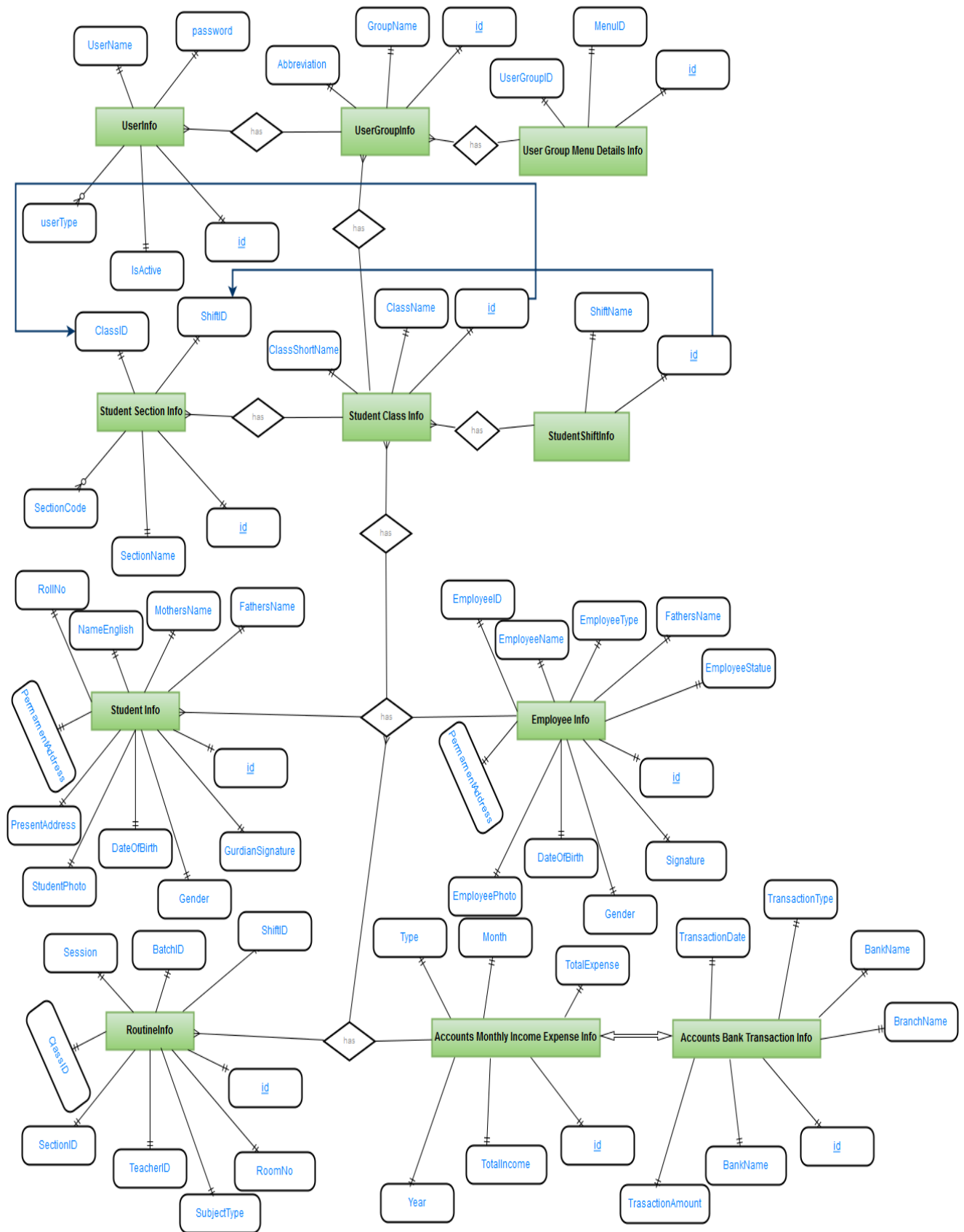
A database management system (DBMS) is a collection of programs that enables you to store, modify, and extract information from a database. There are many different types of database management systems, ranging from small systems that run on personal computers to huge systems that run on mainframes.

DBMS is a software that handles the storage, retrieval, and updating of data in a computer system.

Ex- SQL Server (Microsoft), MySQL (Freeware), Oracle (Oracle), NoSQL (Oracle), NonStop SQL (Hewlett Packard).

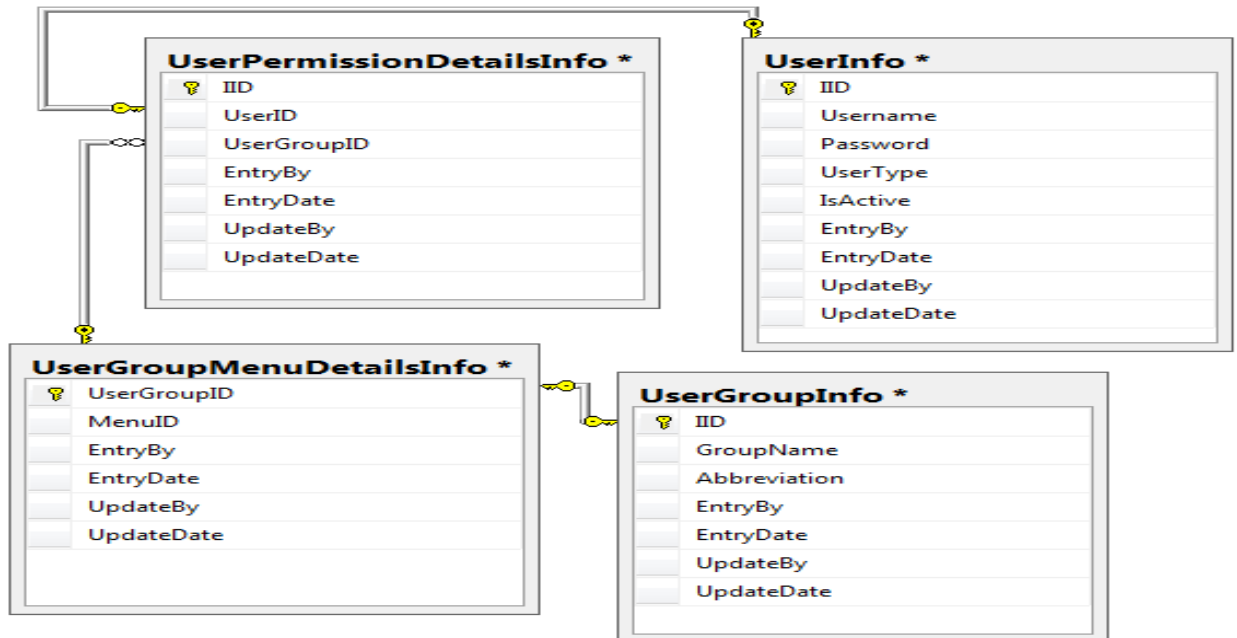
Conceptual design (E-R diagram)

The E-R Diagram is given bellow –

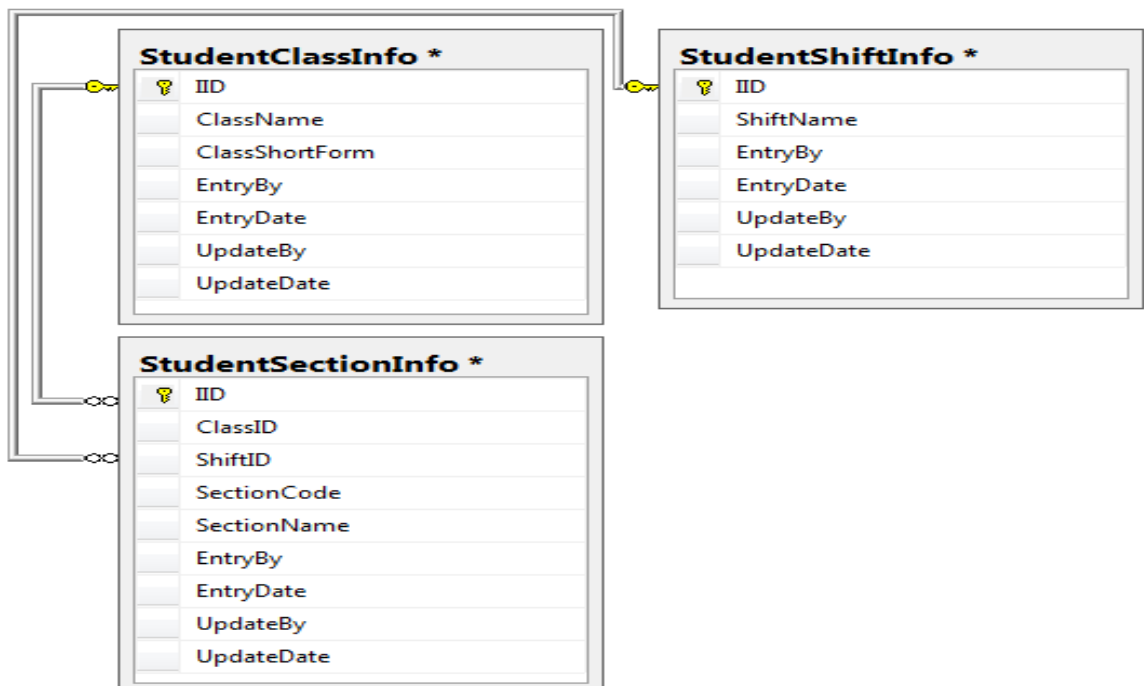


Logical structure design

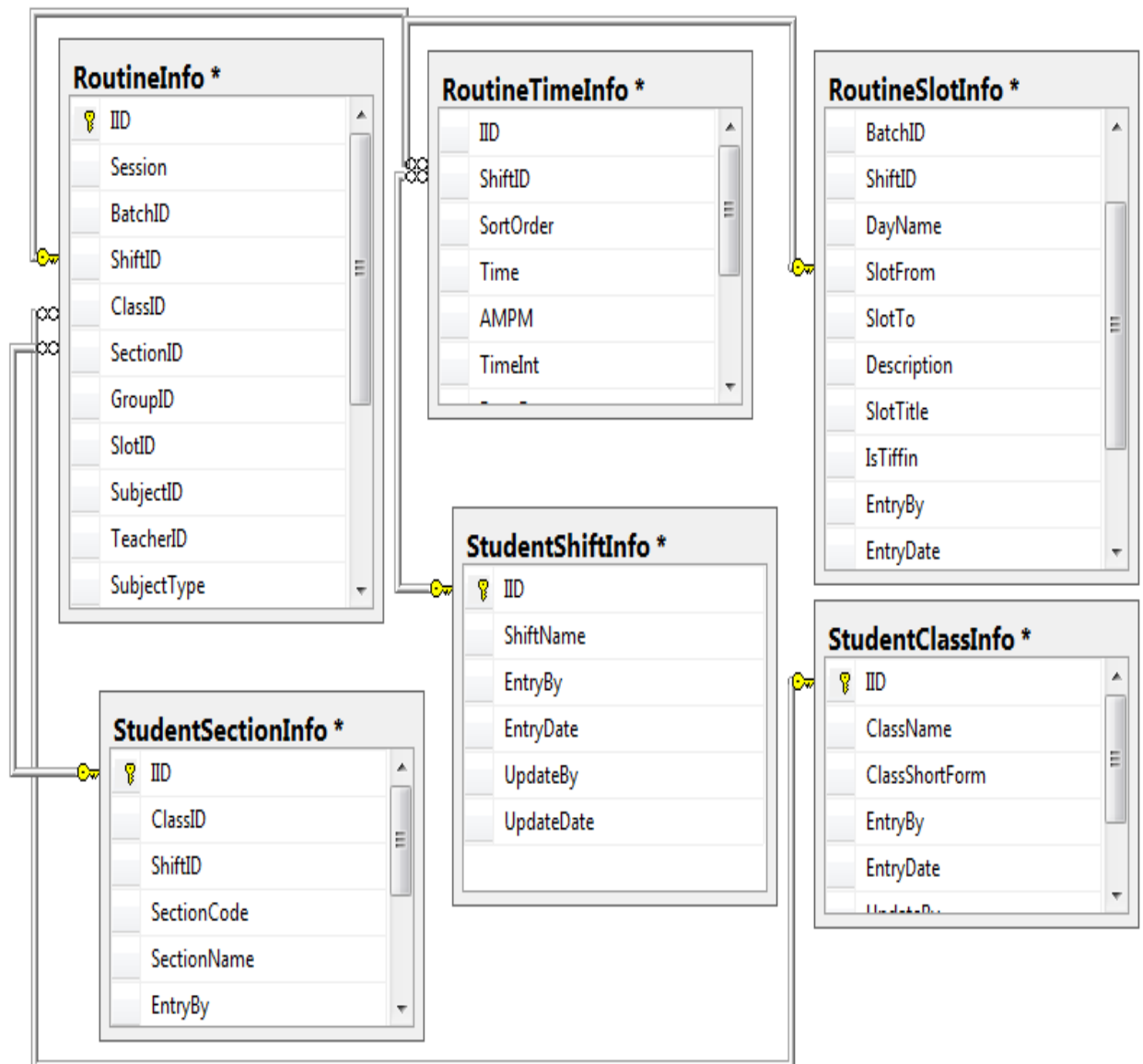
User -



Administration -

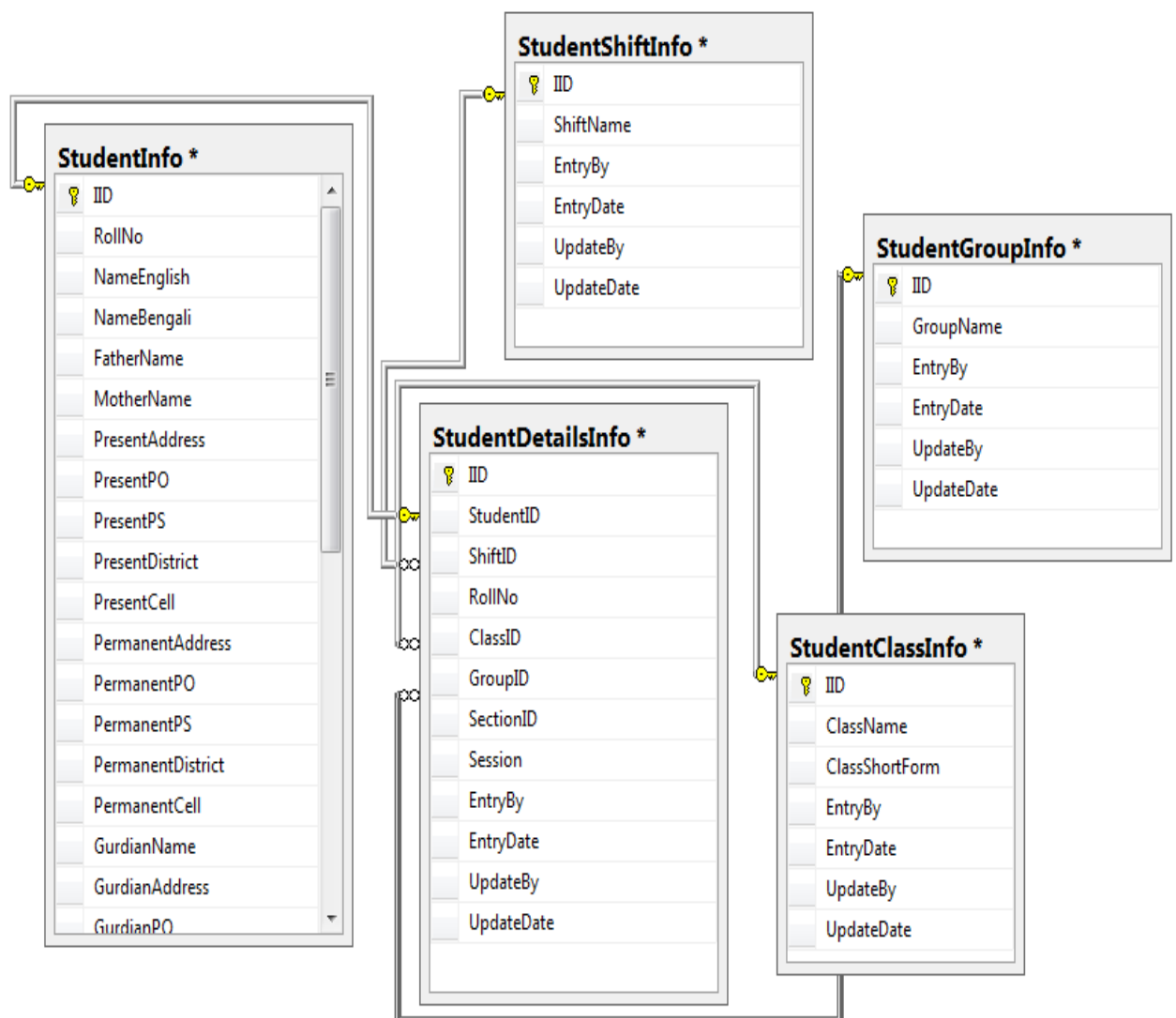


Routine –

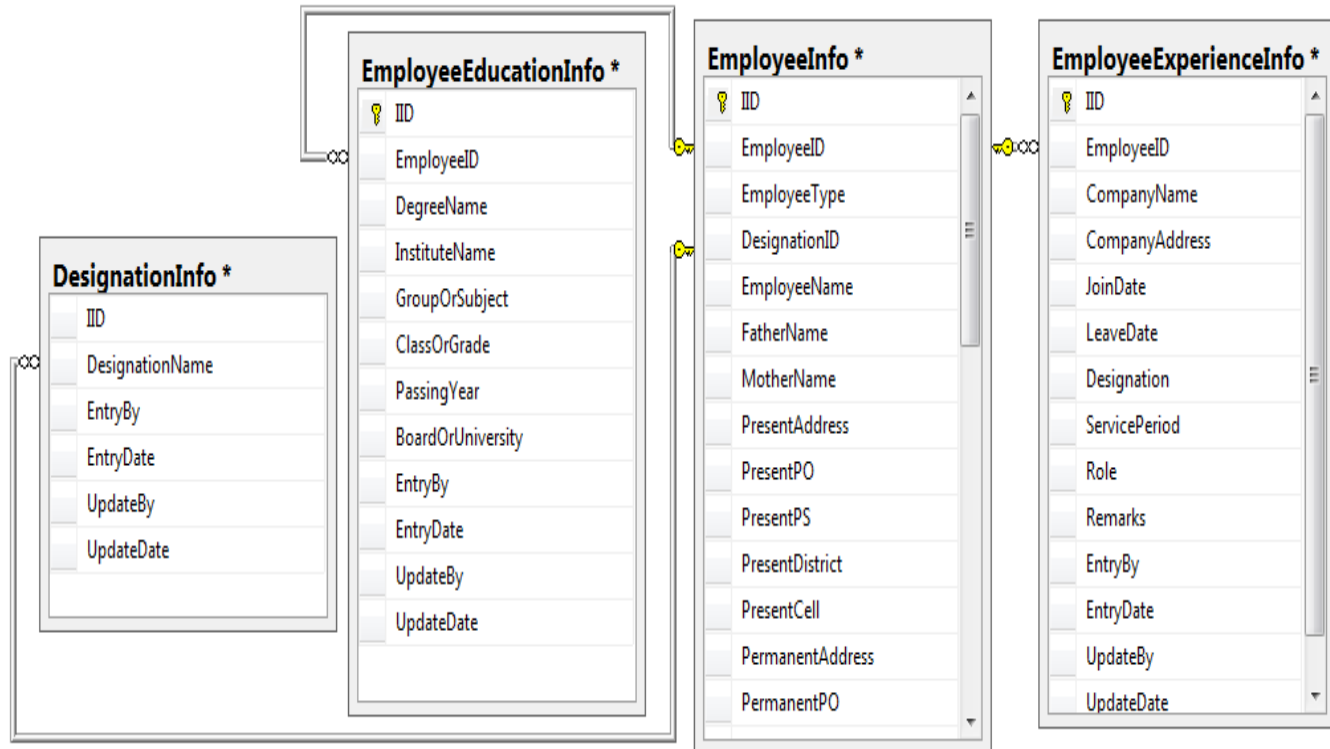


Physical structure design

Student –



Employee –



Database implementation and operation

Implementation refers to the final process of moving the solution from development status to production status. Depending on your project, this process is often called deployment, go-live, rollout or installation. For the purposes of Life cycle Step, all of these terms are synonymous with "implementation."

There is no single way to implement an application. It depends on the characteristics of your project and the solution. Some implementations are as easy as saying "we are now live." This type of implementation can

work when the solution is brand new and you are developing and testing in what will become the production environment. In these cases, implementation is just a state of mind. One day the solution is in development, and the next day it is in production.

Introduction of database management system

DBMS stands for **Database Management System**. We can break it like this DBMS = Database + Management System. Database is a collection of data and Management System is a set of programs to store and retrieve those data. Based on this we can **define DBMS** like this: DBMS is a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.

Database systems are basically developed for large amount of data.

When dealing with huge amount of data, there are two things that require optimization: **Storage of data** and **retrieval of data**.

Storage: According to the principles of database systems, the data is stored in such a way that it acquires lot less space as the redundant data (duplicate data) has been removed before storage. Let's take a layman example to understand this:

In a banking system, suppose a customer is having two accounts, one is

saving account and another is salary account. Let's say bank stores saving account data at one place (these places are called tables we will learn them later) and salary account data at another place, in that case if the customer information such as customer name, address etc. are stored at both places then this is just a wastage of storage (redundancy/ duplication of data), to organize the data in a better way the information should be stored at one place and both the accounts should be linked to that information somehow. The same thing we achieve in DBMS.

Fast Retrieval of data: Along with storing the data in an optimized and systematic manner, it is also important that we retrieve the data quickly when needed. Database systems ensure that the data is retrieved as quickly as possible.

Database creation and data entry

Backstage view is a starting point from which you can create a new database, open an existing database, view featured content from Office.com — anything you can use Access to do *to* a database file or *outside* of a database, as opposed to *within* a database.

Application parts are a feature that let you use several related database objects together as if they were one. For example, an application part might consist of a table and a form that is based on the table. You can add

the table and the form at the same time by using the application part.

Access comes with a variety of templates that you can use as-is or as a starting point. A template is a ready-to-use database that contains all the tables, queries, forms, macros, and reports needed to perform a specific task. For example, there are templates that you can use to track issues, manage contacts, or keep a record of expenses. Some templates contain a few sample records to help demonstrate their use.

If one of these templates fits your needs, using it is usually the fastest way to get a database started. However, if you have data in another program that you want to import into Access, you might decide it is better to create a database without using a template. Templates have a data structure already defined, and it might require a lot of work to adapt your existing data to the template's structure.

1. If you have a database open, on the **File** tab, click **Close**. Backstage view displays the **New** tab.
2. Several sets of templates are available in the **New** tab, some of which are built into Access. You can download additional templates from Office.com. See the next section in this article for details.
3. Select the template that you want to use.
4. Access suggests a file name for your database in the **File Name** box — you can change the file name, if you want. To save the database in a different folder from the one displayed below the

file name box, browse to the folder in which you want to save it, and then click **OK**. Optionally, you can create and link your database to a SharePoint site.

5. Click **Create**.

Access creates a database from the template that you chose, and then opens the database. For many templates, a form is displayed in which you can begin entering data. If your template contains sample data, you can delete each record by clicking the record selector (the shaded box or bar just to the left of the record), and then doing the following:

On the **Home** tab, in the **Records** group, click **Delete**. 

6. To begin entering data, click in the first empty cell on the form and begin typing. Use the Navigation Pane to browse for other forms or reports that you might want to use. Some templates include a navigation form which allows you to move between the different database objects.

To enter data in a table, you start with the **INSERT** combined with the **VALUES** keywords. The statement uses the following syntax:

```
INSERT TableName VALUES(Column1, Column2, Column_n)
```

Alternatively, or to be more precise, you can specify that you are entering data in the table using the **INTO** keyword between the **INSERT** keyword and the **TableName** factor. This is done with the following syntax:

INSERT INTO TableName VALUES(Column1, Column2, Column_n)

The TableName factor must be a valid name of an existing table in the currently selected database. If the name is wrong, the SQL interpreter would simply consider that the table you are referring to doesn't exist. Consequently, you would receive an error.

The VALUES keyword indicates that you are ready to list the values of the columns. The values of the columns must be included in parentheses. Specify the value of each column in the parentheses that follow the VALUES keyword.

If the column is a BIT data type, you must specify its value as 0 or 1.

If the column is a numeric type, you should pay attention to the number you type. If the number is an integer, you should provide a valid natural number without the decimal separator. If the column is for a decimal number, you can type the value with its character separator (the period for US English).

If the data type of a column is a string type, you should include its entry between single quotes. For example, a shelf number can be specified as

'HHR-604' and a middle initial can be given as 'D'.

Database query

A database query is either an action query or a select query. A select query is one that retrieves data from a database. An action query asks for additional operations on data, such as insertion, updating, deleting or other forms of data manipulation.

This doesn't mean that users just type in random requests. For a database to understand demands, it must receive a query based on the predefined code. That code is a query language.

Queries can accomplish a few different tasks. Primarily, queries are used to find specific data by filtering explicit criteria. Queries also help automate data management tasks, summarize data and engage in calculations.

Other examples of queries include append, crosstab, delete, make a table, parameter, totals and updates.

Meanwhile, a query parameter runs variations of a particular query, prompting users to insert a field value and then use that value to create a criterion. Totals queries, on the other hand, allow users to group and summarize data.

SQL Query [?] [X]

Select all records in the current working table where... [Add More Criteria...](#)

Table Name:	Field Name:	Operator:	Condition:
Current Browser	Station_ID	Like	Buford Dam%
Stations	Station_Name	In	
PCodes	Station_Type	=	
CCodes	Data_Freq	<	
RCodes	Latitude	<=	
StaGrp	Longitude	>	
PCodeGrp	Elevation	>=	

[Query syntax is valid](#) ✓

Resulting SQL Statement (you can edit this before execution if you wish):

```
SELECT Dta.* FROM [TESTDAT1] Dta INNER JOIN Stations ON Dta.Station_ID = Stations.Station_ID
WHERE Stations.Station_Name Like 'Buford Dam%'
```

Select from browser and create new result set

☐ Display available values in condition drop-down

[Save...](#) [Select All](#) [Execute](#) [Close](#)

Database control

Database control refers to the task of enforcing regulations so as to provide correct data to authentic users and applications of a database. In order that correct data is available to users, all data should conform to the integrity constraints defined in the database. Besides, data should be screened away from unauthorized users so as to maintain security and privacy of the database. Database control is one of the primary tasks of the database administrator (DBA).

The three dimensions of database control are –

- Authentication

- Access rights
- Integrity constraints

Program design description

Design patterns are solutions to software design problems you find again and again in real-world application development. Patterns are about reusable designs and interactions of objects.

The 23 Gang of Four (GoF) patterns are generally considered the foundation for all other patterns. They are categorized in three groups: Creational, Structural, and Behavioral (for a complete list see below).

To give you a head start, the C# source code for each pattern is provided in 2 forms: structural and real-world. Structural code uses type names as defined in the pattern definition and UML diagrams. Real-world code provides real-world programming situations where you may use these patterns.

A third form, .NET optimized, demonstrates design patterns that fully exploit built-in .NET 4.5 features, such as, generics, attributes, delegates, reflection, and more. These and much more are available in our [.NET Design Pattern Framework 4.5](#).

This structural code demonstrates the Singleton pattern which assures only a single instance (the singleton) of the class can be created.

This real-world code demonstrates the Singleton pattern as a Load

Balancing objects. Only a single instance (the singleton) of the class can be created because servers may dynamically come on-or off-line and every request must go through the one object that has knowledge about the state of the (web) farm.

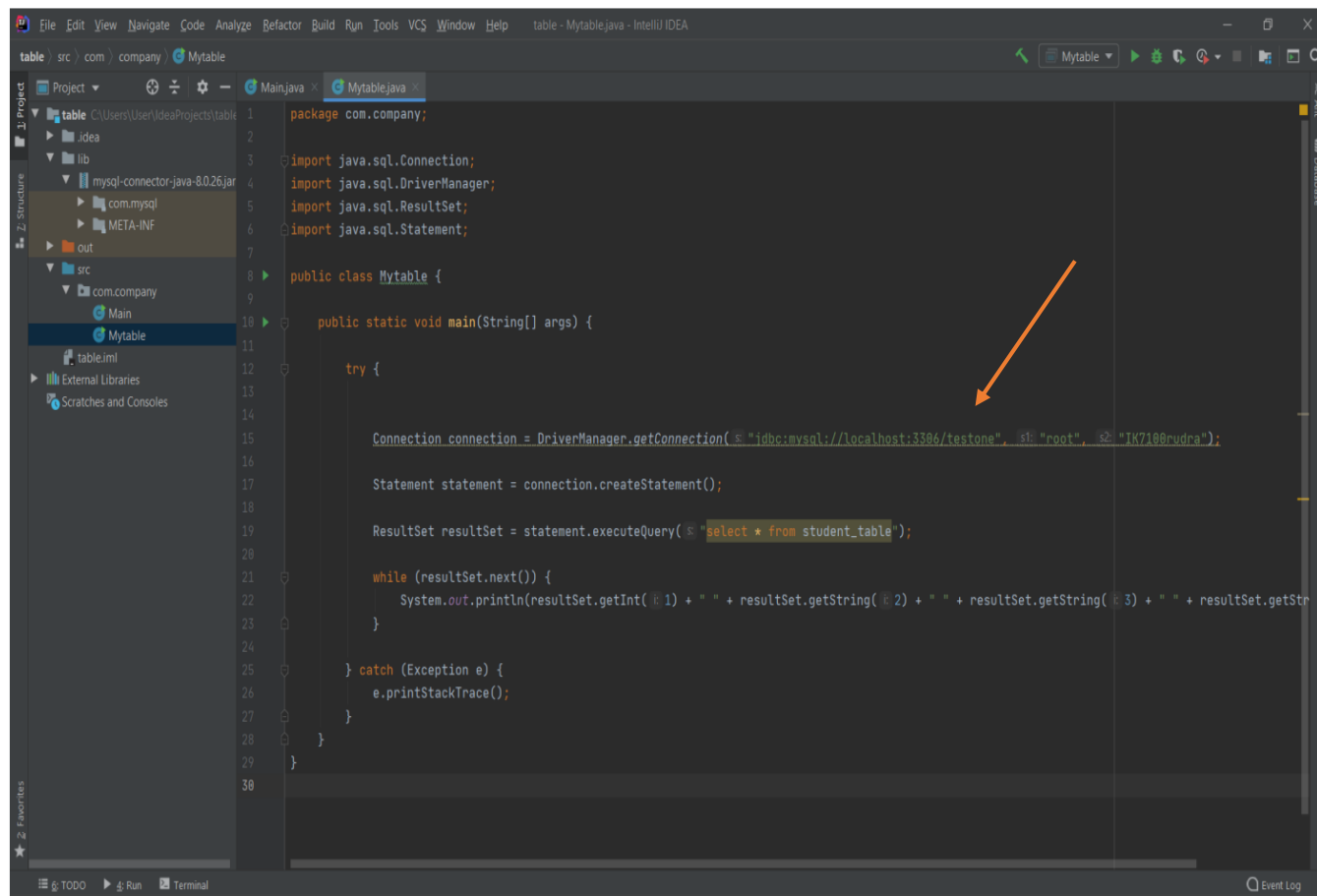
Database connection

A database connection is a facility in computer science that allows client software to talk to database server software, whether on the same machine or not. A connection is required to send commands and receive answers, usually in the form of a result set. Connections are a key concept in data-centric programming. Since some DBMS engines require considerable time to connect, connection pooling was invented to improve performance. No command can be performed against a database without an "open and available" connection to it.

Connections are built by supplying an underlying driver or provider with a connection string, which is a way of addressing a specific database or server and instance as well as user authentication credentials (for example, Server=sql_box;Database=Common;User ID=uid;Pwd=password;). Once a connection has been built it can be opened and closed at will, and properties (such as the command time-out length, or transaction, if one exists) can be set. The Connection String is

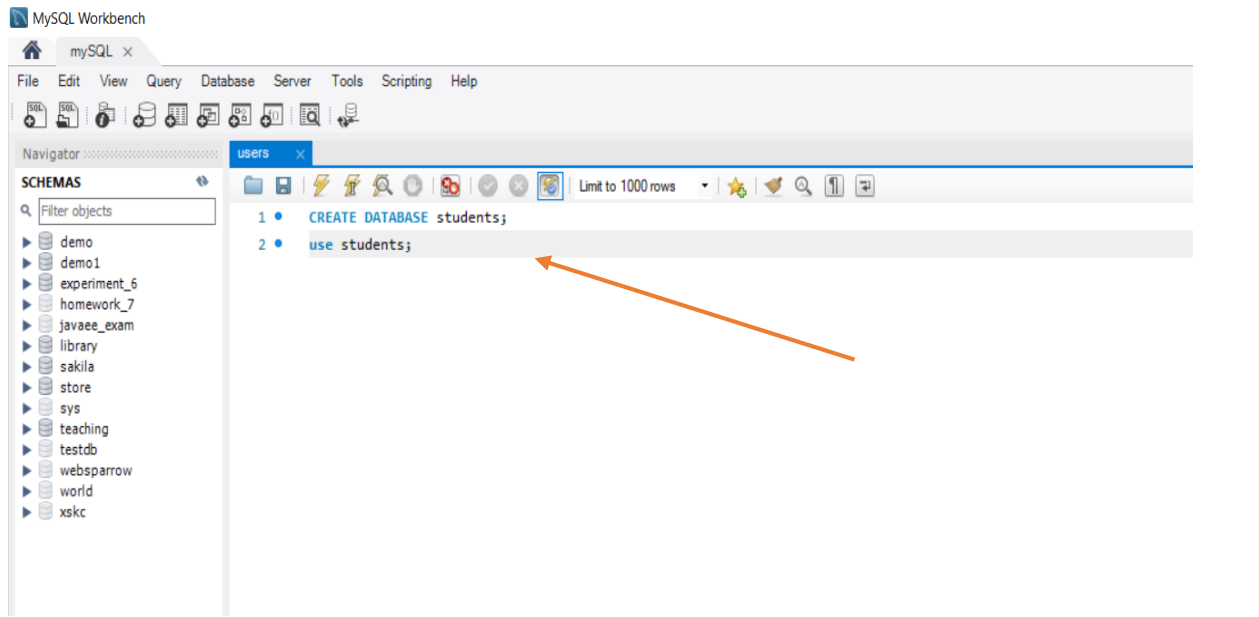
composed of a set of key/value pairs as dictated by the data access interface and data provider being used.

Many databases (such as PostgreSQL) only allow one operation to be performed at a time on each connection. If a request for data (a SQL Select statement) is sent to the database and a result set is returned, the connection is open but not available for other operations until the client finishes consuming the result set. Other databases, like SQL Server 2005 (and later), do not impose this limitation. However, databases that provide multiple operations per connection usually incur far more overhead than those that permit only a single operation task at a time



```
1 package com.company;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.Statement;
7
8 public class Mytable {
9
10     public static void main(String[] args) {
11
12         try {
13
14             Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/testone", "root", "TK7100rudra");
15
16             Statement statement = connection.createStatement();
17
18             ResultSet resultSet = statement.executeQuery("select * from student_table");
19
20             while (resultSet.next()) {
21                 System.out.println(resultSet.getInt(1) + " " + resultSet.getString(2) + " " + resultSet.getString(3) + " " + resultSet.getString(4));
22             }
23         } catch (Exception e) {
24             e.printStackTrace();
25         }
26     }
27 }
28
29 }
```

System menu, User login, Data entry, modification and deletion, Database query



MySQL Workbench

mySQL x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

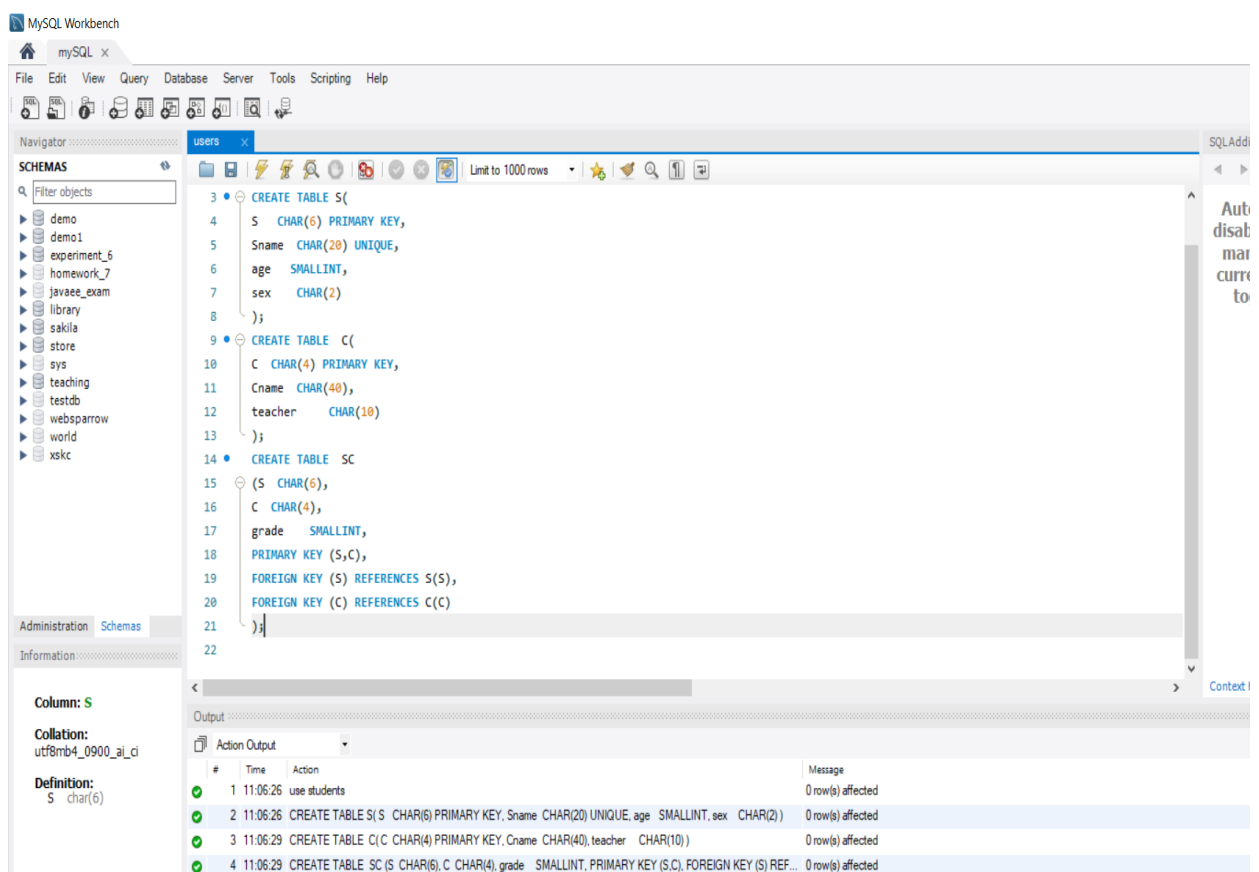
Filter objects

- demo
- demo1
- experiment_6
- homework_7
- javaee_exam
- library
- sakila
- store
- sys
- teaching
- testdb
- websparrow
- world
- xskc

users x

Limit to 1000 rows

```
1 • CREATE DATABASE students;
2 • use students;
```



MySQL Workbench

mySQL x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- demo
- demo1
- experiment_6
- homework_7
- javaee_exam
- library
- sakila
- store
- sys
- teaching
- testdb
- websparrow
- world
- xskc

users x

Limit to 1000 rows

```
3 • CREATE TABLE S(
4   S CHAR(6) PRIMARY KEY,
5   Sname CHAR(20) UNIQUE,
6   age SMALLINT,
7   sex CHAR(2)
8 );
9 • CREATE TABLE C(
10  C CHAR(4) PRIMARY KEY,
11  Cname CHAR(40),
12  teacher CHAR(10)
13 );
14 • CREATE TABLE SC
15 (S CHAR(6),
16  C CHAR(4),
17  grade SMALLINT,
18  PRIMARY KEY (S,C),
19  FOREIGN KEY (S) REFERENCES S(S),
20  FOREIGN KEY (C) REFERENCES C(C)
21 );
22
```

Administration Schemas

Information

Column: S

Collation: utf8mb4_0900_ai_ci

Definition: S char(6)

Output

Action Output

#	Time	Action	Message
1	11:06:26	use students	0 row(s) affected
2	11:06:26	CREATE TABLE S (S CHAR(6) PRIMARY KEY, Sname CHAR(20) UNIQUE, age SMALLINT, sex CHAR(2))	0 row(s) affected
3	11:06:29	CREATE TABLE C (C CHAR(4) PRIMARY KEY, Cname CHAR(40), teacher CHAR(10))	0 row(s) affected
4	11:06:29	CREATE TABLE SC (S CHAR(6), C CHAR(4), grade SMALLINT, PRIMARY KEY (S,C), FOREIGN KEY (S) REFERENCES S(S), FOREIGN KEY (C) REFERENCES C(C))	0 row(s) affected

MySQL Workbench

mySQL x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- demo
- demo1
- experiment_6
- homework_7
- javaee_exam
- library
- sakila
- store
- students
 - Tables
 - c
 - s
 - sc
 - Views
 - Stored Procedures
 - Functions
- sys
- teaching
- testdb
- webparrow
- world
- xskc

Administration Schemas

Information

Table: c

Columns:

- C char(4) PK
- Cname char(40)
- teacher char(10)

users

Limit to 1000 rows

```
21 • insert into SC values('s1','c5',70);
22 • insert into SC values('s2','c1',85);
23 • insert into SC values('s2','c2',null);
24 • insert into SC values('s2','c4',null);
25 • insert into SC values('s3','c1',90);
26 • insert into SC values('s3','c2',85);
27 • insert into SC values('s3','c3',95);
28 • insert into SC values('s4','c1',75);
29 • insert into SC values('s4','c3',null);
30 • insert into SC values('s4','c4',70);
31 • insert into SC values('s5','c1',70);
32 • insert into SC values('s5','c2',60);
33 • insert into SC values('s5','c3',80);
34 • insert into SC values('s5','c5',65);
35 • insert into SC values('s6','c1',90);
36 • insert into SC values('s6','c2',null);
37 • insert into SC values('s6','c3',90);
38 • insert into SC values('s6','c5',null);
39
40
```

Output

Action Output

#	Time	Action
29	11:07:56	insert into SC values('s5','c3',80)
30	11:07:56	insert into SC values('s5','c5',65)
31	11:07:56	insert into SC values('s6','c1',90)
32	11:07:56	insert into SC values('s6','c2',null)
33	11:07:56	insert into SC values('s6','c3',90)
34	11:07:56	insert into SC values('s6','c5',null)

MySQL Workbench

mySQL x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- demo
- demo1
- experiment_6
- homework_7
- javaee_exam
- library
- sakila
- store
- students
 - Tables
 - c
 - s
 - sc
 - Views
 - schoolstudent
 - Stored Procedures
 - Functions
- sys
- teaching
- testdb
- webparrow
- world

Administration Schemas

Information

No object selected

users

Limit to 1000 rows

```
1 • use students;
2
3 • create view schoolstudent
4 as
5 select S,SNAME,AGE
6 from S
7 where sex='M';
8
9
10
```

Output

Action Output

#	Time	Action	Message
30	11:07:56	insert into SC values('s5','c5',65)	1 row(s) affected

Data Query -

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'students' expanded, showing 'schoolstudent' under 'Views'. The main editor contains the following SQL code:

```
2
3 create view schoolstudent
4 as
5 select S,SNAME,AGE
6 from S
7 where sex='M';
8
9 select S,SNAME from schoolstudent where age>19;
```

Two orange arrows point to the SQL code: one to the view definition and another to the query. The 'Result Grid' shows the following data:

S	SNAME
S1	WANG
S3	CHEN

The bottom panel shows the 'Action Output' table:

#	Time	Action	Message	Duration / Fetch
31	11:07:56	insert into SC values('S6','c1',90)	1 row(s) affected	0.156 sec
32	11:07:56	insert into SC values('S6','c2',null)	1 row(s) affected	0.125 sec

On the right, a message states: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.'

Modification -

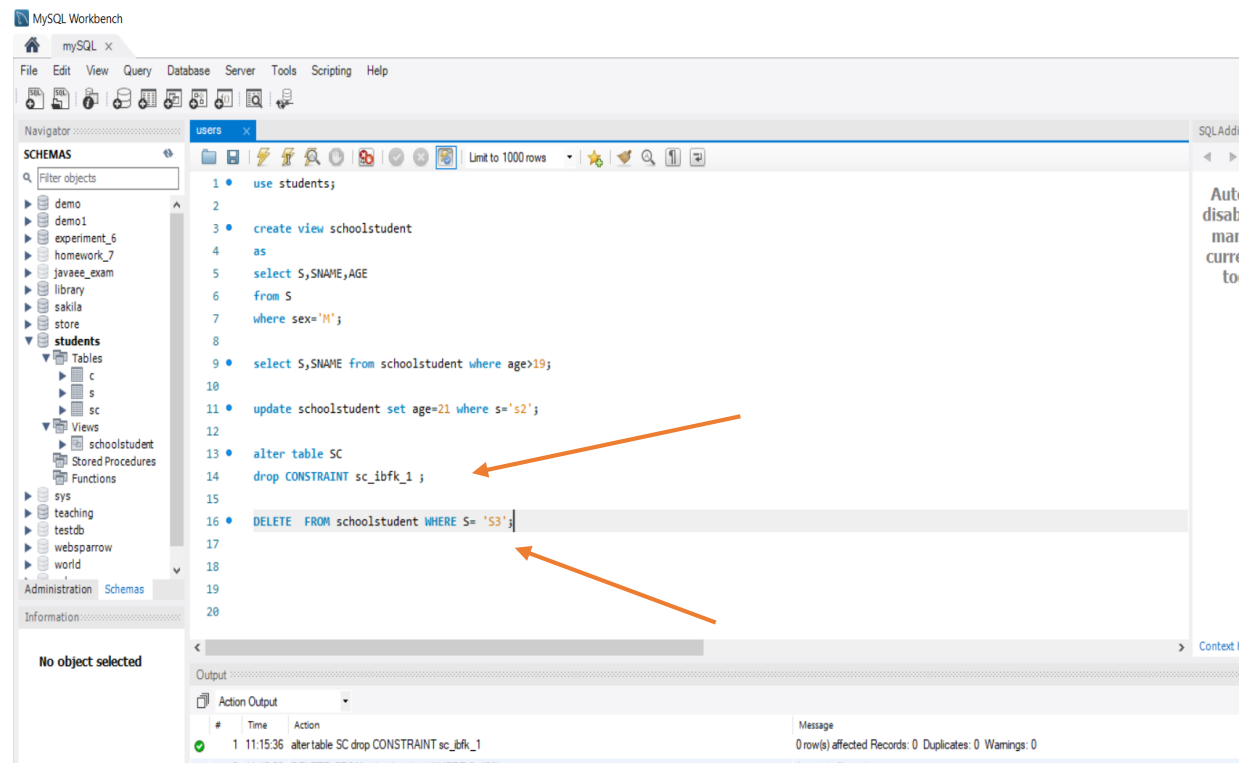
The screenshot shows the MySQL Workbench interface with the following SQL code in the main editor:

```
1 use students;
2
3 create view schoolstudent
4 as
5 select S,SNAME,AGE
6 from S
7 where sex='M';
8
9 select S,SNAME from schoolstudent where age>19;
10
11 update schoolstudent set age=21 where s='s2';
```

An orange arrow points to the update statement. The bottom panel shows the 'Action Output' table:

#	Time	Action	Message
32	11:07:56	insert into SC values('S6','c2',null)	1 row(s) affected

Deletion –



Modifying existing records is done using the UPDATE statement. To do this we tell the database which table we want to update, what we want to change the values to for any or all of the fields, and under what conditions we should update the values. Be careful to not forget the WHERE clause or the update statement will modify *all* of the records in the Site table.

Deleting records can be a bit trickier, because we have to ensure that the database remains internally consistent. If all we care about is a single table, we can use the DELETE command with a WHERE clause that matches the records we want to discard.

User can see all the sections of all classes with their individual section codes. If the user wants to add a new section for any identical class and shifts it needs just a section name with its section codes. After saving it will appear into the right side of user interface. User also can delete or reset any section data if needed.

Summary of course design

We are analysis User Management, Student Profile Management, Routine Management, Result Management, Employee Management and Accounts Management etc. We are complete software that has included this entire requirement.

All good software may have some limitations this software may be some limitations. We are trying to fix it as per possible.

The application Education Management Software is the total package for the school or college management system requirements. With the help of menu bar users can interact with software very easily almost every object has been provided with tool bar.

This application is also supports terminal services so that database will be more secure by centralized the database. In fine we want to say we are not yet an experienced system analyst, but tried our best to fulfill this project.

Problems and solutions in curriculum design

The most common problems/barrier in curriculum design and their solution is given bellow –

Change

Change occurs in every facet of life. However, change can greatly impact curriculum design. When a superintendent resigns it can change the course of curriculum work. The new superintendent may select a different framework or abandon the existing curriculum documents. Standards have become a focus across the nation. When policymakers change standards, curriculum designers must align the curriculum with new standards and this often requires starting over. As school districts focus on new skills to emphasize or magnet programs highlight STEM or leadership themes, school staff will need to revise the curriculum to meet the program goals.

Barrier-Breaking:

School leaders are wise when they design an archive or a website that houses the district's curriculum documents. Curriculum design requires constant change and moving parts. If a new principal is hired in the middle of the school year, he or she should be able to identify the district's curriculum in a timely manner. While we cannot stop the forces of change, we can focus on organization, communication, timelines for curriculum revision, and storing curriculum units. Consistency in how we support teaching and learning allows all educators to have a clear understanding of the written curriculum. If the framework or template changes with each standards revision, then it will be an additional ingredient to throw into the change

stew.

Communication Tools

A major barrier to curriculum design and implementation is a lack of communication. A school district may identify twelve – twenty teacher leaders to serve on a district curriculum committee. The committee may meet for twelve months or up to two years. At the end of the period, a curriculum document is distributed to all of the teachers in the school district. While months of conversations and deliberation went into the document, the experts are the committee members. Districts need to stop handing documents (hard copy with shrink wrap or electronic) to teachers and expecting the document to transform teaching and learning. The teachers who implement the curriculum need to be able to ask questions, post suggestions, and explain the strengths and weaknesses of the curriculum. Curriculum deliberation should not be finished when the committee adopts the final edits.

Barrier-Breaking:

Online tools make it easy for curriculum leaders to remove this common barrier. Google Docs, Google Forms, Live Binders, Padlet, Today's Meet, Google Hangout, Web sites, and Voxer are among the tools that educators can use to communicate. Online programs are also available for purchase. The written curriculum is critically important. However, educators need to have an opportunity to discuss the implemented curriculum, even if they were not part of the original design team. Wiggins and McTighe (2007) wrote, "Schooling at its best reflects a purposeful arrangement of parts and details, organized with deliberate intention, for achieving the kinds of learning we seek" (p. 9). If the curriculum design is not reaching its desired intentions, classroom teachers need a mechanism to share observations and make recommendations. Continuous improvement is not possible if teachers are only able to communicate at face-to-face meetings or during the curriculum writing process.

Hidden Curriculum

The 'hidden curriculum' is the unintended curriculum. "It defines what students learn from the physical environment, the policies, and the procedures of the school" (Glatthorn & Jailall, 2009, p. 110). Most teachers and administrators don't recognize the hidden curriculum, because it is 'the way we do business.' There are messages in every school that interfere with the written curriculum. The way that students are treated when they enter school is part of the

hidden curriculum. The rules or lack of rules throughout the school sends a message to students. The ability of parents to voice their concerns and recommendations is part of the hidden curriculum.

In a school that administers common formative assessments on a regular basis, a student may ask the teacher, “Why do we have tests every three weeks?” If the teacher replies, “We want all of our students to be prepared for the End-of-Grade tests in the spring,” then the ‘hidden’ message will imply that the purpose for school is to prepare for a single test. The message educators want to send is that learning is an ongoing process and their intention is to develop lifelong learners. However, the answer that the teacher provided indicates that common formative assessments are intended to prepare students for a high-stakes test, rather than a method to assess students’ understanding of the current unit of study.

Barrier-Breaking:

Trying to identify the hidden curriculum is like changing the culture in a nation that is 200 years old. It is difficult to identify the things we cannot see. The following questions will help curriculum design teams identify the hidden curriculum. The answers to these questions will support the ongoing work of curriculum designers.

Analysis of existing problems

Some of the most crucial current issues in educational management are related to governance, effective schooling and performance indicators, conditions of employment of teachers, and in-service education. For example, what different types of policy and operational decisions should be made by central, regional, and school bodies? What are the most important indicators of school performance that can be reliably assessed? How can we assure that teacher-evaluation

practices are conducted fairly and appropriately so that the interests of both teachers and students are served? How can schools effectively integrate their teaching and continual professional development activities so that minimal disruption occurs with student learning? These issues were identified during study leave in 1988-89. Interviews were conducted with staff in universities, colleges, schools, government departments, research institutes, and local education authorities in several countries. The crucial nature of the issues was assessed by frequent attention to the intensity with which interviewees spoke about an issue, and the author's subjective assessment.

Experience of curriculum design

Curriculum design is a rewarding process which enables teachers to have professional conversations about the intended and taught the curriculum. In addition to horizontal alignment, curriculum design provides educators with the opportunity to discuss what is essential at other grade levels and courses. My experiences with curriculum design and observing the work of teacher teams in other school systems have led me to identify the following barriers. With proper planning, budget allocations, scheduling, communication, and support from district administrators, each of these barriers can be removed.

Learning experience design (LX design) is the process of creating learning experiences that enable the learner to achieve the desired learning outcome in a human centered and goal oriented way.

Learning experience design is rooted in a combination of several design disciplines with the field of learning. Key design principles used in LXD come from interaction design, user experience design, experience design, graphic design and game design. These design principles are combined with elements of education, training and development, instructional design, cognitive psychology, experiential learning, educational sciences and neuroscience.

Everything we learn comes from experience, that's a fact. As mentioned earlier an experience is any situation you encounter that takes an amount of time and leaves an impression. These experiences don't necessarily have to take place in an educational setting like a school. They can take place at home, outside, in the office or anywhere else.

Not every experience is as educational as the next. Some experiences can be straight out boring or annoying. Fortunately, we've all had experiences that were very educational and that will last a lifetime. Being able to design such powerful experiences is the main quality of a good LX designer.