**Student Management System**

Version 1

Table of Contents

[1. Introduction 3](#_Toc57021959)

[1.1 **Purpose**: 3](#_Toc57021960)

[**1.2.** **Scope**: 3](#_Toc57021961)

[**1.3.** **Need for the proposed system:** 3](#_Toc57021962)

[**1.4. Features and Benefits:** 3](#_Toc57021963)

[2. Overall Description 3](#_Toc57021964)

[**2.1: Product Perspective:** 3](#_Toc57021965)

[**2.2: Product Function:** 4](#_Toc57021966)

[**2.3: Constraints:** 4](#_Toc57021967)

[**2.4: Assumptions:** 5](#_Toc57021968)

[3. Technical Requirements 5](#_Toc57021969)

[**3.1: User class and characteristics:** 5](#_Toc57021970)

[**3.2: Project Scope** 5](#_Toc57021971)

[4. Functional and Non-Functional Requirements 5](#_Toc57021972)

[**4.1 Functional Requirements**: 5](#_Toc57021973)

[**4.2 Non-functional Requirement:** 6](#_Toc57021974)

[5. Business Requirements 6](#_Toc57021975)

[6. User Scenarios and Persona 7](#_Toc57021976)

[7. Context Diagram: 7](#_Toc57021977)

[8. Use case Modelling 8](#_Toc57021978)

[**7.1 Use cases** 9](#_Toc57021979)

[9. Domain Modelling 11](#_Toc57021980)

[10. ERD 12](#_Toc57021981)

[11. System Architecture 13](#_Toc57021982)

[10.1 Architectural Design 13](#_Toc57021983)

[10.2 Three-Tier Client Server Architecture 15](#_Toc57021984)

[11. Conceptual and Concrete UI Design 16](#_Toc57021985)

[12. Prototype UI design for both, front and back end system. 19](#_Toc57021986)

[13. Test Cases: 22](#_Toc57021987)

# Introduction

**Student Management System offers hundreds of courses (from certificate to degree level), to over 500,000 students at various locations. These coursed are offered in three modes: full time, part time and online. It employs staff including teachers who are required to teach at various locations.**

## 1.1 **Purpose**:

Our product is student management system gives all the services that must be provided to a student over the internet to find fee details provided by that administrator of the college.  
This product contains each and every data regarding student, payment etc., personal details which can be updated by the student and viewed by the administrator. It provides the detailed information about the fee details and the location (place) of college.

Its purpose is to automate and centralize the whole system. The main goal of the system is to automate the process carried out in the organization with improved performance.

## **1.2.** **Scope**:

**The students can access this online application using their user id and password. The students are allocated a student id and an email address at the time of enrolment.**

Moreover, storing and retrieving of the information is easy, so work can be done speedily and in time.

## **1.3.** **Need for the proposed system:**

TAFE management system will use the centralized database of the whole system of the department.

## **1.4. Features and Benefits:**

* Reduces the manual workload.
* Complete details of the student can be stored and retrieved.
* Admin can see all the student’s payment details.
* The student can view all his details, payment details and the location of the college.

# Overall Description

## **2.1: Product Perspective:**

* + 1. **User interface:** The application that will be developing will have a user friendly and menu-based interface.

**Following screens will be provided:**

* A login screen for entering the username and passwords, so that the authorized user can have an access without any problems.
* There will be a screen which will be displaying the major tasks that the system will be performing e.g. add details, delete, view details of the students.
* All the major tasks mentioned above will have their separate forms and will perform the desired actions.
  + 1. **Hardware Interface:**
    - Intel Pentium 4 or higher processor
    - 1.5Ghz
    - 512MB of RAM or More
    1. **Software Interface:**
* Operating system: Window XP, Vista,7,8,8.1 and higher
* Platform: .NET
* Database: SQL server
* Language: Visual Studio 2019 (Windows & C#)

1. **Communication Interface:**

The communications functions required by this product are LAN connection within the whole company so that the Admin, employee, and customer can interact with each other.

1. **Memory Constraints:**

At least 512MB RAM and 4GB of the Hard disk space will be required for running the application.

1. **Operations:**

* The system will have the user-friendly interfaces. The system will maintain the records of the students and staff.
* Only the admin can perform the operations on the databases. User can only see their details or information’s.
* There will be additional backup for any kind damages or data lost.

1. **Site Adaptation Requirements:**

The centralized database is used so that the system can communicate to retrieve the information.

## **2.2: Product Function:**

There will be username and password to allow access only to authorized users (Admin). The user can take only their information.

**Software will perform the following functions:**

* Only the admin can modify the data.
* Student can only retrieve their information.

## **2.3: Constraints:**

* There is a backup for system.
* GUI feature available.

## **2.4: Assumptions:**

* The product requires a computer with internet connectivity.
* The system must be able to respond to the database Software within reasonable time.

# ****Technical Requirements****

## **3.1: User class and characteristics:**

* **Administrator**: The admin keeps tracks of entries of students enrolled to the course, entries of new staff to the department, entry of the attendance information etc. Admin can modify the data.
* **User**: User can retrieve the information from database by sending the queries.

## **3.2: Project Scope**

**Coursed are offered in three modes: full time, part time and online.** **The process starts with enrolling the students for courses by making a payment which varies from course to course. The students are informed about their successful enrolments with the timetable which spells out the time and locations of the classes and the teachers teaching the courses.**

# ****Functional and Non-Functional Requirements****

## **4.1 Functional Requirements**:

* Authorized users must be able to login with their identification number (or username) and their password.
* The system shall allow an admin to create Courses, Location, Semester, Cluster, Units and register a Teacher
* The system shall allow an admin to edit, modify, view, and delete the Courses, Location, Semester, Cluster, Units
* The system shall allow Admin can be able to edit, modify, view student and teacher registration
* The system shall allow Admin can be able to enrol a teacher to a course
* The system shall allow student can be enrol to a course
* The system shall allow teacher to prepare lecture schedules
* The system shall allow an admin student to subscribe/unsubscribe to courses
* The system shall allow teacher can see the number of students subscribing to a course
* The system shall enable students to retrieve contact information of students and lecturers of subscribed courses
* The system shall provide the history of a course
* The system shall provide the history of attended courses
* The system shall be able to let students upload personal profile
* The system shall facilitate searches in all static information of courses.
* The system shall allow students to view course grade statistics per semester

## **4.2 Non-functional Requirement:**

1. **Performance Requirements:**

The proposed system that we are going to develop will be used as the chief performance system for providing help to the system in managing the whole database of the student studying in the course. Therefore, it is expected that the database would perform functionally all the requirements that are specified.

* The system should be easy to handle.
* System should give expected performance results.
* The response time should be small.

1. **Security Requirement:**

* We are going to develop a secured database. There are different categories of the users namely administrator, restricted users who will viewing either all or some specific information from the database.
* Depending upon the category of the user the access rights are decided. It means if the user is an administrator then he can be able to modify the data, append etc. All other user has the rights to retrieve the information about database.

1. **Safety Requirement:**

The database may get crashed at any time certain time due to virus or OS failure. Therefore, it is required to take the database backup.

# ****Business Requirements****

**Business requirements are:**

* **Display all the courses and locations for a teacher teaching the courses any semester.**
* **Display all the courses which are not offered in the any semester.**
* **Display all the units/subjects which are not allocated to any course in any semester.**
* **Display cluster units for any course and semester.**
* **Display history of all the courses a teacher has taught in the past.**
* **Display all the teachers based on a particular location.**
* **Display results of any student for a course and semester.**
* **Display enrolments of any student for a course and semester.**
* **Display timetable for a course being offered at all locations.**
* **Display a list of all part time teachers for a semester and location.**
* **Display students who have enrolled but have not paid the fees.**
* **Display part time/full time students for a semester and location**
* **Display all the full-time teachers who are teaching at locations other than their based location.**

# ****User Scenarios and Persona****

This Student Management System project is divided into 3 modules

* Administrator
* Teacher
* User

**Module Description**

**Admin**: Admin is a person whose responsibility is to maintain the database that contains each data regarding the all the student. Admin can add all the details into the database, can be able to delete student details and can update the student fee details.  
Admin has some other responsibilities they are  
• Admin is can maintain the course details of each year and semester.  
• Admin can generate the reports of the students and teachers.

**Teacher**: Responsibility of the teacher is to login into the site and can view his/her Course and Cluster details and can able update his/her personal details if there are any wrong details are present. Teacher can be able to view student register to a course.

**User**: Here the user means the student. The responsibility of the student is to login into the site and can view his/her fee details and can able update his/her personal details if there are any wrong details are present. Whenever the student will register his/her name then the student will be given by one individual username and password. When the student will type the correct username and password then they will enter another page. In that page, the student can select two options that are updated details and view details. A student can be able to update his/her personal details and can be able to view the fee details but cannot update the fee details.

# ****Context Diagram:****

A Data-Flow Diagram (DFD) is a graphical visualization of the movement of data through an information system. DFDs are one of the three essential components of the structured-systems analysis and design method.

276 Marini Abu Bakar et al. / Procedia Social and Behavioral Sciences 18 (2011) 273–281

c. Status of all students under a lecturer's supervision is displayed in a single screen so that the overall

progress can be viewed.

d. Notification messages are sent through email to remind students and lecturers on important dates. Students

will be alarmed when the deadline is fast approaching, turned up or passed.

e. Generates report that lists the overall student’s status for supervisor reviewing and administrative purposes.

4. Log book module

a. Logs the discussion during meetings.

b. Provides separate areas or layout for entries regarding system development process and writing.

c. Is able to upload reference documents.

d. Is able to upload report documents for evaluation and reference.

5. Administrator module

a. Managed by department officer.

b. Is able to assign students to supervisors.

c. Sets the system development schedule and chapters writing.

d. Records the submissions of document to the department office.

By clustering the system functions into modules, we hope that designing of processes and system implementation

will be easier. Any requirements for new functionalities to be integrated into the existing modules will be done after

the system is being tested and evaluated by the stakeholder.

3. System Design

Figure 1 shows the context diagram of Final Year Supervision Management System. Context diagram is a level 0

of data flow diagram which illustrates the overall system. It also defines the scope and boundary between the system

and an external entity. This diagram also shows the input and output flows of data to and from the system and

external entities.

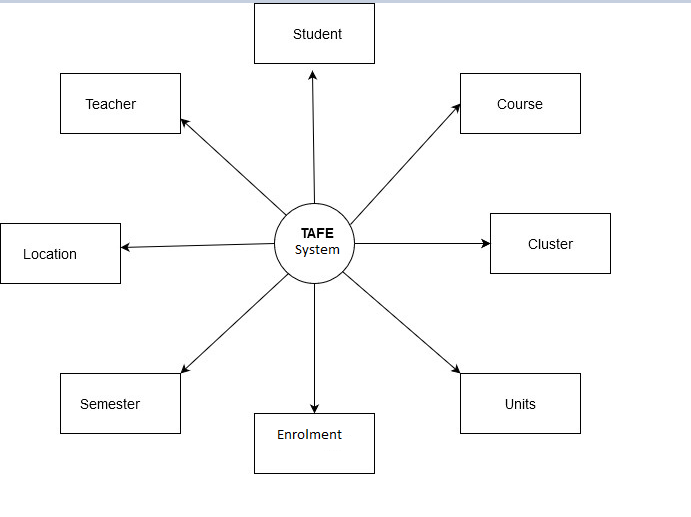
Figure 1 shows the context diagram of Final Year Supervision Management System. Context diagram is a level 0

of data flow diagram which illustrates the overall system. It also defines the scope and boundary between the system

and an external entity. This diagram also shows the input and output flows of data to and from the system and

external entities.

Figure 1 shows the context diagram of TAFE System Management System. Context diagram is a dataflow diagram which illustrate the overall system. It also defines the scope and boundary between the system and an external entity. This diagram also shows the input and output flows of data to and from the system and external entities.



# ****Use case Modelling****

The following figure shows the use case modelling of the student management system:



## **7.1 Use cases**

A use case is a list of steps, typically defining interaction between a role (known in Unified modelling language (UML) as an “actor”) and a system to achieve a goal.

In system engineering use cases are used at a higher level that within software engineering, often representing missions or stakeholders’ goals.

|  |
| --- |
| **Use case 1:** Login.  **Primary Actor:** All Users  **Precondition:** User clicks the “Login” button from the home page  **Main Success Scenario:**   * User enters his/her ID or username and password and launches the login process. * System verifies whether the username/password pair is valid. * System displays the “Welcome” page, tailored to user’s permissions   **Exception Scenario:**   * User enters an invalid username/password combination. * System denies access to user. * System displays error message and prompts user to retry. * User may repeat login procedure. |

|  |
| --- |
| **Use case 2:** Update an entry of the student.  **Primary Actor:** Admin  **Precondition:** Admin has logged in.  **Main Success Scenario:**   1. Admin checks all the previously filled data. 2. Admins retrieve the student data which is meant to update. 3. Admin updated the selected student data from the database. 4. System confirm the modification.   **Exception Scenario:**  **a)** There is no such student data, which the searched for.   * System shows error message. |

# ****Domain Modelling****

Domain modelling is the activity of translating an informal description of a domain into a domain model.

Domain models are used during the requirements gathering phase to clarify important domain-specific terms.

Domain models are useful during the requirements analysis phase to identify the entity classes in the Entity-Control-Boundary architecture.

*Course*

CourseId:Integer

CourseName:String

StartDate:Date

AddCourse()

Update()

ViewAll()

Delete()

*Student*

StudentId:int

StudentName:String

Email:String

AddStudent()

Update()

Delete()

ViewAll()

*Teacher*

TeacherId:Integer

TeacherName:String

Email:String

Add()

Update()

Delete()

ViewAll()

*Cluster*

ClusterId:Integer

ClusterName:String

Add()

Update()

ViewAll()

Delete()

*Unit*

UnitId:Integer

UnitName:String

Add()

Update()

ViewAll()

Delete()

*Location*

LocationId:Integer

LocationName:String

Add()

Update()

ViewAll()

Delete()

*Semester*

SemesterId:Integer

SemesterName:String

Add()

Update()

ViewAll()

Delete()

*Enrolment*

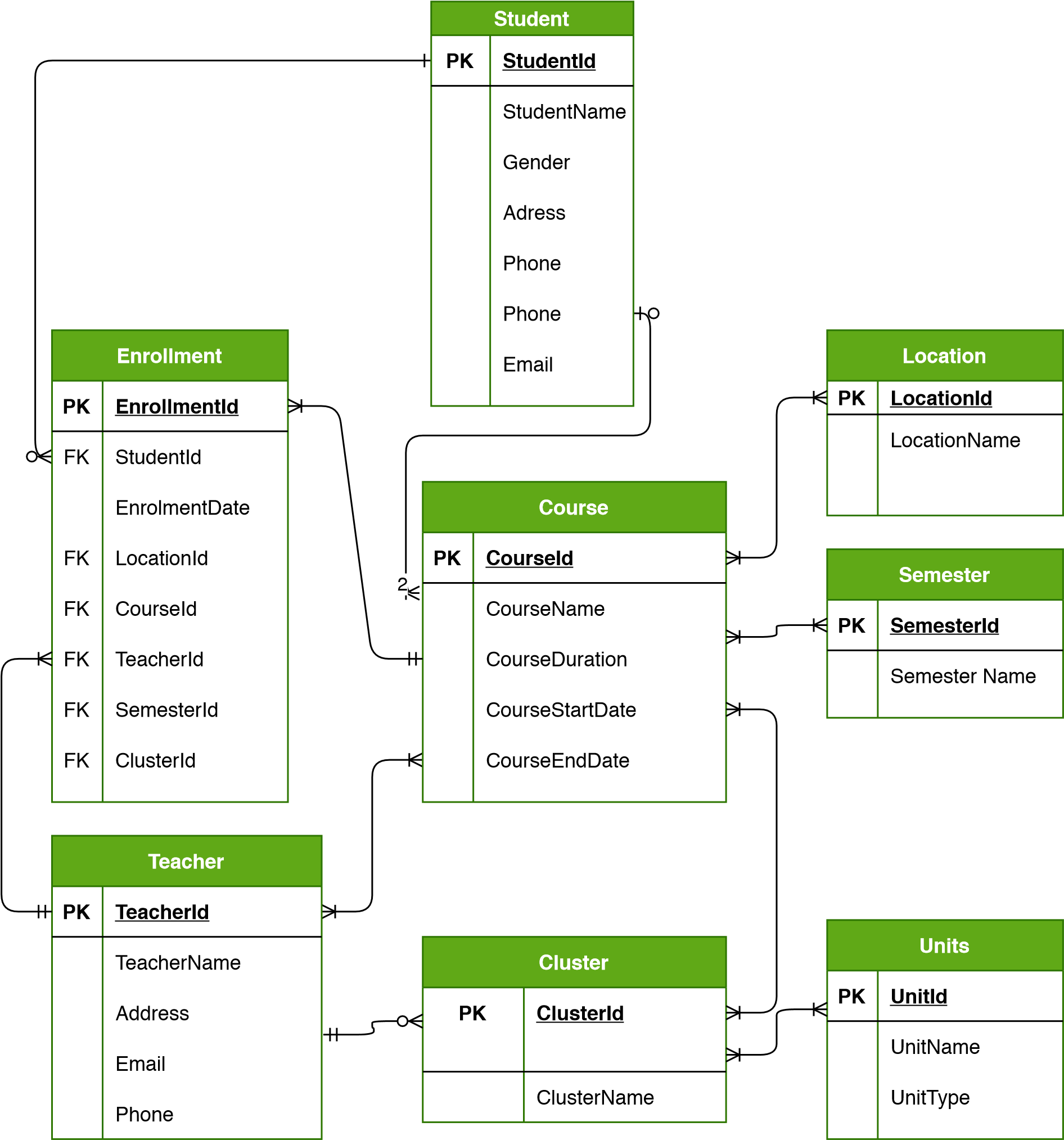
EnrolmentId:Integer

CourseId:Integer

EnrolmentDate:Date

Enrol()

# ****ERD****



# ****System Architecture****

## 10.1 Architectural Design

Generate Report

Manage data

Get info

View info

**User**

**Software**

**Admin**

**Hardware**



Deleting entry

Editing entry

User ID/ password

**Remove**

**Modify**

**Login**

**Admin**

**View info**

**User**

**Generate** **report**

View details

Print Report

## 10.2 Three-Tier Client Server Architecture

In a client server architecture, the functionality of the system is organized into services, with each service delivered from separate server. Clients are users of these services and access servers to make use of them. We will use this 3- Tier Client Server Architecture because, when data in a shared database must be accessed from a range of locations. Because servers can be replicated, may also be used when the load on a system is a variable.

**• Data Tire**

The data tire maintains the applications data such as Users’ data, Teachers data, unit’s data, courses’ data, timetables’ data, and the SQL queries. It stores these data in a relational database management system (RDBMS). All the connections with the RDBMS are managed in this tier.

**• Middle Tire**

The middle tier (windows application server) implements the business logic, controller logic and presentation logic to control the interaction between the applications’ clients and data. Business rules enforced by the business logic dictate how clients and cannot access application data and how applications process data.

**• Client Tire**

The client tire is the applications user interface connecting data entry forms and client-side applications. It displays data to the user. Users interact directly with the application through user interface. The client tier interacts with the application server to make requests and to retrieve data from the database. It then displays to the user the data retrieved from the server

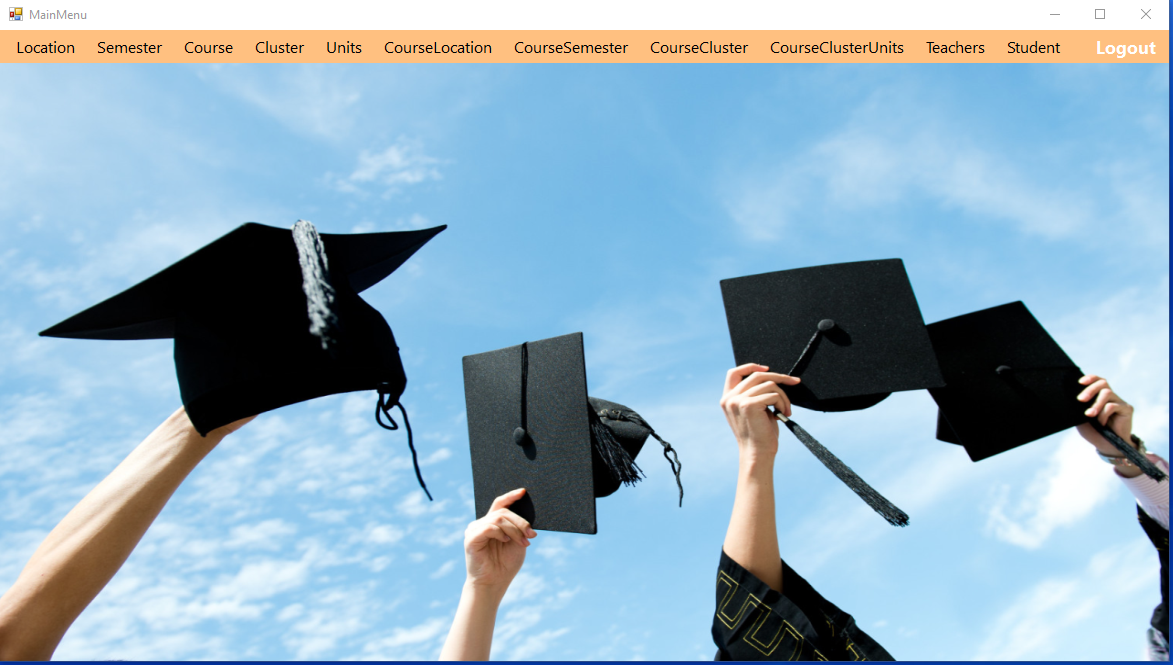
**Example of the 3-tier architecture** in the Student Registration System If a student needs to view his current semester course combination, first he/she has to login to the system. Then he must click “User” option, after he has to login “Summary” option. Then system will display the information. In this process, Login screen, users’ main screen and Subject combination summary screen are defined into the Client tier, data for login information and profile information and LINQ queries for those information are maintained into the Data tier and controller logic for login process and loading profile information from the database are defined in Middle tier.

# ****Conceptual and Concrete UI Design****

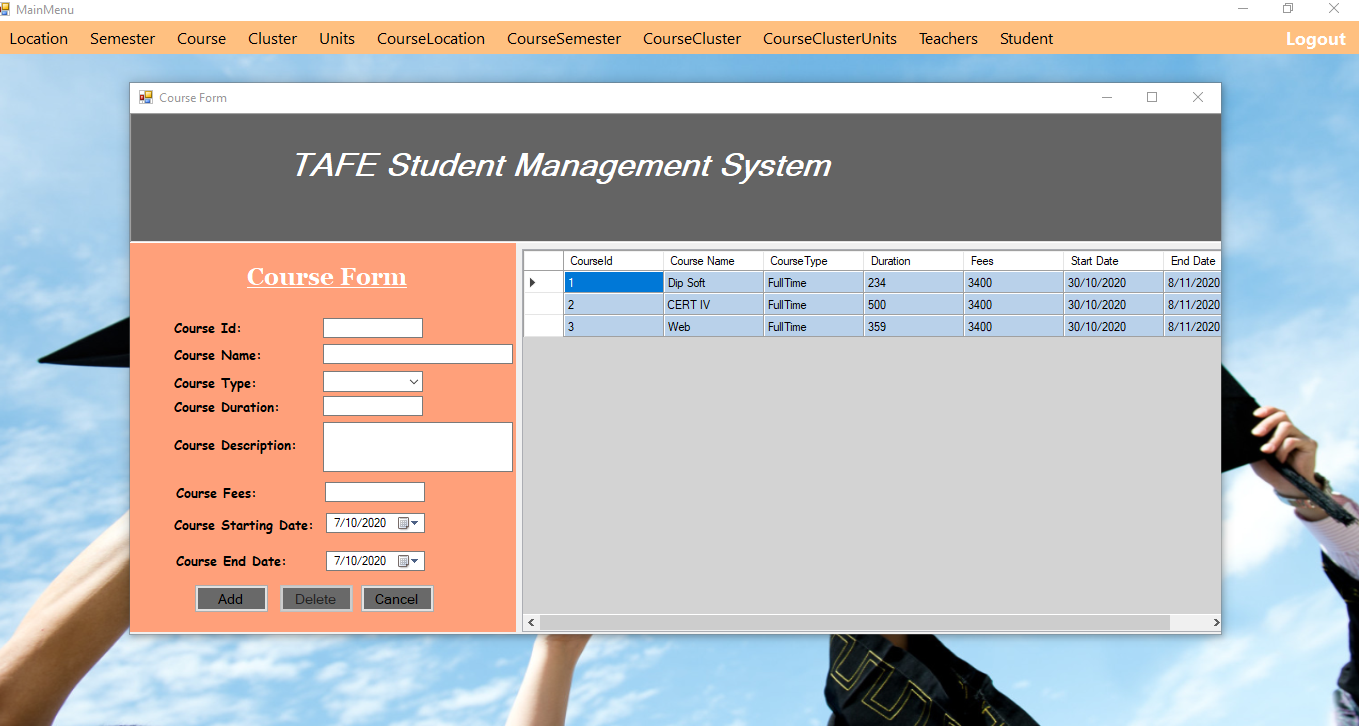
**Login Page**



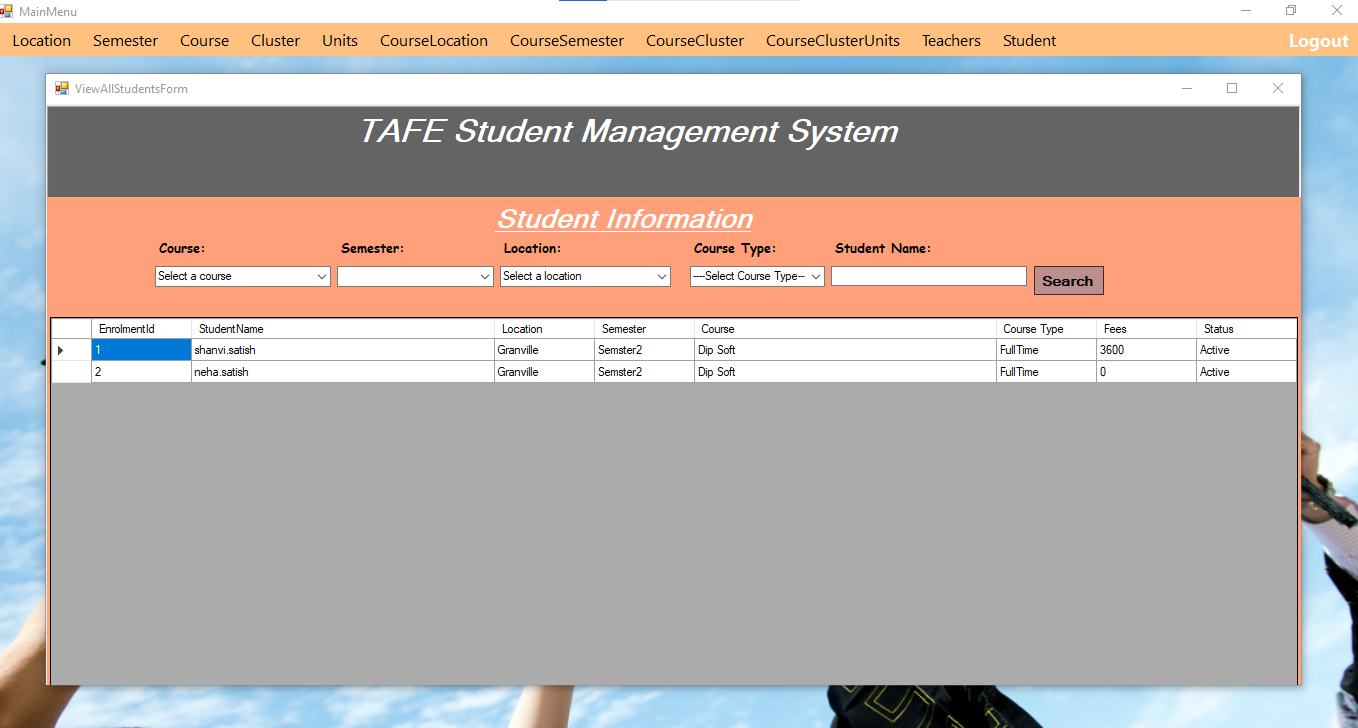
**Admin Main Menu Page**



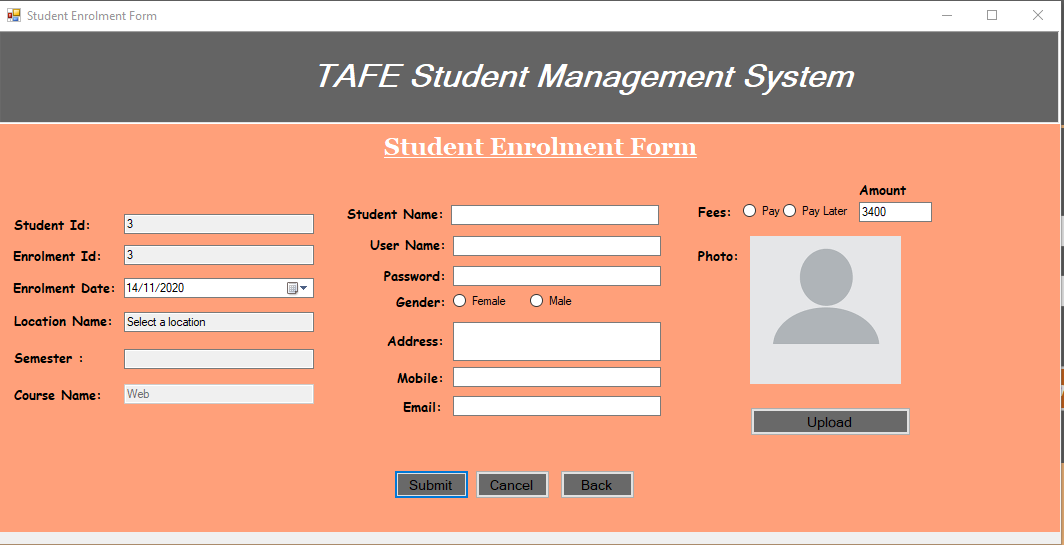
**Course Form**



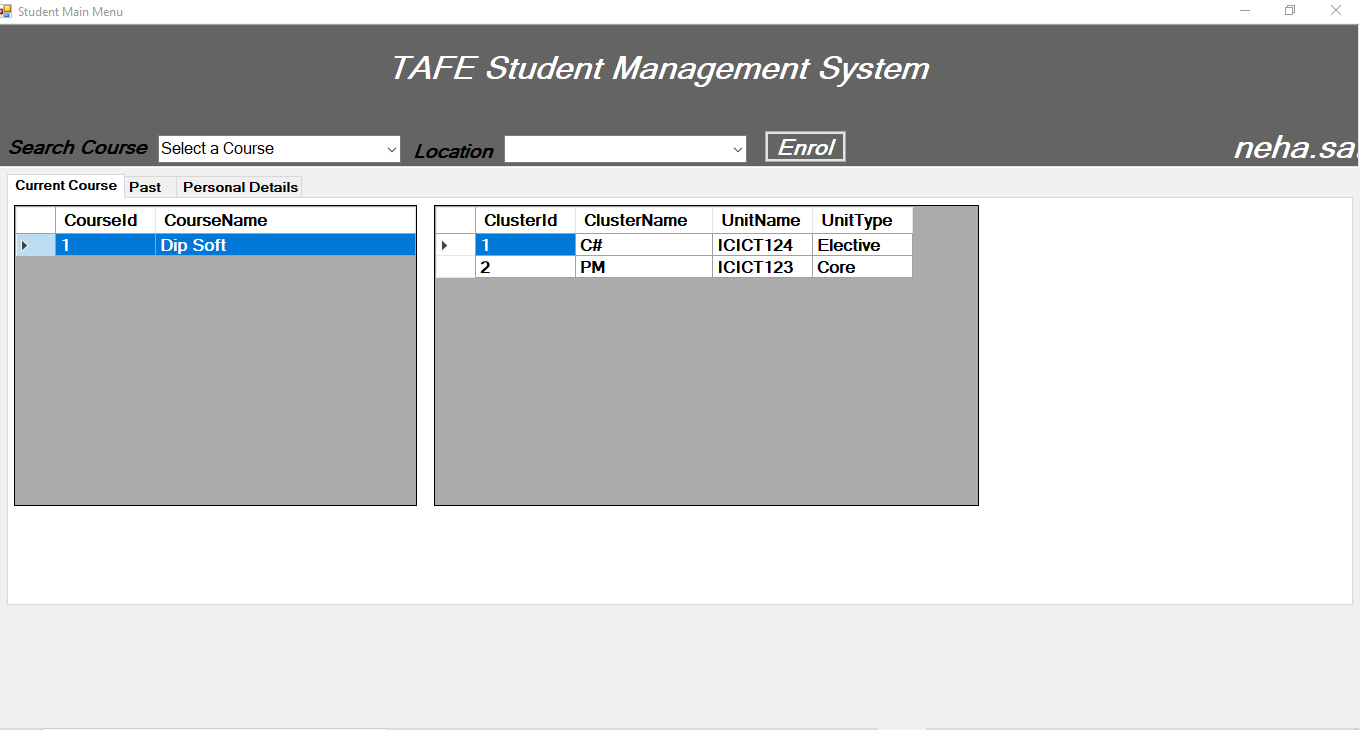
**View Student Information Page**



**Student Enrolment Form**

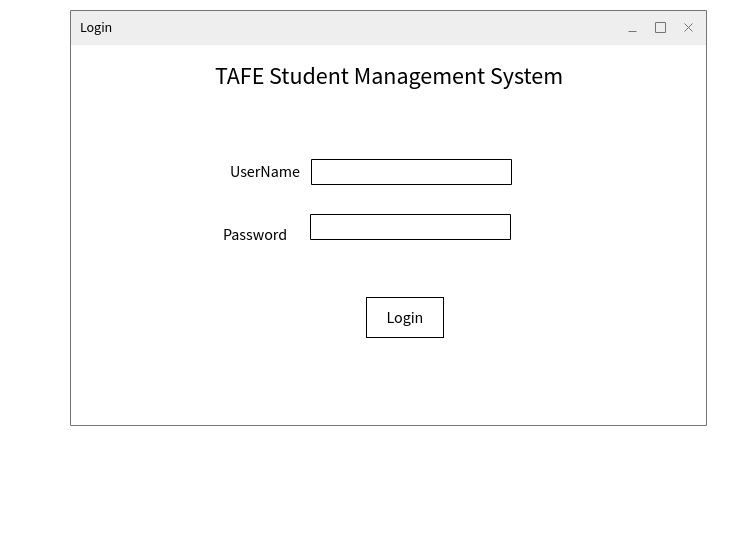


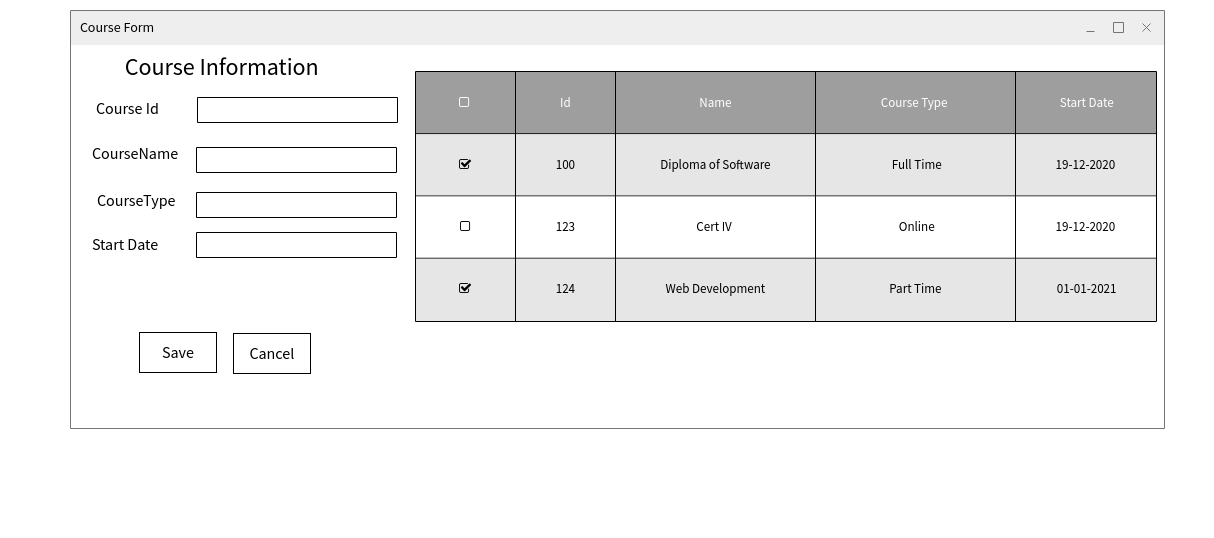
**Student Main Menu Form**



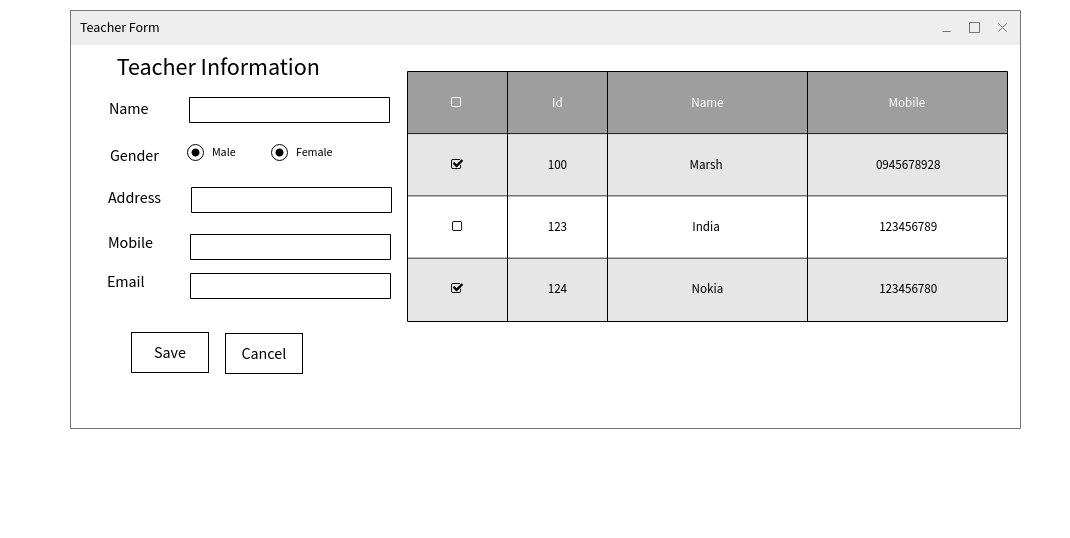
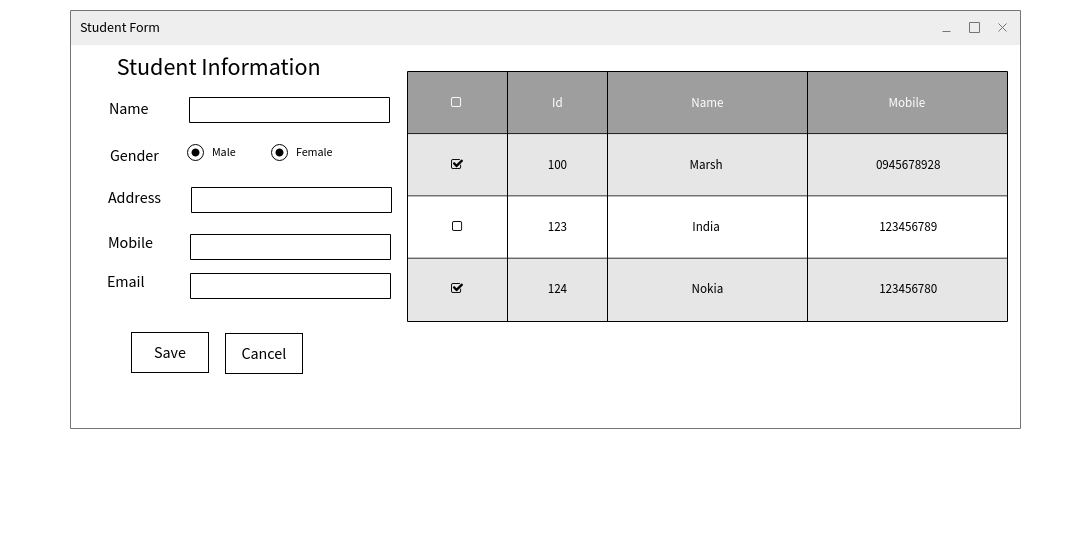
# ****Prototype UI design for both, front and back end system.****

**Login Page:**



**Course Form:**

Teacher Form:

**Student Form:** 

**Data Base Diagram:**

****

# Test Cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Scenario | Requirements | Test cases | Test Data | Result |
| Check Login Functionality | * All correct combinations entered in login/password field will let user in. * All other combinations will be rejected. * Login includes minimum 8 letters and 4 digits/special characters or both (except space character). * Login is not case sensitive. * Password cannot be less than any 8 characters (except space character). * Password is not case sensitive. * “Login in “button is disable unless both fields typed in. | Check response on entering **valid** username and password. | Username: Admin  Password: admin | Login successful |
| Check response on entering **invalid** username and password. | Username: MOHIT  Password: temp | username is invalid |
| Check response when username and password field is **empty** and login button is pressed. | Username:  Password: | Username and password invalid |