Software Requirement Specification(SRS) for

College Management System

# Introduction

### 1.1 Purpose:

The main objective of the College Management System is to manage the details of College, Faculty, Course, etc . The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the work such as registration,

Add student, attendance,view performance of the student etc. It tracks all the details related to the college.

### 1.2 Scope:

### As Colleges are growing day by day more and more, and also increasing the complexity of storing information of students and related to the college system

### The College management system is an automated version of manual College Management System. It can handle all details about a student.

### This project is based on the educational institute system where this application gives maximum services in a single software product that is used by teacher and system administration. This project is based on an application that is

### sharing information on different departments in a college.

### 1.3 Definitions:

CMS – College Management System

SRS- Software Requirement Specification

GUI- Graphical User Interface

**1.5 Overview:**

The objective of Student Portal is to provide an online web-based solution for academic use. In this application all data will be available such as student details, faculty details, course details, exam details, report card etc. It is Easy to use. In future it helps our students and staff to organize and access information about all aspects. This website provides a way for student and faculty to communicate to each other, this will solve all drawbacks of manual process, increase the efficiency and speed up all works to be completed. At college managements side a person can view the most important student and staff data. Every student and staff will provide with unique login id and password. All the data will be at least once the validated from the college database. Hence this process also helps in maintaining consistency and integrity. The student and staff acquaintance system that helps to users can readily store the students and staff information through online. The user can readily store and retrieve the data through online. This system helps the user to generate the dynamic legwork. Thus the user can convenient to interact with the system to readily and accessibility the resources through internet or intranet.

**EXISTING SYSTEM**

Previously the college management primary focus was on managing students and account sections and rest are going besides them. But this is not the actual scenario of today’s environment. Apart from these, there are other modules which exist in a particular college, which the management will have to look after. Existing system was not paying attention for managing their working employees, integration of library management was not possible, thus another software has to be used which requires extra investment and effort. If the college providing boarding facility, the requirement for hostel management and store management was the requirement which was not available in the existing system.

**NEED FOR NEW SYSTEM**

* The new system is totally computerized system.
* A new system provides features like time efficiency to show student, faculty, exam, attendance details.

**2.Overall Description**:

Portal is a web system that provides the function and features to authenticate and identify the users and provide then with easy, intuitive, personalized and user-customizable web-interface for facilitating access to information and services that are of primary relevance and interests to the users. Student Portal is nothing but a portal which thinks students as the main target users & provides so many useful services to students at a single place & through a single interface but in customized form.

**2.1 Product Perspective**:

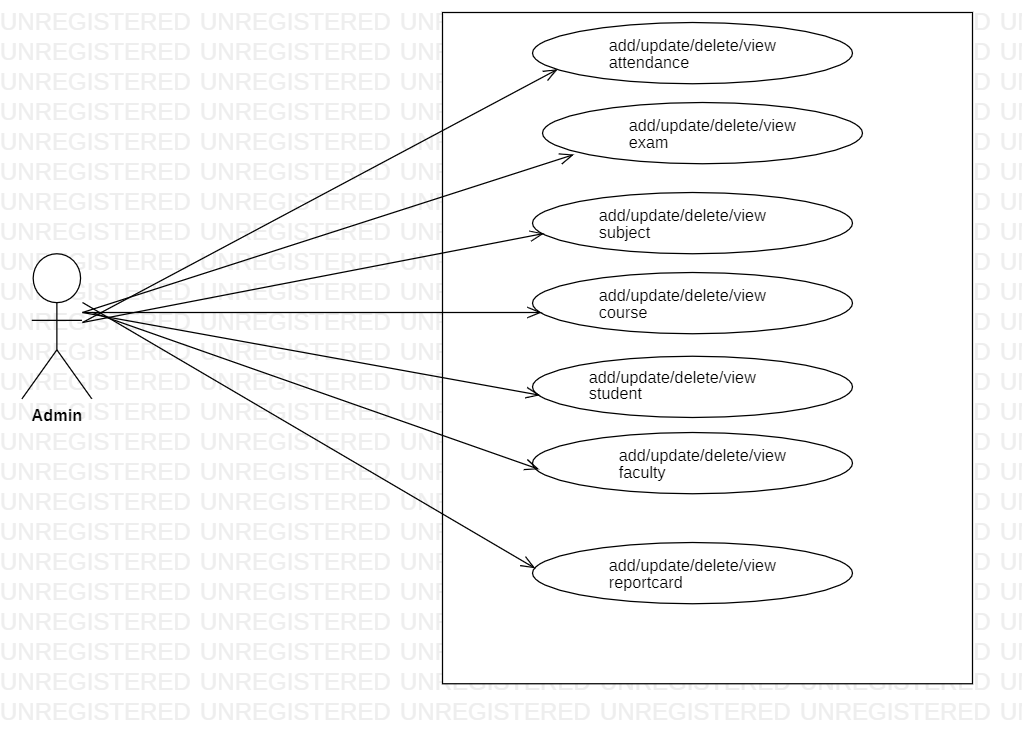
This product aimed toward a person who don’t want to visit the center as he might don’t get time for that or might not interested in visiting there and dealing with lot of formalities.

**2.2 Product Functions**:

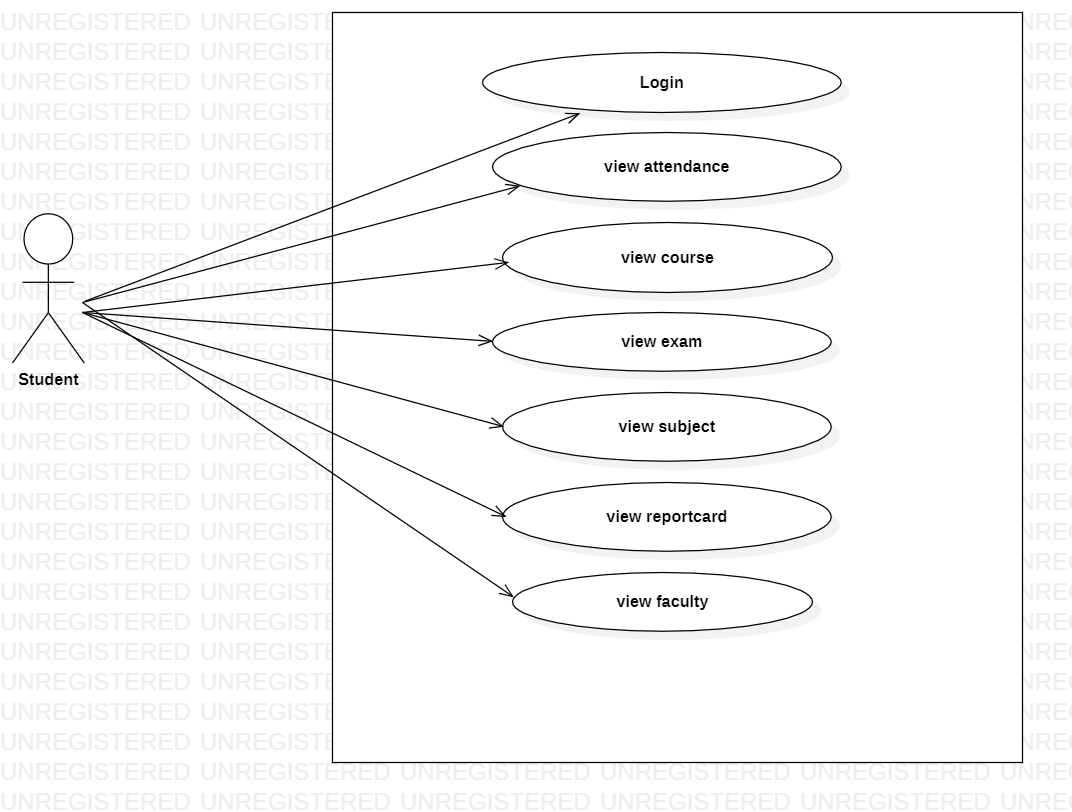
College Management System should support this use case:

**Use Case Diagrams** : A Use case is a description of set of sequence of actions. Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor - Sender, Secondary- Actor Receiver.

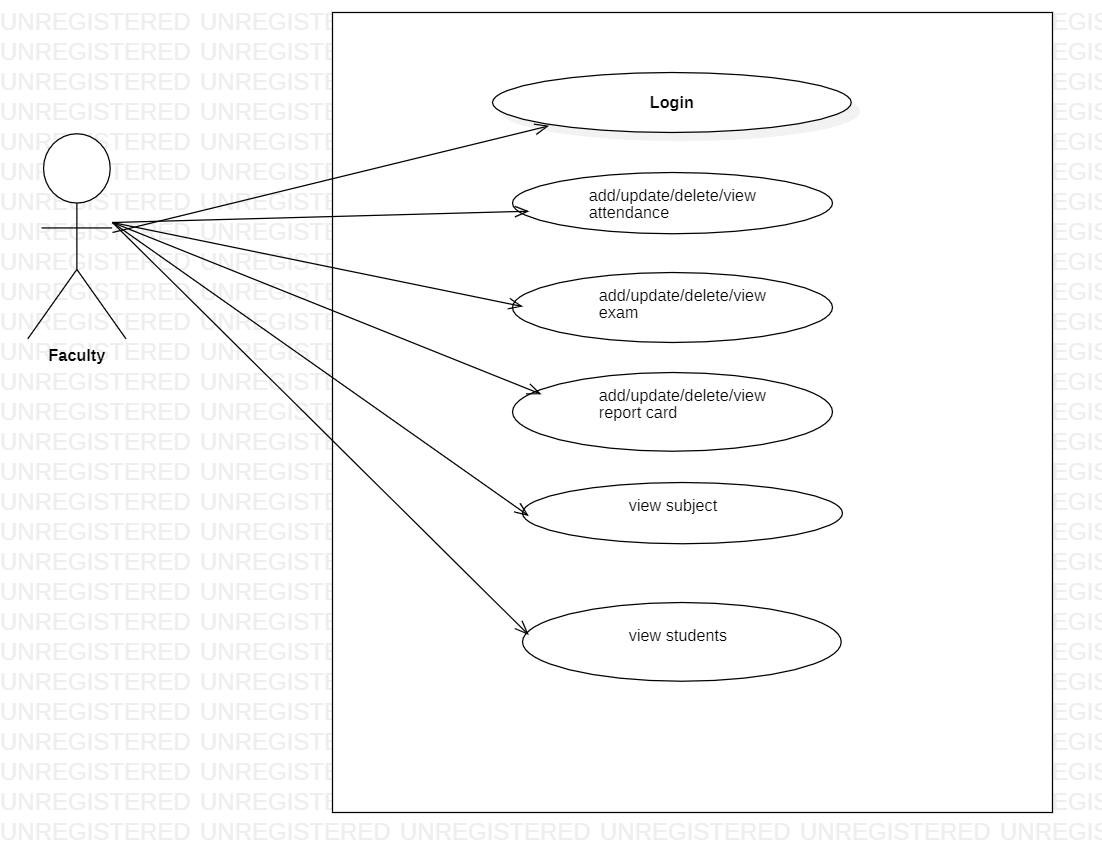
**Use case diagram for admin**



**Use Case diagram for Student**



**Use Case diagram for Faculty**



**2.3 User Characeristics**:

User should be familiar with the terms like login, register etc.

**2.4 Principle Actors**:

3 Principle Actors are Admin Faculty and Student.

**2.5 General Constraints**:

A full internet connection is required for CMS.

**2.6 Assumptions and Dependencies**:

Working of CMS need Internet Connection.

**3. Specific Requirements**:

3.1 **FUNCTIONAL SPECIFICATION**

User Specification

**Admin:**

Admin can add/view/update/delete a student, faculty, attendance, subjects and courses.

**User:**

User can view information of available student, faculty, attendance, subjects and courses.

**MODULE SPECIFICATION**

**User**

**Login:**

The login page only used by the authority or admin.

**View Student:**

Users can only see the students details.

**Attendance:**

This module displays the attendance record of the student and also displays the defaulter’s list. Faculty user can update the attendance of the student.

**Exam**:

Users can view the exam date and time.

**Courses:**

Users can view the courses.

**Subject:**

Users can view the subjects.

**Admin**

**Dashboard:**

In this section admin can view the overview of the CMS (Like total Students, total Faculty, report card, attendance of students,etc)

**Students:**

Admin can create/edit/delete students,faculty,attendance of students.

**Manage Subscribers:**

Admin can manage subscribers.

**3.2 Non-Functional Requirements**:

Following Non-Functional Requirements will be there in the

insurance to the internet:

(i) Secure access to CMS’s confidential data.

(ii) 24X7 availability.

(iii) Better component design to get better performance at peak

time.

(iv) Flexible service based architecture will be highly desirable for

future extension. Non-Functional Requirements define system

properties and constraints.

Various other Non-Functional Requirements are:

 Security

 Reliability

 Maintainability

 Portability

 Extensibility

 Reusability

 Compatibility

 Resource Utilization

**3.3 Performance Requirements**:

In order to maintain an acceptable speed at maximum number of uploads allowed from a particular customer as any number of users can access to the system at any time. Also the connections to the servers will be based on the attributes of the user like his location and server will be working 24X7 times.

**3.4 Technical Issues**:

This system will work on client-server architecture. It will require an internet server and which will be able to run application. The system should support some commonly used browser such as IE ,mozzila firefox, chrome etc.

**Operating Environment:**

Server Side:

**Processor :** Intel® Xeon® processor 3500 series

**HDD :** Minimum 500GB Disk Space

**RAM :** Minimum 4GB

**OS :** Windows 10, Linux 6

**Database :** MySQL

Client Side (minimum requirement):

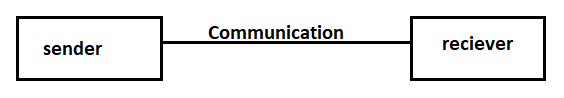
**Processor :** Intel Dual Core

**HDD :** Minimum 80GB Disk Space

**RAM :** Minimum 2GB

**OS :** Windows 7, Linux

**Communication Interfaces**:

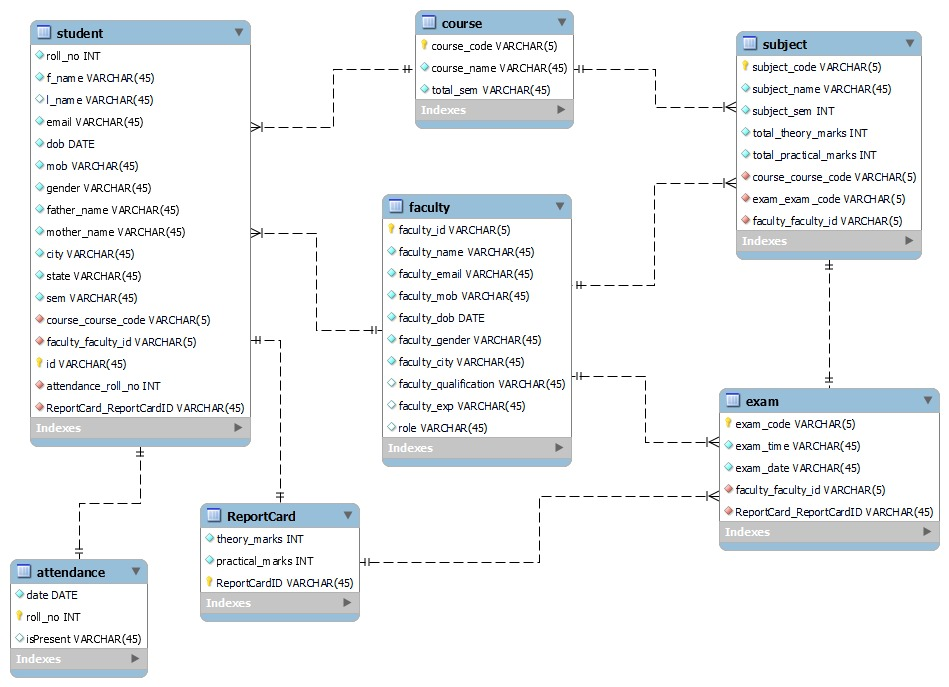
The two parties should be connected by LAN or WAN for the communication purpose.

**ER DIAGRAM**

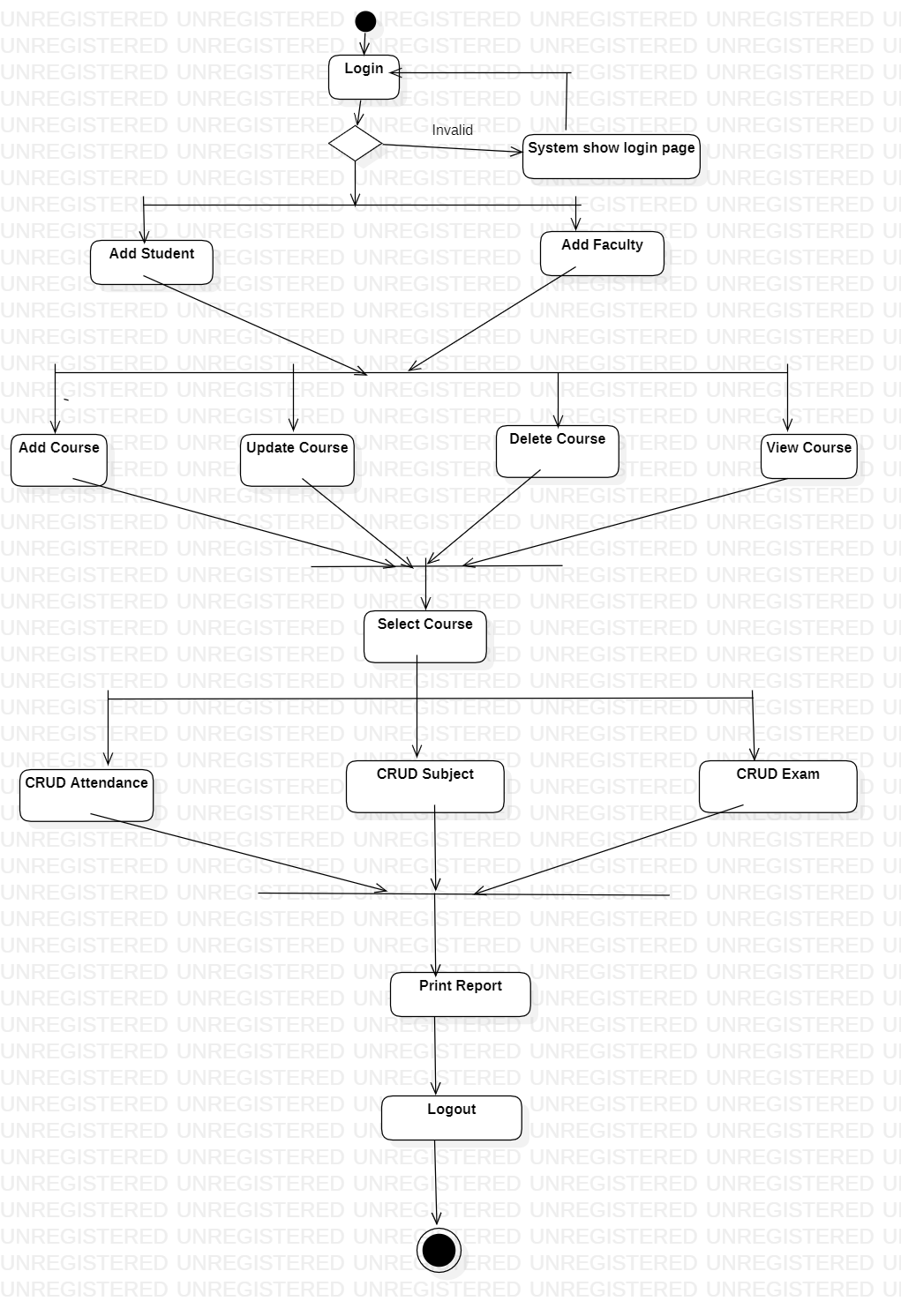
The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

* It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
* It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
* In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

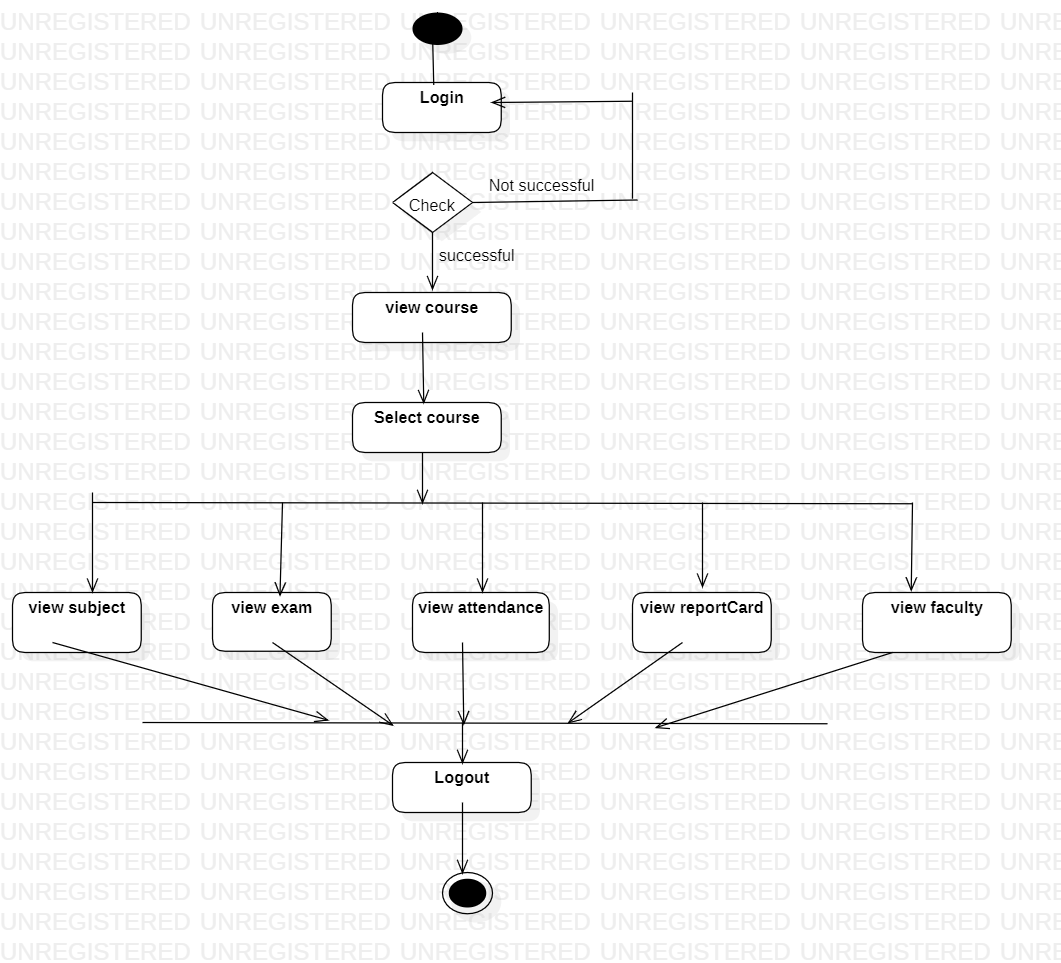
.

****

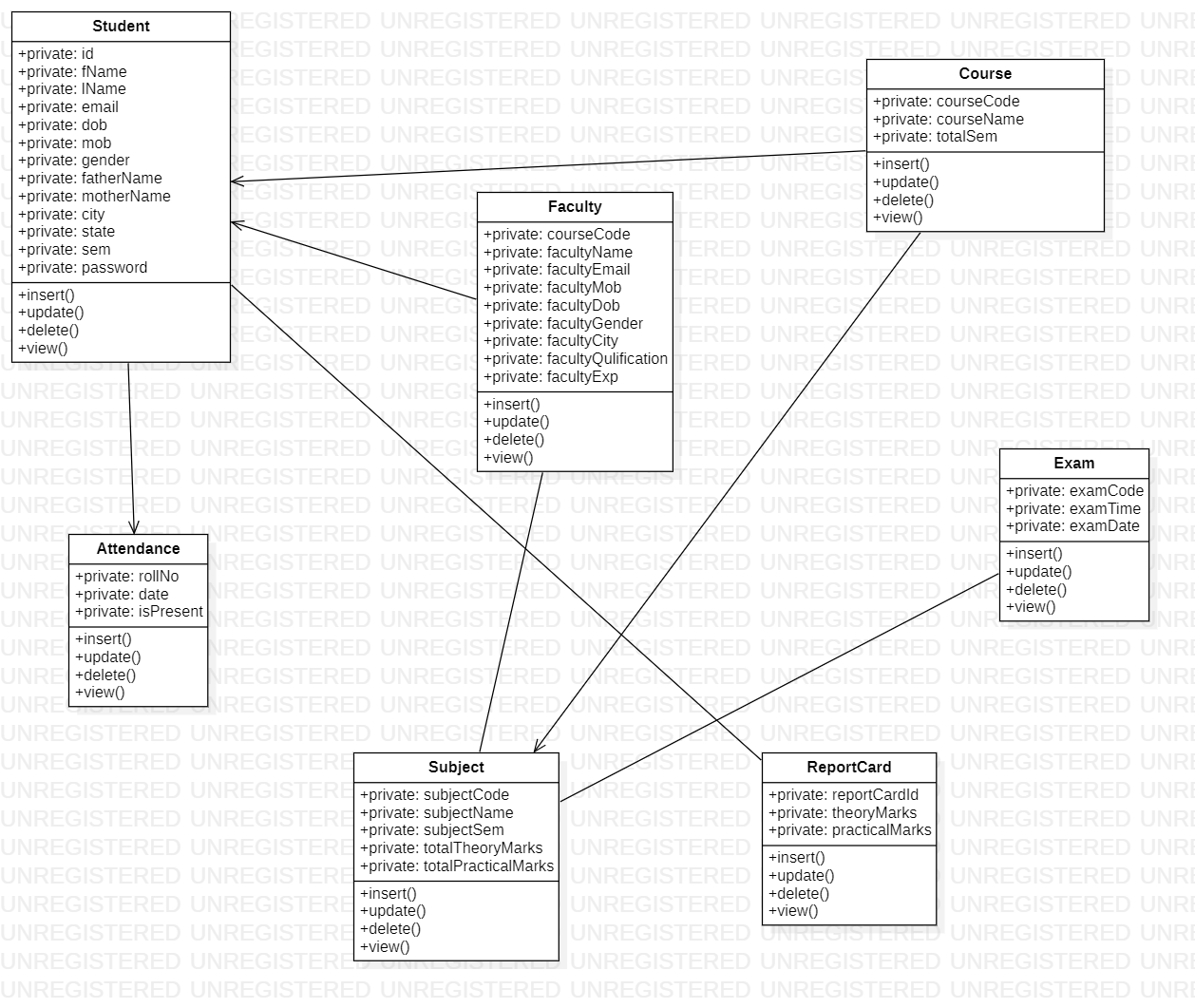
**Activity Diagram for admin**



**Activity Diagram for user**



**Class Diagram for admin**



**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 MS Access database has been chosen for developing the relevant databases.

* Student:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| Id | int | NO | PRI | NULL | auto\_increment |
| city | varchar(255) | YES |  | NULL |  |
| dob | date | YES |  | NULL |  |
| email | varchar(255) | YES | UNI | NULL |  |
| f\_name | varchar(255) | YES |  | NULL |  |
| father\_name | varchar(255) | YES |  | NULL |  |
| l\_name | varchar(255) | YES |  | NULL |  |
| mob | varchar(255) | YES |  | NULL |  |
| mother\_name | varchar(255) | YES |  | NULL |  |
| password | varchar(255) | YES |  | NULL |  |
| sem | varchar(255) | YES |  | NULL |  |
| state | varchar(255) | YES |  | NULL |  |
| course\_code\_course\_code | int | YES | MUL | NULL |  |
| faculty\_id\_faculty\_id | int | YES | MUL | NULL |  |
| report\_card\_id\_report\_card\_id | int | YES | MUL | NULL |  |

* Faculty:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| faculty\_id | int | NO | PRI | NULL | auto\_increment |
| faculty\_city | varchar(255) | YES |  | NULL |  |
| faculty\_dob | datetime(6) | YES |  | NULL |  |
| faculty\_email | varchar(255) | NO | UNI | NULL |  |
| faculty\_exp | varchar(255) | YES |  | NULL |  |
| faculty\_gender | varchar(255) | YES |  | NULL |  |
| faculty\_mob | varchar(255) | YES |  | NULL |  |
| faculty\_name | varchar(255) | YES |  | NULL |  |
| faculty\_qulification | varchar(255) | YES |  | NULL |  |
| role | varchar(255) | YES |  | NULL |  |

* Course

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| course\_code | int | NO | PRI | NULL | auto\_increment |
| course\_name | varchar(255) | YES |  | NULL |  |
| total\_sem | int | NO |  | NULL |  |

* Subject

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| subject\_code | int | NO | PRI | NULL | auto\_increment |
| subject\_name | varchar(255) | YES |  | NULL |  |
| subject\_sem | varchar(255) | YES |  | NULL |  |
| total\_practical\_marks | varchar(255) | YES |  | NULL |  |
| total\_theory\_marks | varchar(255) | YES |  | NULL |  |
| course\_code\_course\_code | int | YES | MUL | NULL |  |
| exam\_code\_exam\_code | int | YES | MUL | NULL |  |
| faculty\_id\_faculty\_id | int | YES | MUL | NULL |  |

* Attendance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| roll\_no | int | NO | PRI | NULL | auto\_increment |
| date | varchar(255) | YES |  | NULL |  |
| is\_present | varchar(255) | YES |  | NULL |  |
| id\_id | int | YES | MUL | NULL |  |

* Exam

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| exam\_code | int | NO | PRI | NULL | auto\_increment |
| exam\_date | varchar(255) | YES |  | NULL |  |
| exam\_time | varchar(255) | YES |  | NULL |  |
| faculty\_id\_faculty\_id | int | YES | MUL | NULL |  |
| report\_card\_id\_report\_card\_id | int | YES | MUL | NULL |  |
| subject\_code\_subject\_code | int | YES | MUL | NULL |  |

* Report Card

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| report\_card\_id | int | NO | PRI | NULL | auto\_increment |
| practical\_marks | varchar(255) | YES |  | NULL |  |
| theory\_marks | varchar(255) | YES |  | NULL |  |
| id\_id | int | YES | MUL | NULL |  |