Software Design Specification

Student Information System (SIS) for Institute of Information Technology

Version 1.2 Approved

Prepared by <BIT-01 Group 3>

Institute of Information Technology

16th June, 2012

BIT-01 Group 3:

BIT-01 03	Alim Ul Gias
BIT-01 06	Ashraf Siddique
BIT-01 09	Jobaer Islam Khan
BIT-01 12	A.N.M Bazlur Rahman
BIT-01 15	Md. Iftekharul Alam
BIT-01 18	Asif Khan Shakir
BIT-01 21	Kishore Kumar Roy
BIT-01 24	Khalid Hossain
BIT-01 27	Suman Kumar Dash
BIT-01 30	Md. Ruhul Quddus

Table of Contents:

List of Figures:

1. Introduction

This section provides an overall bird's eye view of the system. It defines what the system is supposed to do and what the system will cover, as well as what the system will not include. It also includes a brief overview of the whole document.

1.1 Purpose and Scope

A student information system (SIS) is a software application for education establishments to manage student data. These systems are designed specifically to cater the need of a particular educational organization, and therefore the actual functionalities always vary greatly. Typically, A SIS maintains information about handling inquiries from prospective students, handling the admissions process, enrolling new students and storing teaching option choices, automatically creating class & teacher schedules, handling records of examinations, assessments, marks, grades and academic progression and maintaining records of absences and attendance.

In this particular case, the system is only a part of the Student Management System. It maintains the attendance information of the students, and performs additional tasks to help the attendance management process.
The purpose of the system is to simplify the task of attendance management. The key task of the system is to assist the teachers in the task of taking the attendances in each class. To achieve this, the system must also perform a set of other tasks. They're listed below:

- The system needs to have the ability to add/remove courses, teachers, students and coordinators.
- The system will have the feature to schedule and cancel classes if required.
- The system should have interfaces to enforce rules to the percentages of the attendances for a particular student.

Even though these features may not seem like a part of the attendance system, they need to be included, because it is assumed that these features are not available elsewhere. Therefore, for obvious reasons, they're invariably a part of the system.

1.2 Overview of the Document

The document aims to provide an insight into the overall design of the whole system. The whole document is divided into 10 chapters. Each of the chapters will describe different aspects of the design.

Section	Overview
Design Considerations	This section focuses in the assumptions of the operating environment, and the hardware and software requirements for running the system. Additionally, it'll include the general constraints, and goals and guide lines of the system
System Architecture	This section contains two things: description of the components of the system, and class diagram
Architectural design	This section contains architectural representation, as well as activity diagrams
Data Design	Most of the contents of this section cover about the database of the system Primarily; it'll contain the entity-relationship diagram, and the data dictionary.
Use case	Provides description about the actors and their roles, and how the actors interact and in which sequence they interact with the system.
Data Flow Diagram	Contains context level diagram, 0-DFD and 1-DFD.
Conclusion	Wraps up the document

1.3 General Constraints

The general constraints on the development of the system are as follows:

- The system will not be accessible to unauthorized users.
- All data transmitted to the central database will be encrypted.
- The system will be completed by the end of September 2012.
- The development team will use this SDS to implement the system.

1.4 Goals and Guidelines

The goals of the SIS are to deliver the following:

- Central, up-to-date repository of information on all courses and course offerings; all prospective students, applicants, and matriculated students; student academic history; student housing; and degree progress that is easy to access and manage.
- An efficient and effective Admissions' environment that includes campaign management, document and image management, and travel management.
- An efficient course management system for entry and management of course descriptions, rules, and restrictions that can be easily accessed across the campus.
- A self-service system that is highly *intuitive* and *efficient* for student and faculty.
- A tool where data is safe and secure.
- A well-designed system that can handle thousands of concurrent users, can process thousands of requests simultaneously, is stable, and integrates easily with other applications.

2. System Architecture Description

The core models and functionalities which are derived from the functional requirements are generated as some basic components of the systems. By this section it will be very clear for the development team to find out all these at a glance.

2.1 Overview of Modules and Components

The Student Information System should have the four basic modules-

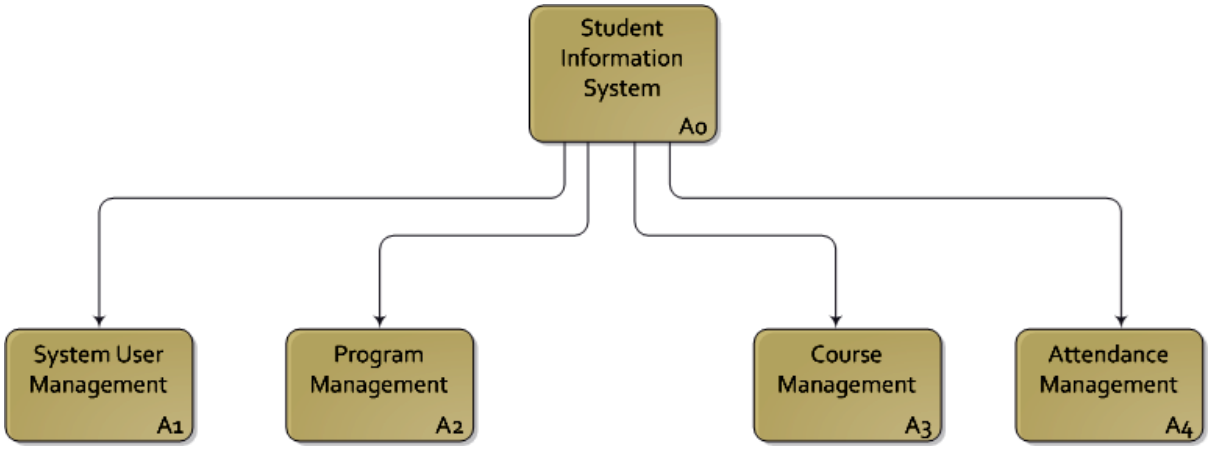


Fig 2.1: Basic Modules of the Student Information System

The system user management will handle the addition and update of different users and their roles throughout the system. Its components will be-

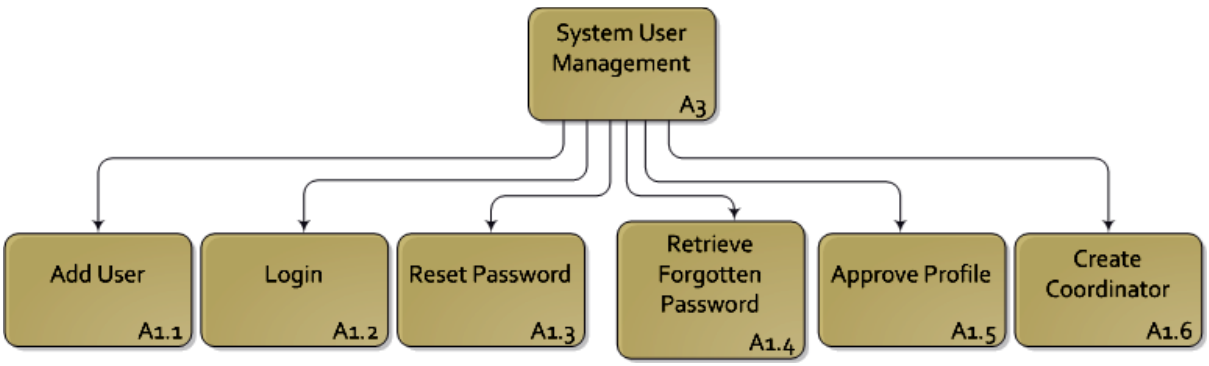


Fig 2.2: Components of User Management

The program management must have the following components-

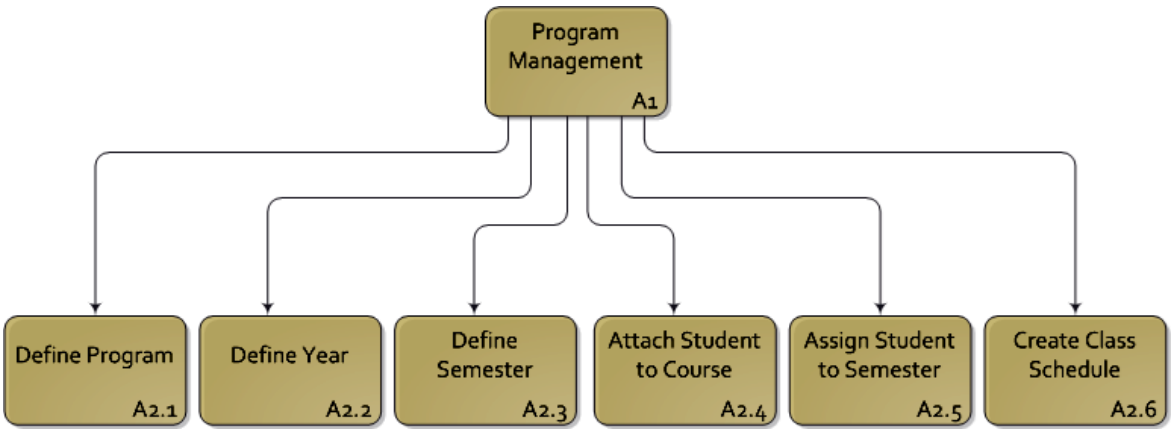


Fig 2.3: Components of Program Management

The course management includes-

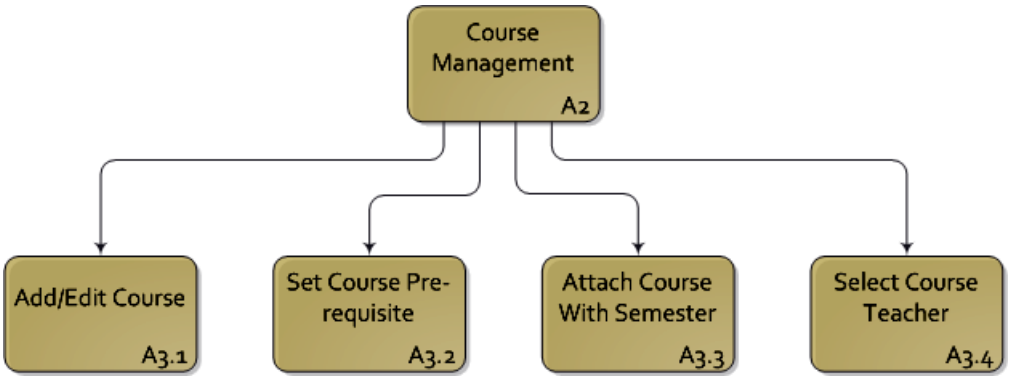


Fig 2.4: Components of Course Management

The Attendance management system should contain the following features-

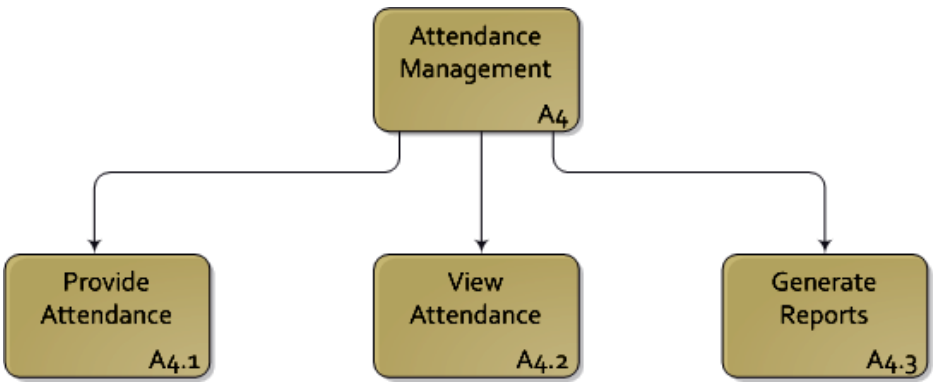


Fig 2.5: Components of Attendance Management

2.2 Class Diagram

The total system will follow the MVC pattern for these the model and controller class diagrams are given below:

The first figure describes the model classes.

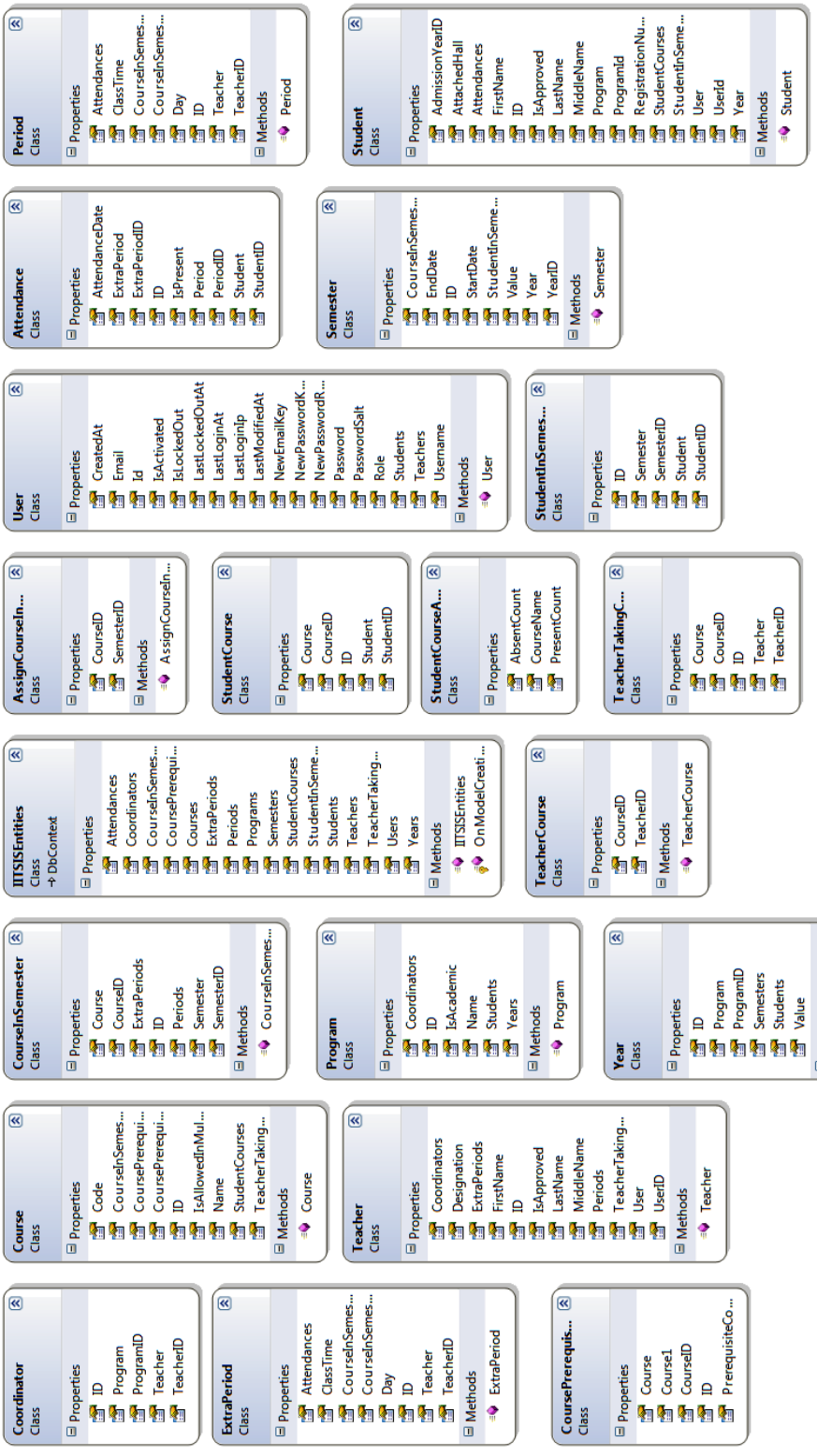


Fig: 2.6 Model classes for SIS

In the following figure the controller classes of SIS is given by which we can control the views and can get access from anywhere in the application.

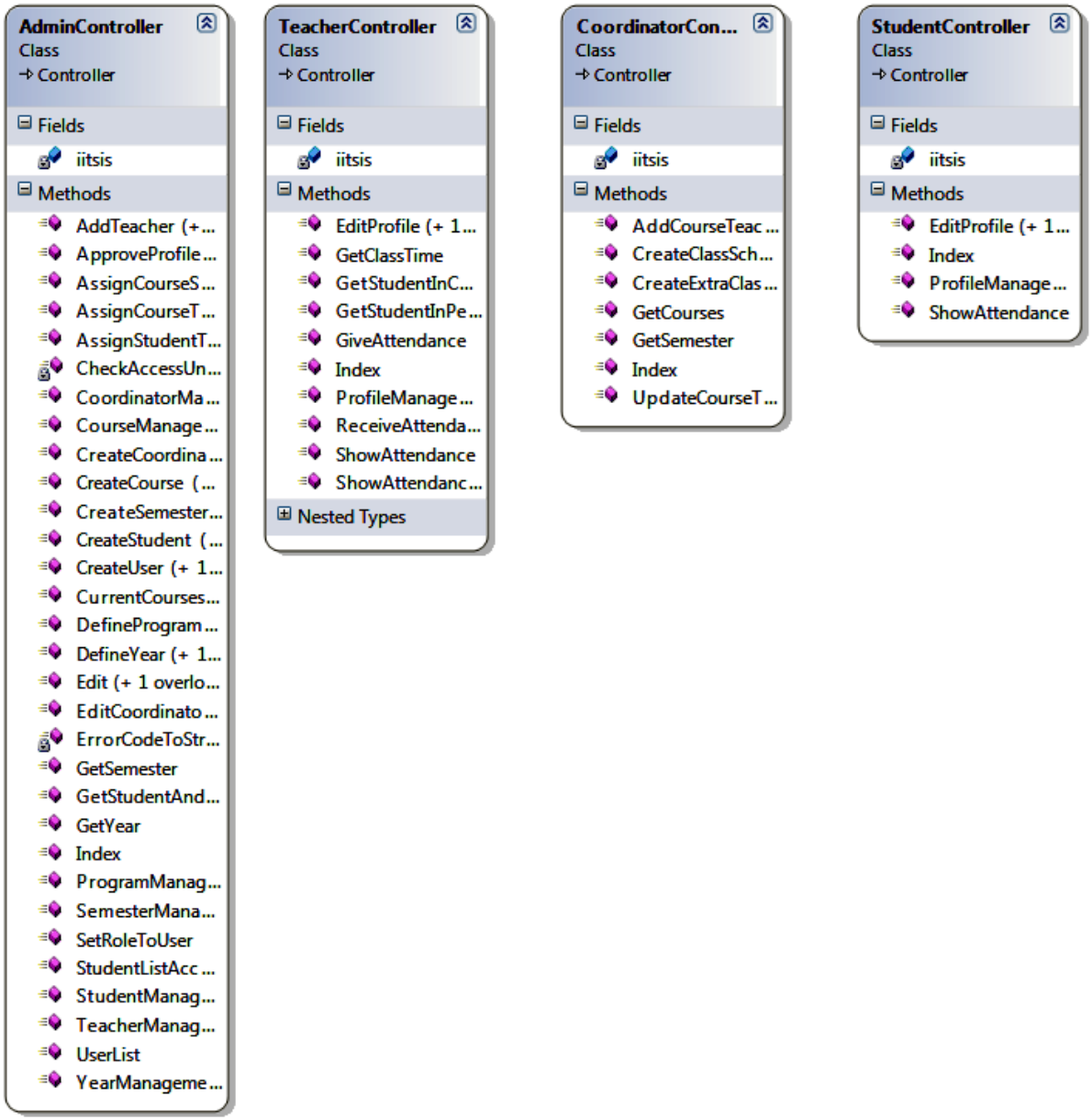


Fig: 2.7 Controller classes for SIS

3. Architectural Design and Sequence Diagrams

In this section the business and data layer activities are shown by which the non-functional requirements are fulfilled and ease to the vendor to realize the system as well.

3.1 Architectural Representation

Here, the architectural representation is shown through the following figure.

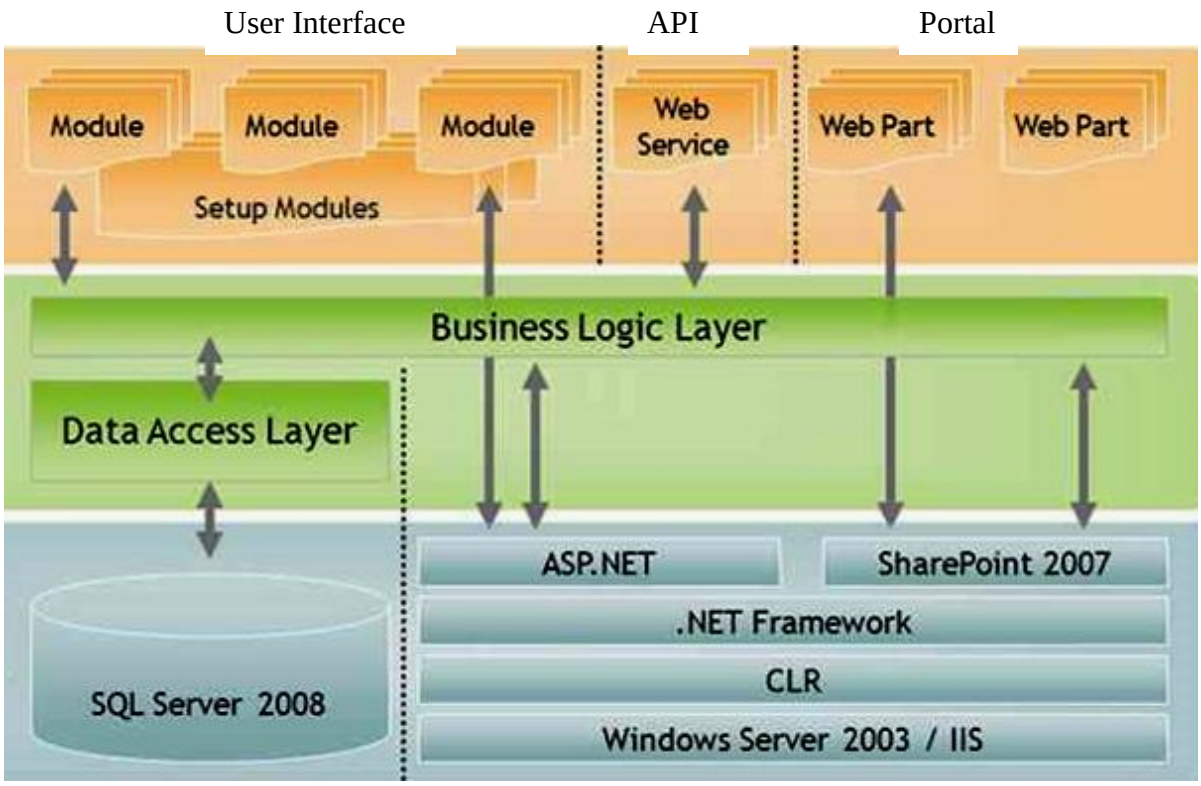


Fig: 3.1 Access Layer of SIS

Here is the network diagram and overall representation of the system. The diagram is given in the next page.

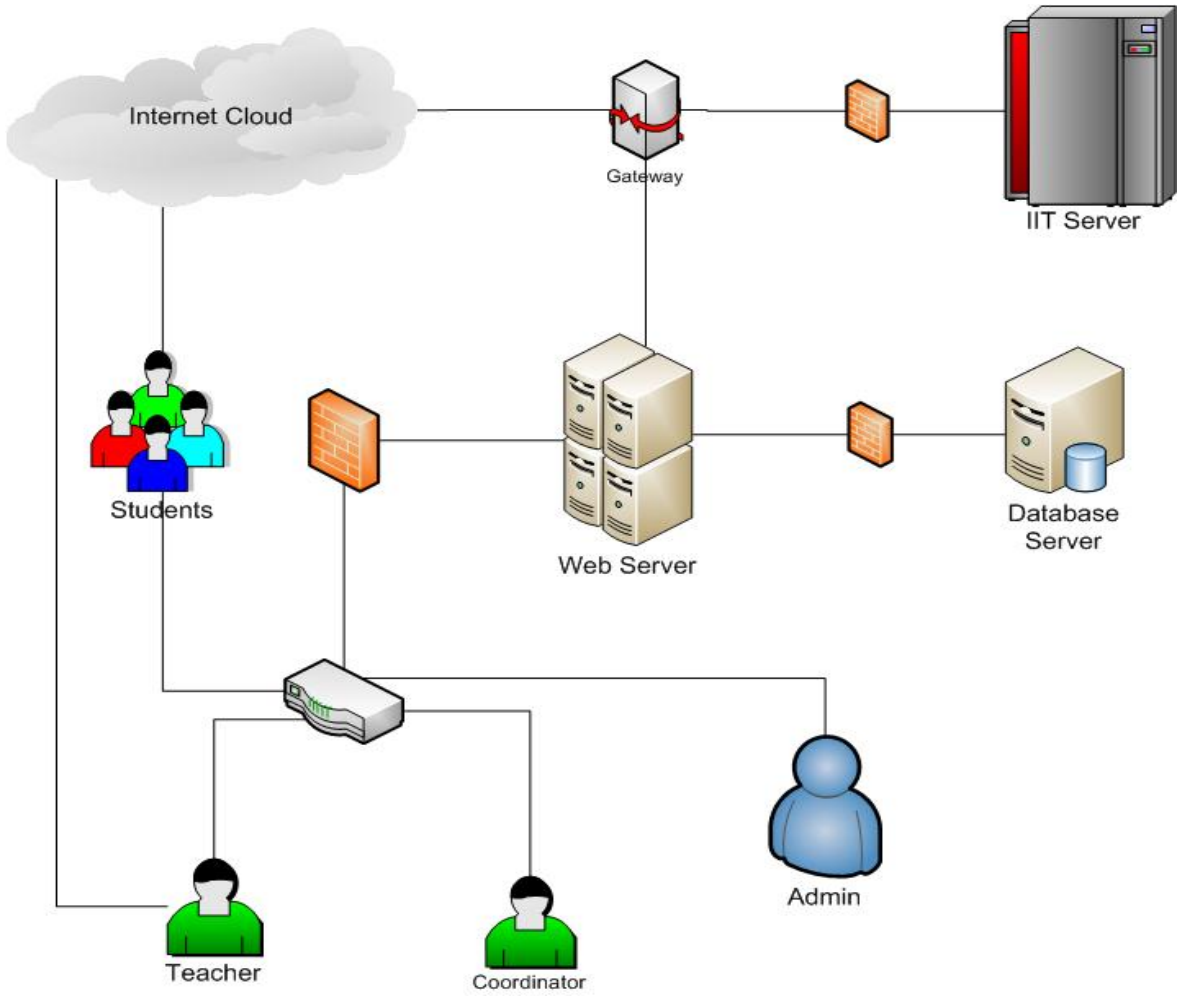
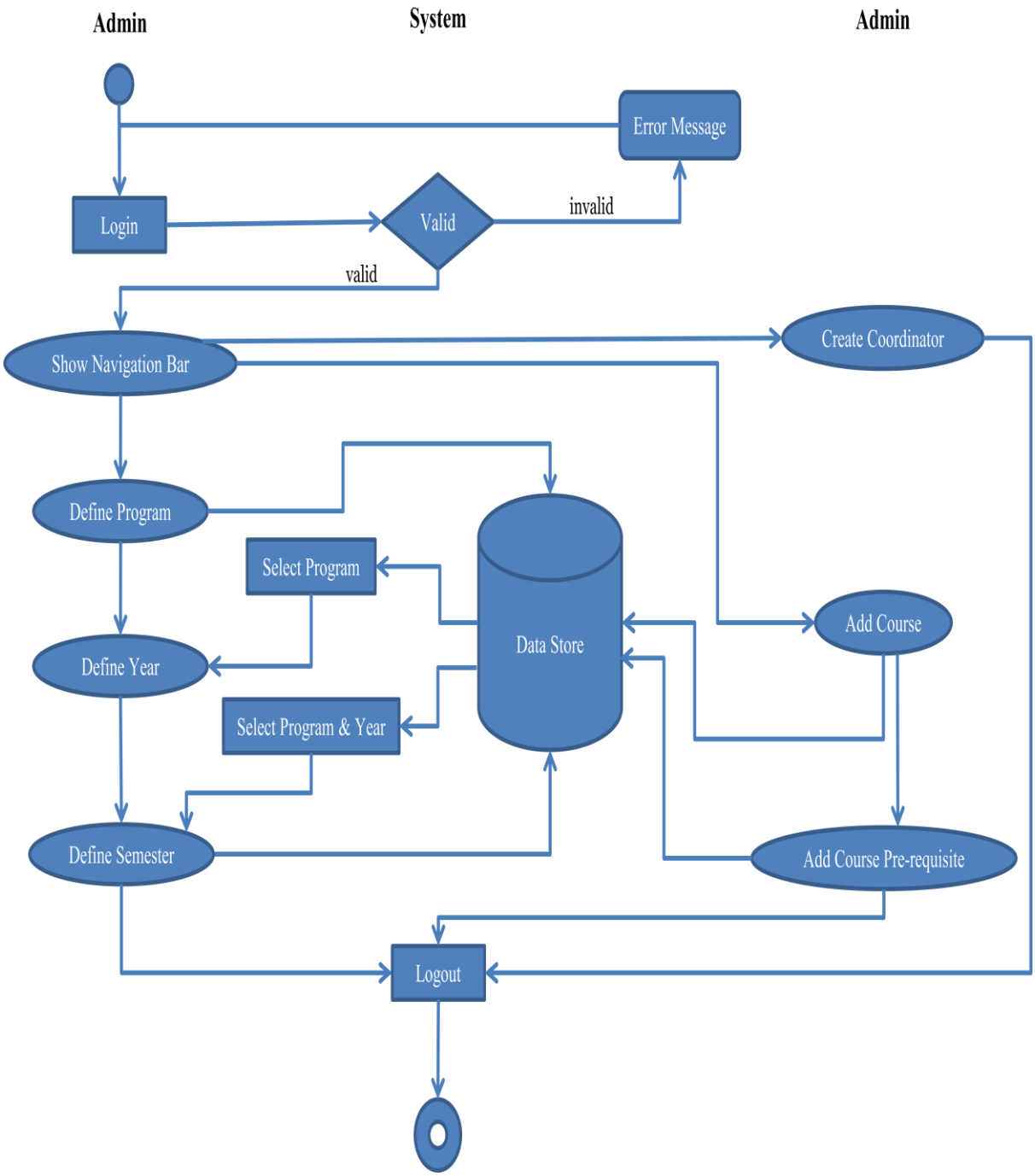


Fig: 3.2 Architectural Representations

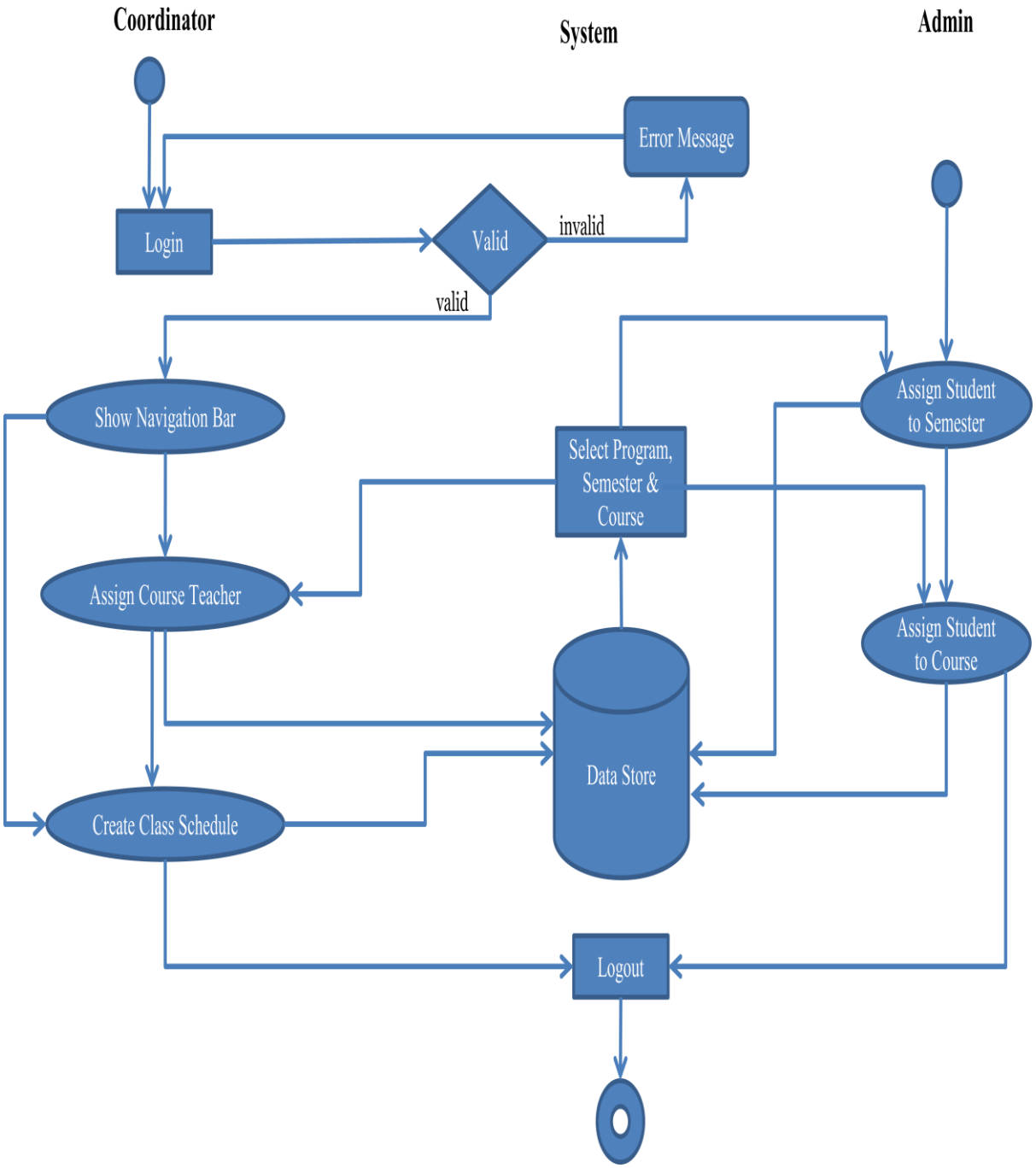
3.2 Activity Diagram

To describe the SDS more specifically there are some activity diagrams to elucidate the system more distinctively.

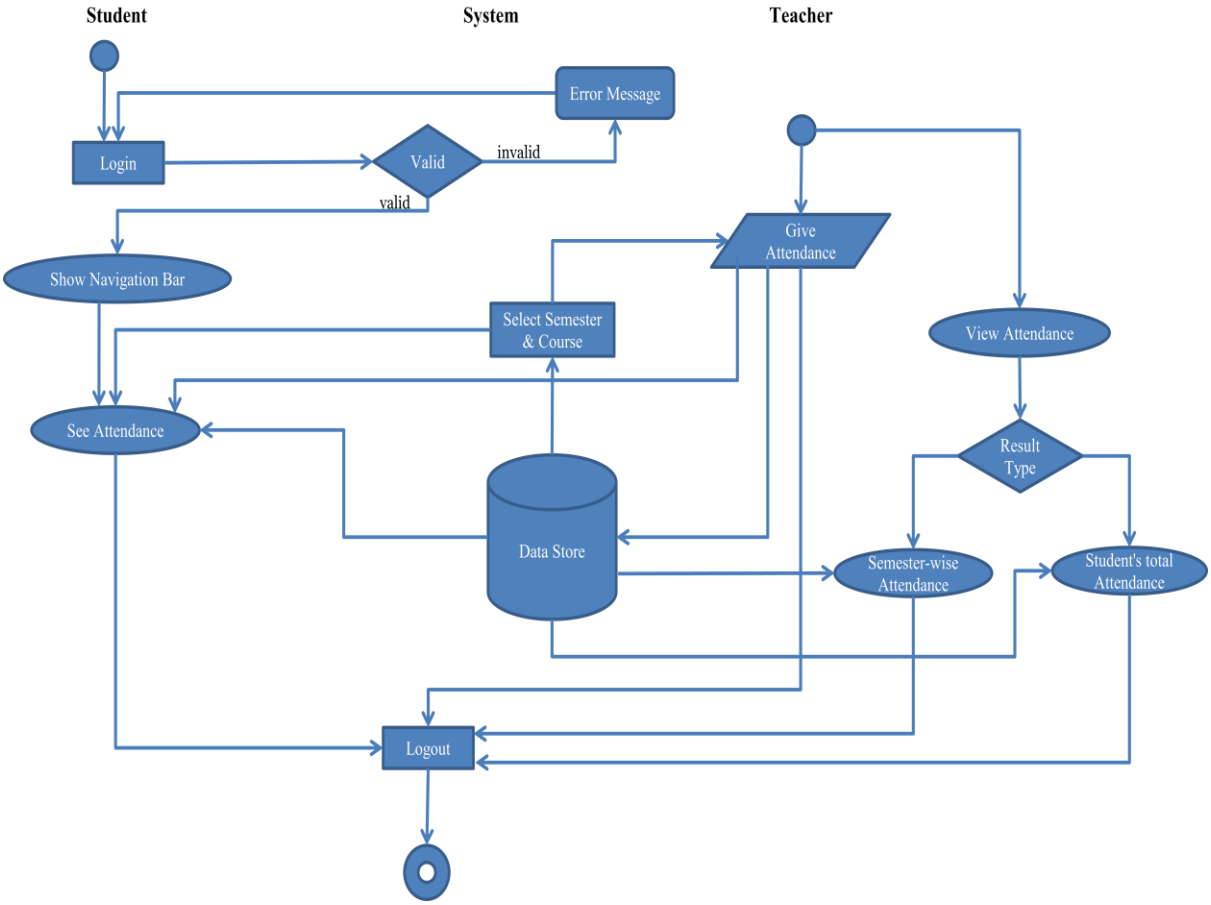
3.2.1 Academic Tasks by Admin



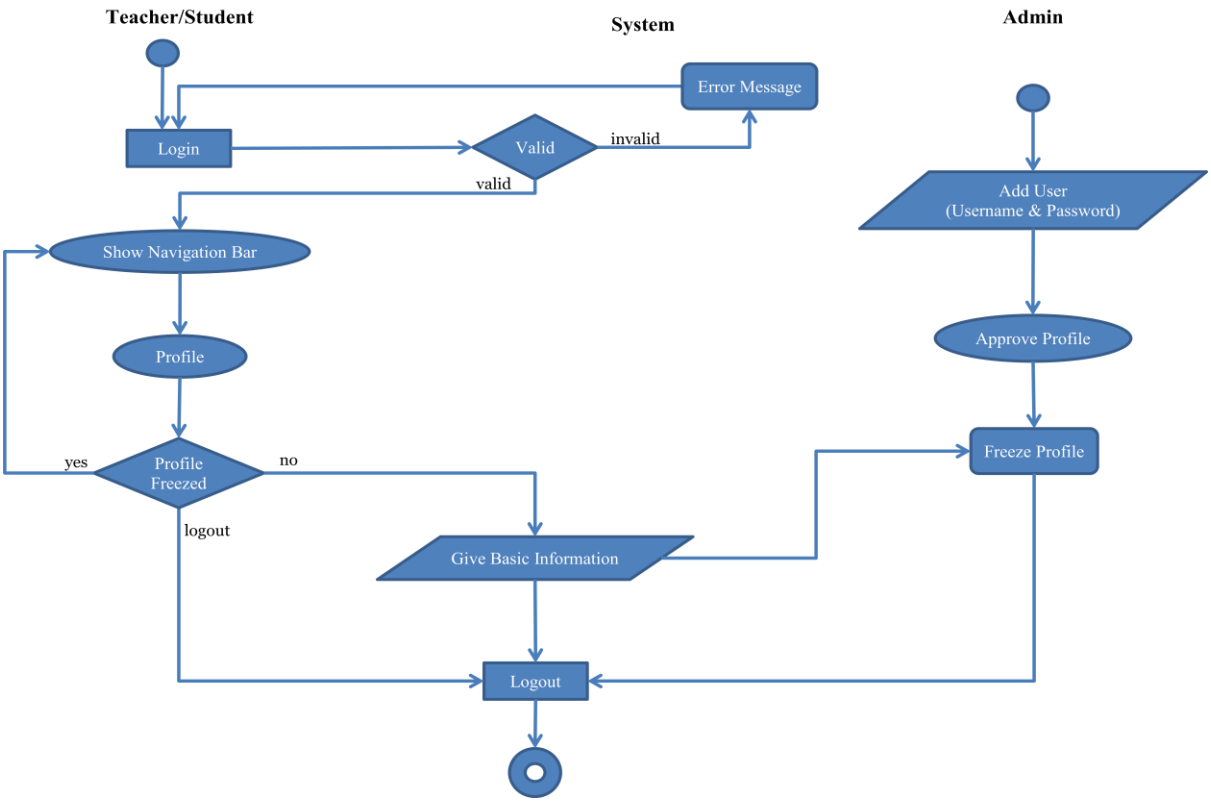
3.2.2 Enrollment of Student to Course



3.2.3 Attendance Management

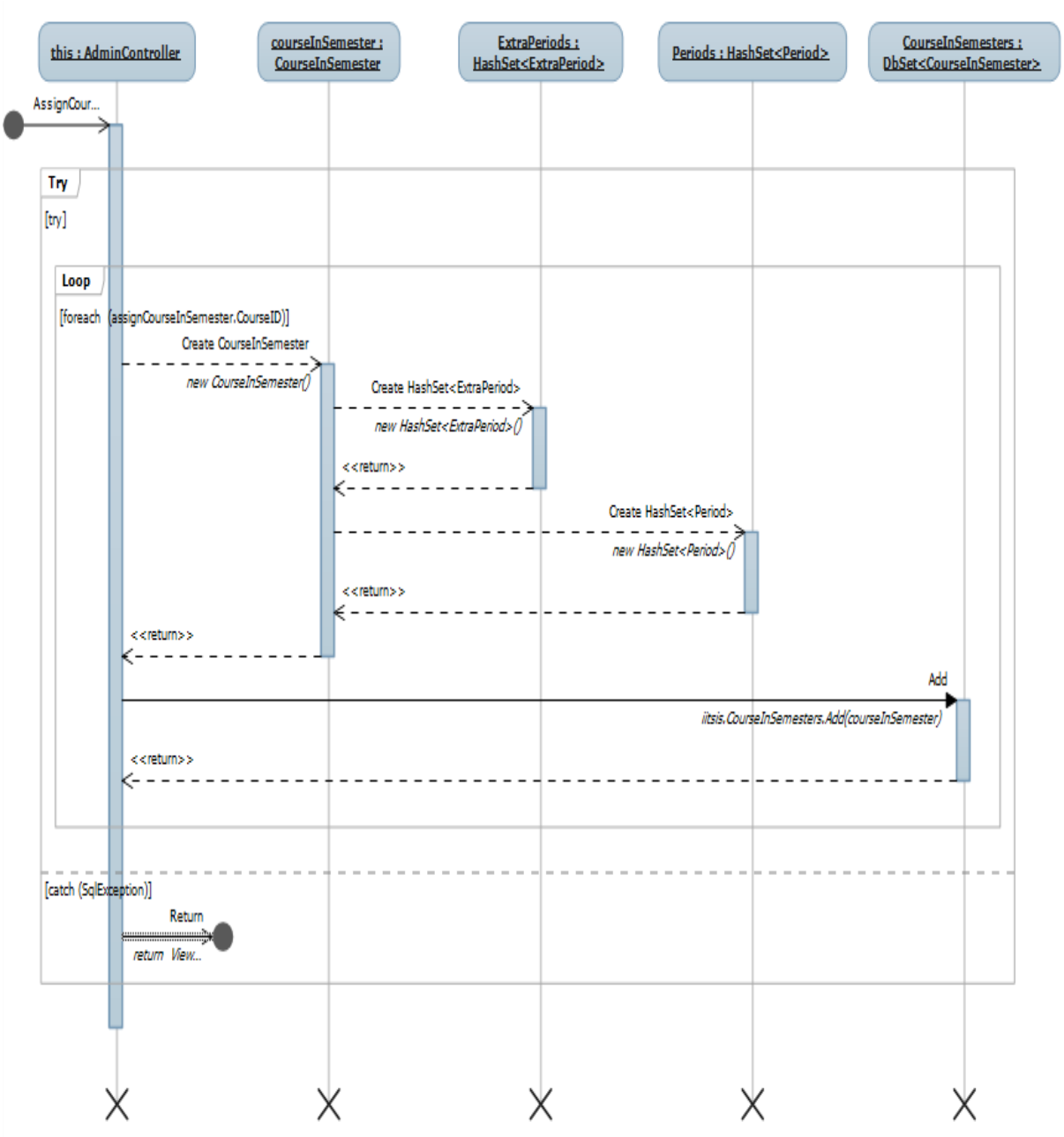


3.2.4 User Profile

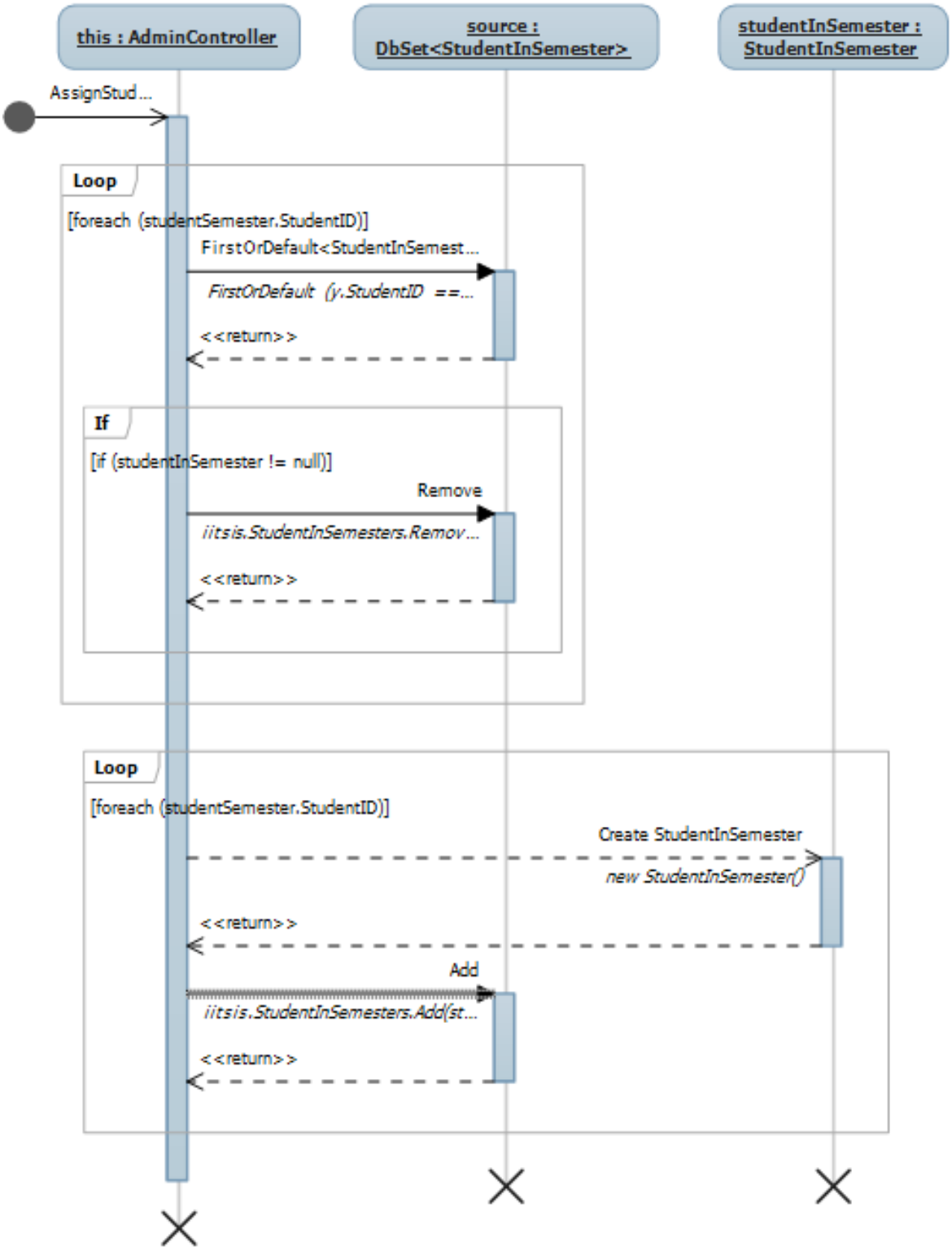


3.3. Sequence Diagram

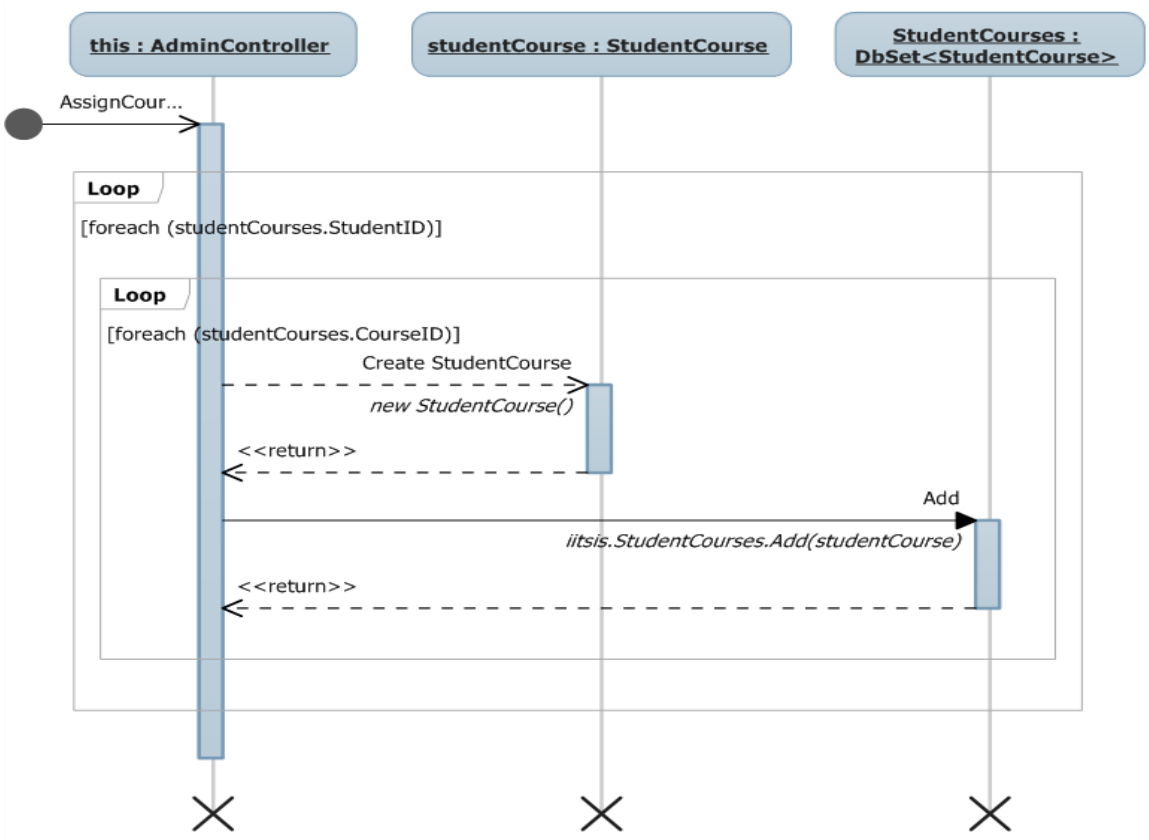
3.3.1 AdminController_AssignCourseSemester



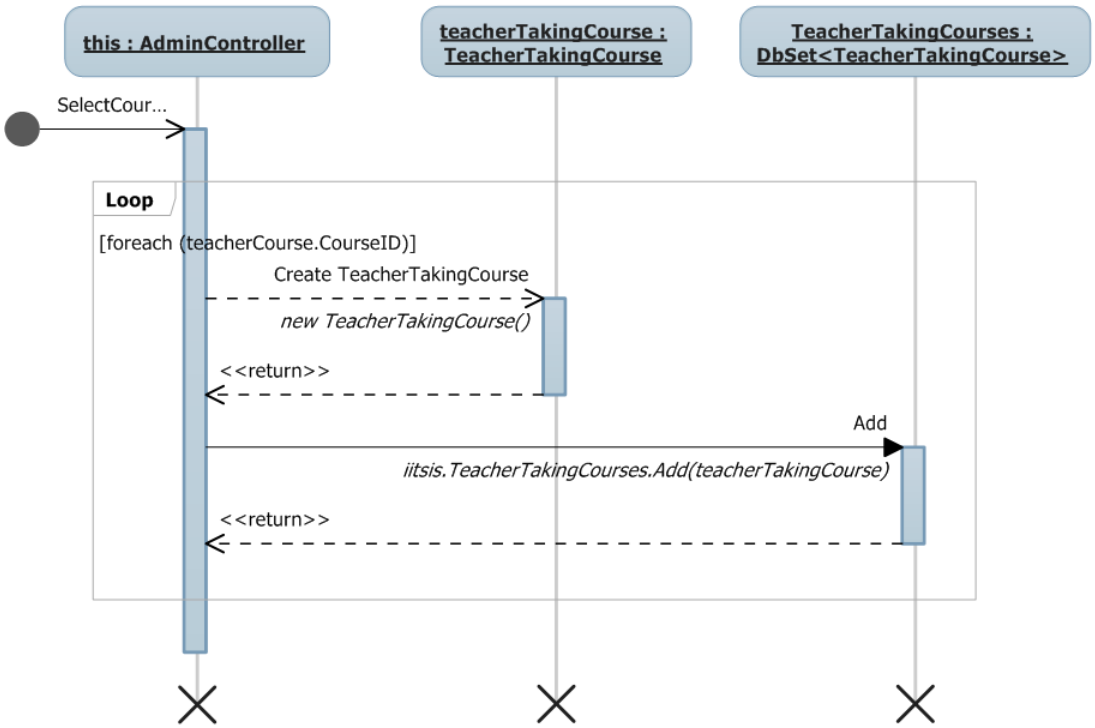
3.3.2 AdminController_AssignStudentSemester



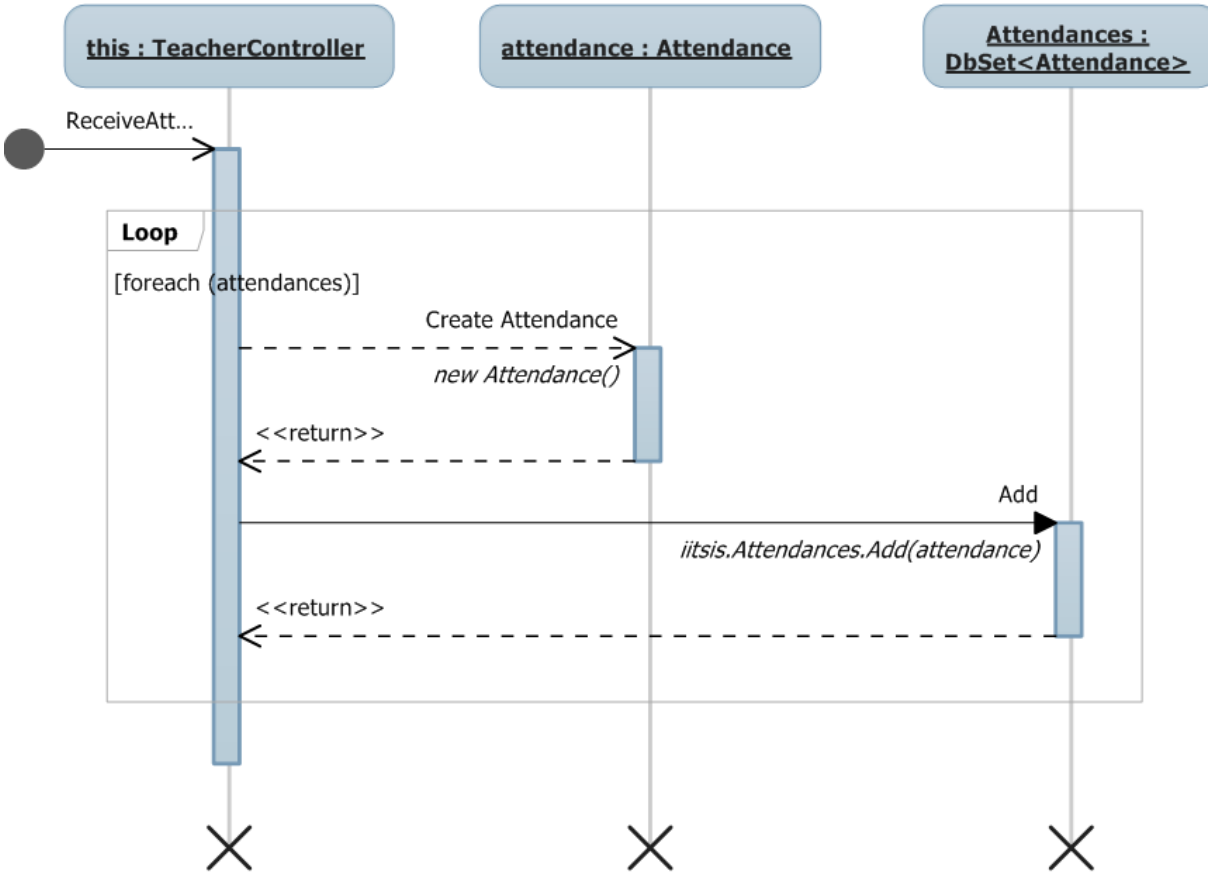
3.3.3 AdminController_AssignCourseStudent



3.3.4 Coordinator_AddCourseTeacher

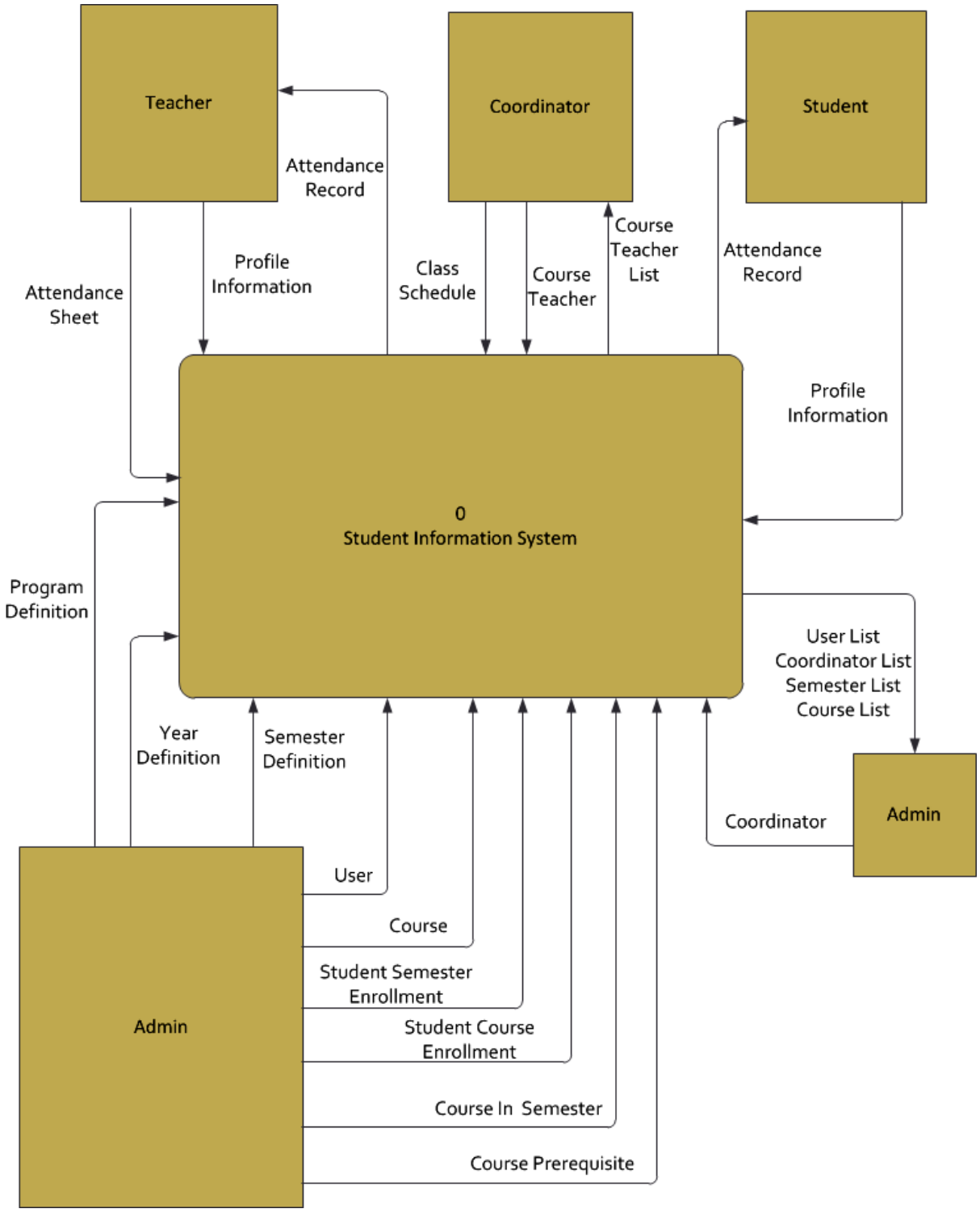


3.3.5 TeacherController_GiveAttendance

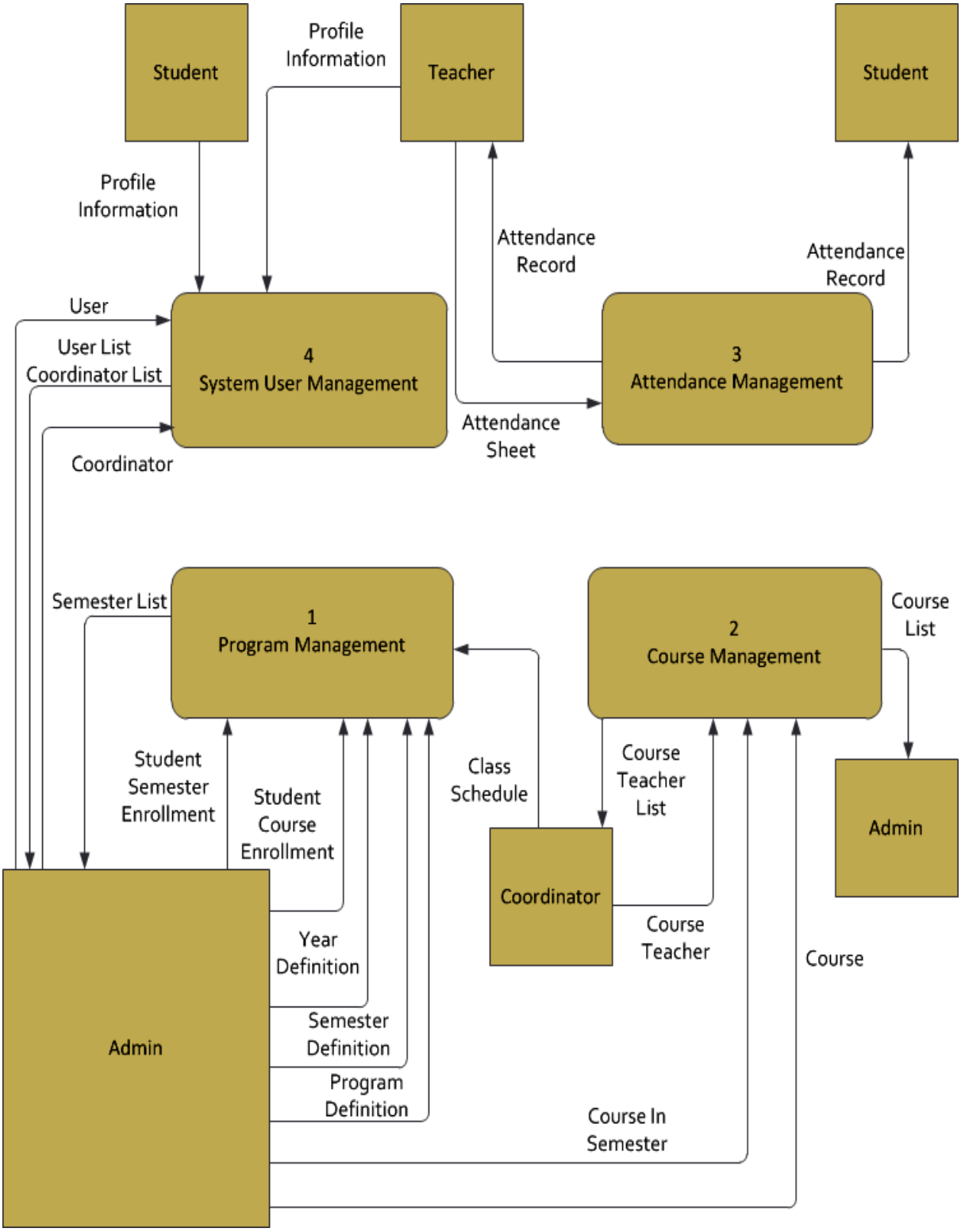


4. Data Flow Diagram

4.1 Context Diagram

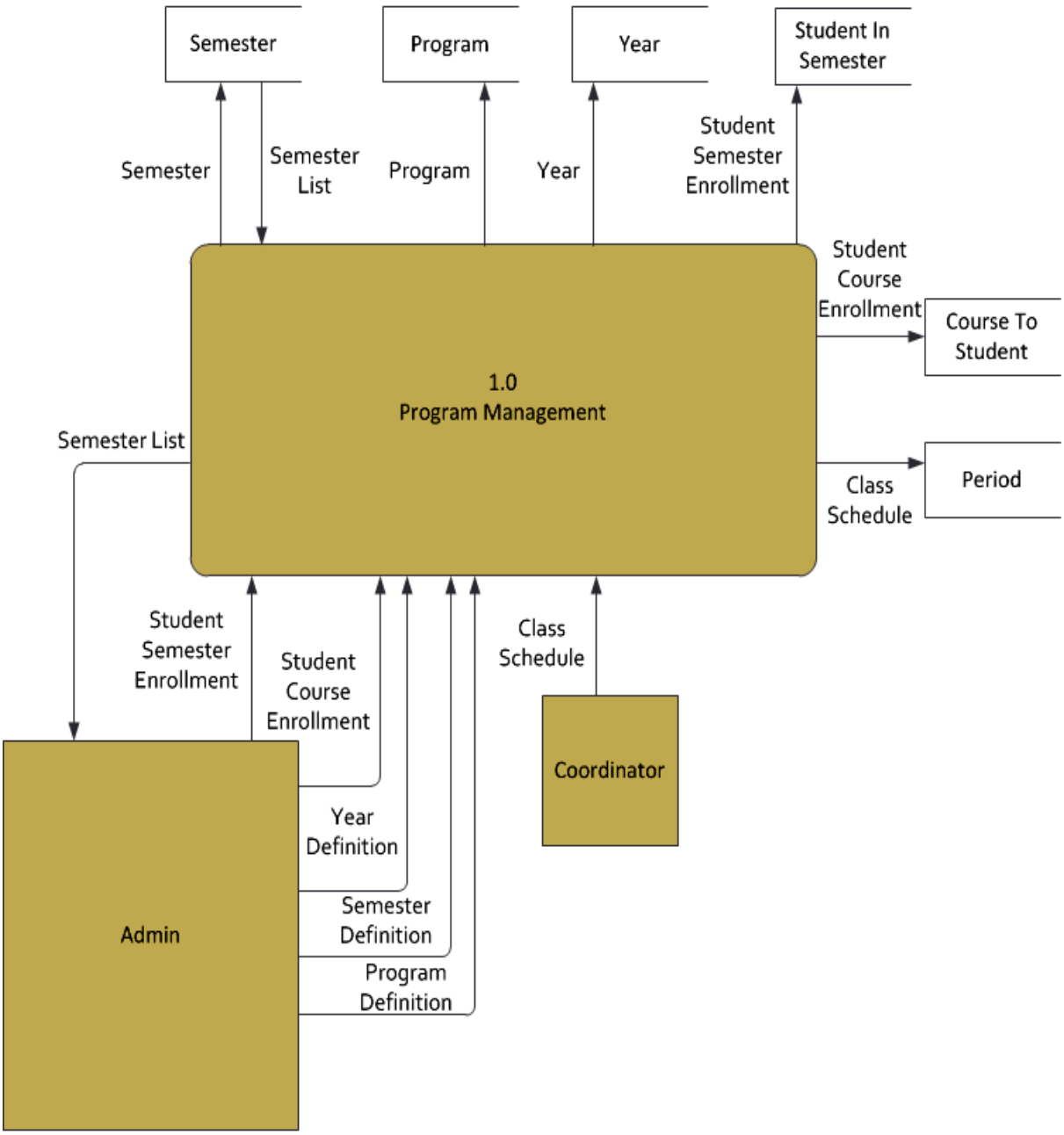


4.2 First Level DFD

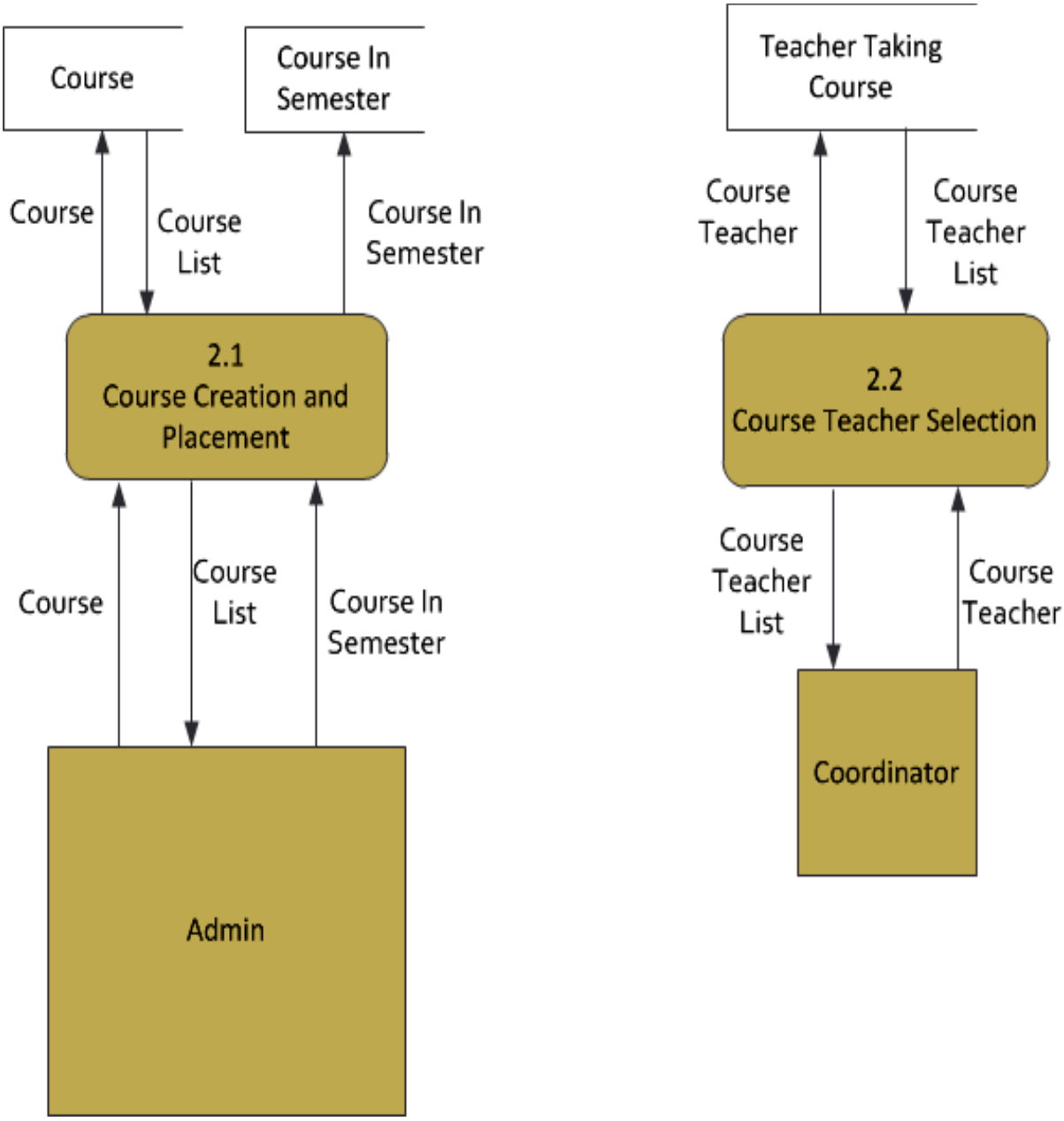


4.3 Second Level DFD

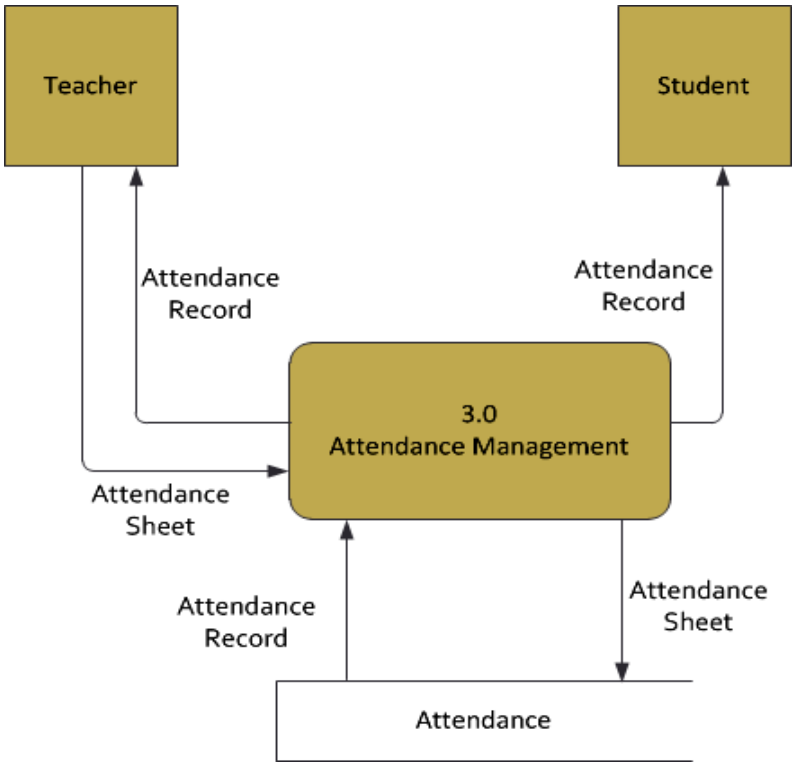
4.3.1 Program Management



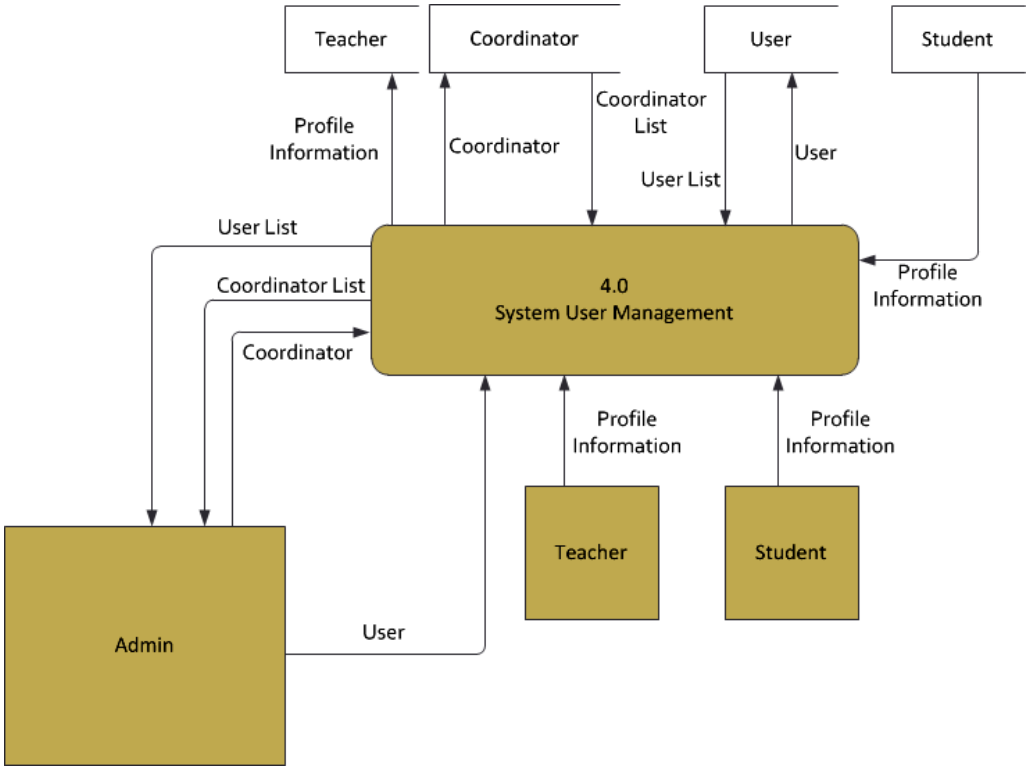
4.3.2 Course Management



4.3.3 Attendance Management



4.3.4 User Management



5. User Interface Design

In this section there is a concise physical view how the system is intended with the input and output. With the design the development team can find ease to validate the system with Graphical representation.

5.1 Input Design

The first figure is for how the courses are assigning to the respective students. Here there are two multiple list box to select the students and courses.

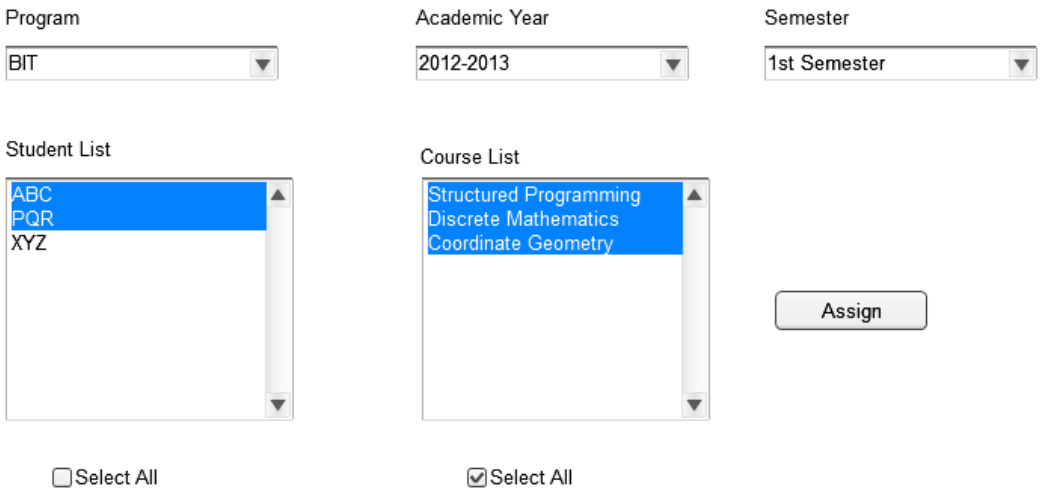


Fig: 5.1 Assign Course to Student

Next is enrollment of the students to respective semester.

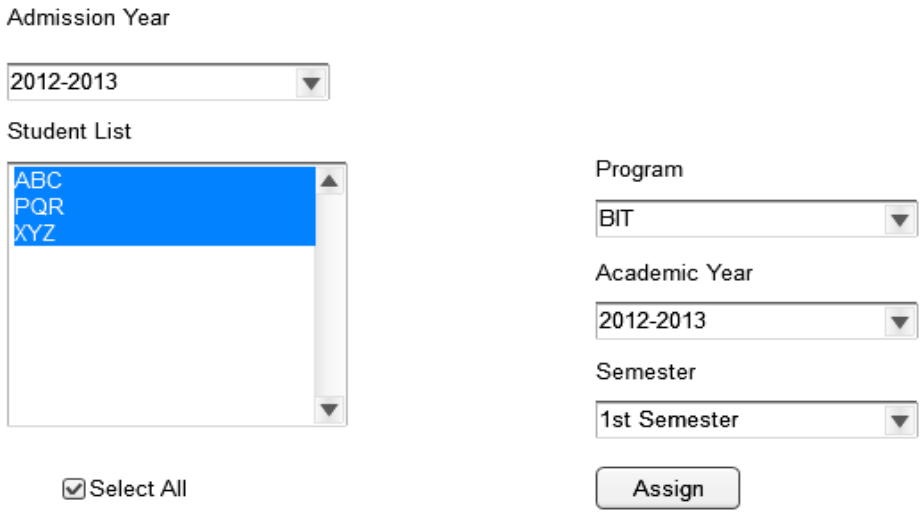


Fig: 5.2 Assign Students to Semester

To contrast, here is the assigning of the courses to a semester by which a bridge of semester, courses and student is made.

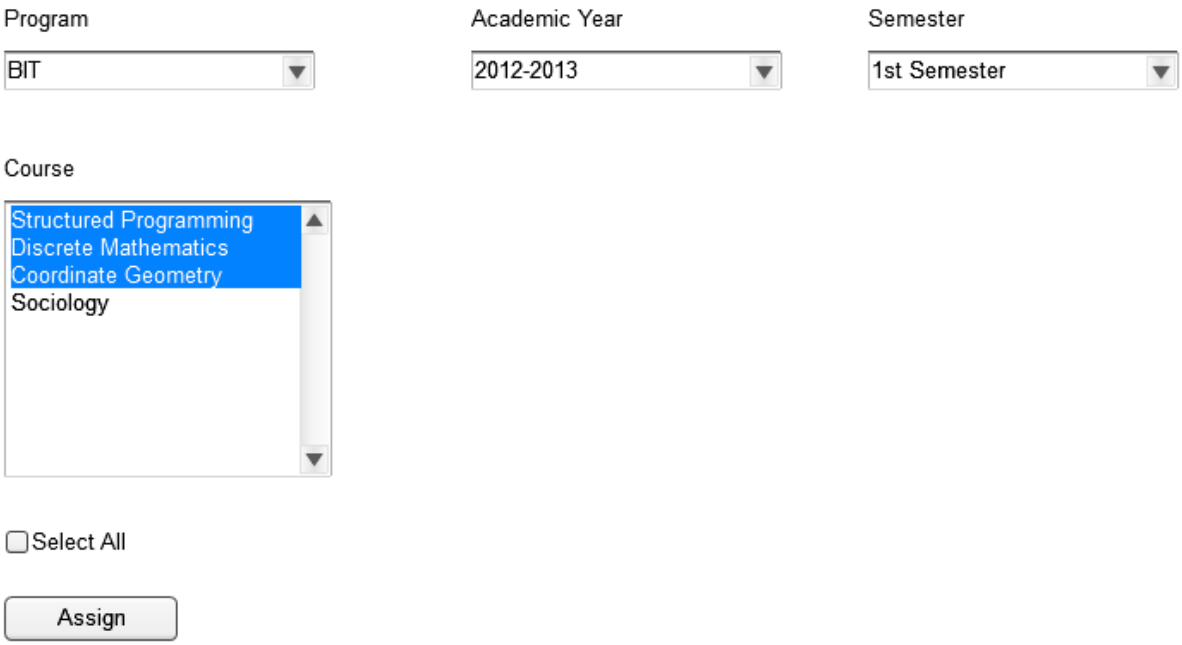


Fig: 5.3 Assign Course to Semester

Finally, the deserved goal of interface of the attendance sheet is given below:

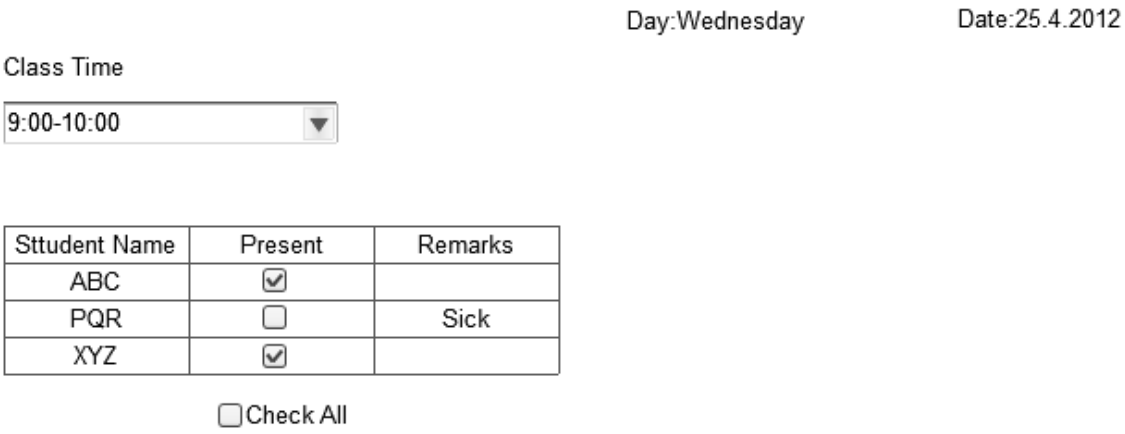


Fig: 5.4 Attendance Sheet

5.2 Output Design

Here there are some views of the attendance sheet to different users (students/teachers) from different queries.

First: Student

Course Name	Present Count	Absent Count	Percentage
C++	3	1	75
Java	5	0	100
PHP	4	2	66.67

Next Previous First Last

Fig: 5.5 Course wise Attendance sheet for Students

Second: Teacher

	25.4.2012	29.4.2012	30.4.2012	1.5.2012	2.5.2012
ABC	P	P	A	P	P
PQR	P	P	P	A	A
WXZ	P	P	P	P	P

Next Previous First Last

Fig: 5.6 Date wise Attendance sheet for Teachers

Third: Teacher

Student Name	Present Count	Absent Count	Percentage
ABC	4	1	80
PQR	3	2	60
WXZ	5	0	100

Next Previous First Last

Fig: 5.7 Student wise Attendance sheet for Teachers

6. Data Design

6.1 Entity Relationship Diagram

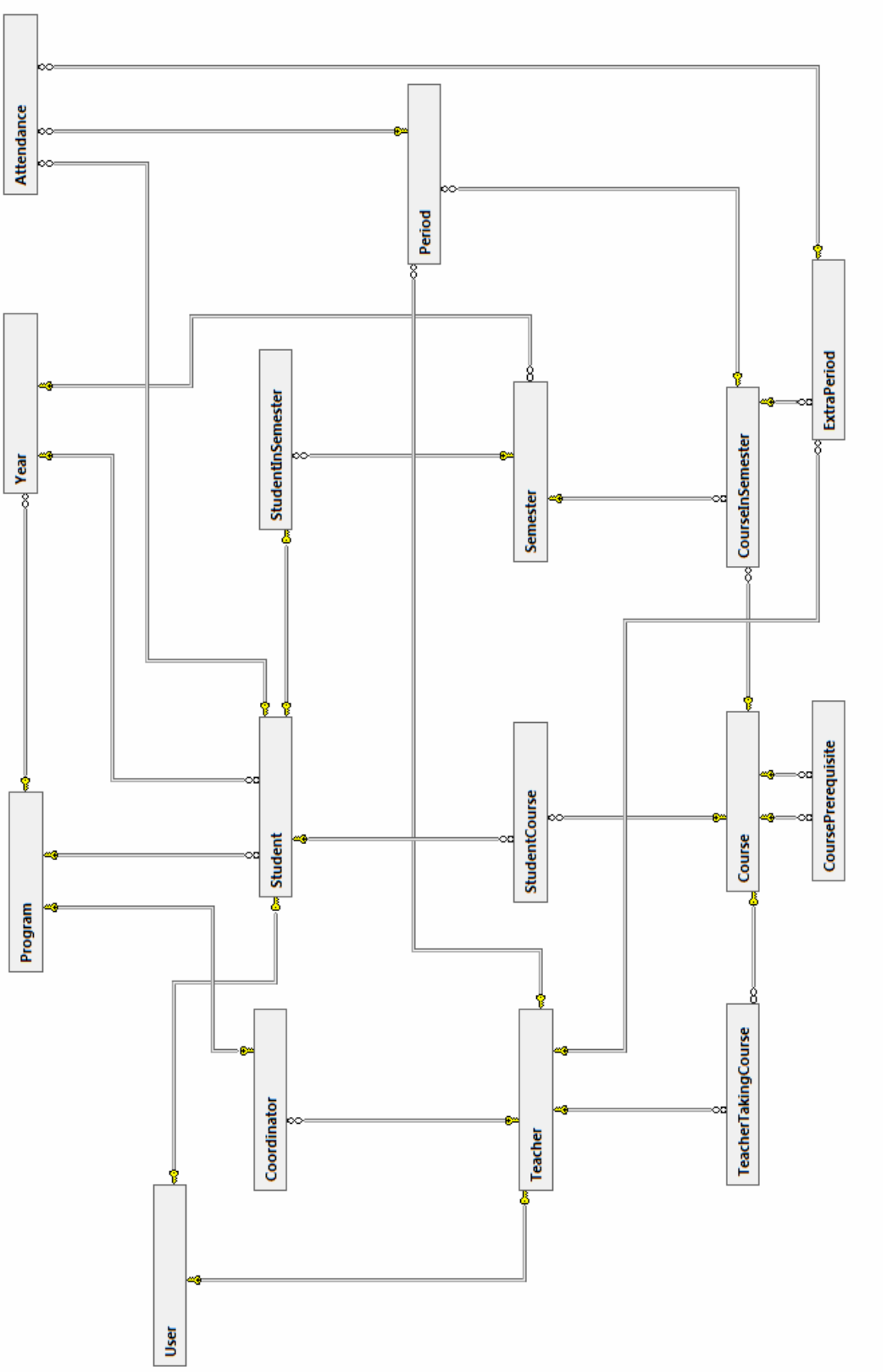


Fig: 6.1 Entity Relationship Diagram of SIS

6.2 Data Dictionary

Attendance

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
PK	ID	int	4	✗	1 - 1
FK	StudentID	int	4	✗	
FK	PeriodID	int	4	✓	
	AttendanceDate	date	3	✗	
	IsPresent	bit	1	✗	
FK	ExtraPeriodID	int	4	✓	

Indexes

	Name	Columns	Unique
PK	PK_Attendance	ID	✓

Foreign Keys

Name	Columns
FK_Attendance_ExtraPeriod	ExtraPeriodID->ExtraPeriod.ID
FK_Attendance_Period	PeriodID->Period.ID
FK_Attendance_Student	StudentID-> Student.ID

Coordinator

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
PK	ID	int	4	✗	1 - 1
FK IX	ProgramID	int	4	✗	
	TeacherID	int	4	✗	

Indexes

	Name	Columns	Unique
PK	PK_Coordinator	ID	✓
	IX_Coordinator	ProgramID	✓

Foreign Keys

Name	Columns
FK_Coordinator_Program	ProgramID->Program.ID
FK_Coordinator_Teacher	TeacherID->Teacher.ID

Course

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ID	int	4	✗	1 - 1	
	Name	varchar(150)	150	✗		
	Code	varchar(50)	50	✗		
	IsAllowedInMultipleSemester	bit	1	✗		((0))

Indexes

	Name	Columns	Unique
	PK_Course	ID	✓

CourseInSemester

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
	ID	int	4	✗	1 - 1
	SemesterID	int	4	✗	
	CourseID	int	4	✗	

Indexes

	Name	Columns	Unique
	PK_CourseInSemester	ID	✓
	IX_CourseInSemester	CourseID, SemesterID	✓

Foreign Keys

Name	Columns
FK_CourseInSemester_Course	CourseID->Course.ID
FK_CourseInSemester_Semester	SemesterID->Semester.ID

CoursePrerequisite

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
	ID	int	4	✗	1 - 1
	CourseID	int	4	✗	
	PrerequisiteCourseID	int	4	✗	

Indexes

	Name	Columns	Unique
	PK_CoursePrerequisite	ID	✓

Foreign Keys

Name	Columns
FK_CoursePrerequisite_Course	CourseID->Course.ID
FK_CoursePrerequisite_Course1	PrerequisiteCourseID->Course.ID

ExtraPeriod

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
	ID	int	4	✗	1 - 1
	Day	varchar(50)	50	✗	
	ClassTime	varchar(50)	50	✗	
	CourseInSemesterID	int	4	✗	
	TeacherID	int	4	✗	

Indexes

	Name	Columns	Unique
	PK_ExtraPeriod	ID	✓

Foreign Keys

Name	Columns
FK_ExtraPeriod_CourseInSemester	CourseInSemesterID->CourseInSemester.ID
FK_ExtraPeriod_Teacher	TeacherID->Teacher.ID

Period

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
	ID	int	4	✗	1 - 1
	Day	varchar(50)	50	✗	
	ClassTime	varchar(50)	50	✗	
	CourseInSemesterID	int	4	✗	
	TeacherID	int	4	✗	

Indexes

	Name	Columns	Unique
	PK_Period	ID	✓

Foreign Keys

Name	Columns
FK_Period_CourseInSemester	CourseInSemesterID->CourseInSemester.ID
FK_Period_Teacher	TeacherID->Teacher.ID

Program

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ID	int	4	✗	1 - 1	
	Name	varchar(50)	50	✗		
	IsAcademic	bit	1	✗		((1))

Indexes

	Name	Columns	Unique
	PK_Pogram	ID	✓

Semester

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
	ID	int	4	✗	1 - 1
	YearID	int	4	✗	
	Value	varchar(50)	50	✗	
	StartDate	date	3	✗	
	EndDate	date	3	✗	

Indexes

	Name	Columns	Unique
	PK_Semester	ID	✓

Foreign Keys

Name	Columns
FK_Semester_Year	YearID->Year.ID

Student

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ID	int	4	✗	1 - 1	
	ProgramId	int	4	✗		
	UserId	uniqueidentifier	16	✗		
	FirstName	varchar(50)	50	✗		
	LastName	varchar(50)	50	✗		
	MiddleName	varchar(50)	50	✓		
	RegistrationNumber	varchar(50)	50	✓		
	AttachedHall	varchar(50)	50	✗		
	AdmissionYearID	int	4	✗		
	IsApproved	bit	1	✗		((0))

Indexes

	Name	Columns	Unique
	PK_Student	ID	✓
	IX_Student	UserId	✓

Foreign Keys

Name	Columns
FK_Student_Year	AdmissionYearID->Year.ID
FK_Student_Pogram	ProgramId->Program.Id
FK_Student_User	UserId->User.Id

StudentCourse

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
	ID	int	4	✗	1 - 1
	StudentID	int	4	✗	
	CourseID	int	4	✗	

Indexes

	Name	Columns	Unique
	PK_StudentCourse	ID	✓

Foreign Keys

Name	Columns
FK_StudentCourse_Course	CourseID->Course.ID
FK_StudentCourse_Student	StudentID->Student.ID

StudentInSemester

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
	ID	int	4	✗	1 - 1
	StudentID	int	4	✗	
	SemesterID	int	4	✗	

Indexes

	Name	Columns	Unique
	PK_StudentInSemester	ID	✓
	IX_StudentInSemester	StudentID	✓

Foreign Keys

Name	Columns
FK_StudentInSemester_Semester	SemesterID->Semester.ID
FK_StudentInSemester_Student	StudentID->Student.ID

Teacher

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity	Default
	ID	int	4	✗	1 - 1	
	UserID	uniqueidentifier	16	✗		
	Designation	varchar(50)	50	✗		
	FirstName	varchar(50)	50	✗		
	LastName	varchar(50)	50	✗		
	MiddleName	varchar(50)	50	✓		
	IsApproved	bit	1	✗		((0))

Indexes

	Name	Columns	Unique
	PK_Teacher	ID	✓
	IX_Teacher	UserID	✓

Foreign Keys

Name	Columns
FK_Teacher_User	UserID->User.Id

TeacherTakingCourse

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
	ID	int	4	✗	1 - 1
	TeacherID	int	4	✗	
	CourseID	int	4	✗	

Indexes

	Name	Columns	Unique
	PK_TeacherTakingCourse	ID	✓
	IX_TeacherTakingCourse	ID	✓

Foreign Keys

Name	Columns
FK_TeacherTakingCourse_Course	CourseID->Course.ID
FK_TeacherTakingCourse_Teacher	TeacherID->Teacher.ID

User

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls
	Id	uniqueidentifier	16	✗
	Username	nvarchar(50)	100	✓
	Password	nvarchar(50)	100	✓
	Role	nvarchar(15)	30	✓
	Email	nvarchar(50)	100	✓
	PasswordSalt	nvarchar(50)	100	✓
	CreatedAt	datetime	8	✓
	LastModifiedAt	datetime	8	✓
	LastLoginAt	datetime	8	✓
	LastLoginIp	nvarchar(50)	100	✓
	IsActivated	bit	1	✓
	IsLockedOut	bit	1	✓
	LastLockedOutAt	datetime	8	✓
	NewPasswordKey	nvarchar(50)	100	✓
	NewPasswordRequestedAt	datetime	8	✓
	NewEmailKey	nvarchar(50)	100	✓

Indexes

	Name	Columns	Unique
	PK_User	Id	✓

Year

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
	ID	int	4	✗	1 - 1
	Value	varchar(50)	50	✗	
	ProgramID	int	4	✗	

Indexes

	Name	Columns	Unique
	PK_Year	ID	✓

Foreign Keys

Name	Columns
FK_Year_Pogram	ProgramID->Program.ID

8. Conclusion

The Student Information System which capable of storing resources such as students and staff of IIT and their relationship was implemented. It is easily to track the relations of students and courses they have taken, courses and teacher they are given by using the friendly interface of the system. The system also supports the Attendance Management system by which the administrative officer can easily identify the students are capable of giving the final exam and also easy way to give promotion. On the other hand, the teacher can get ease to find the percentage of attendance of the students of his/her course so that s/he can get some report also about the student. The system can work in local or distributed manner. It means that the system can be used not only with the local connection of IIT but also from remote access of the students.

The system can be easily extended by introducing new modules. An example of such, future work is evaluation questions module that can be used to evaluate teachers, and output the statistics of the evaluation. However, very soon this system is integrating with two other systems, one is Result Management and another one is Asset Management.

This system is supposed to run as a pilot project here in IIT for 5-6 months to get the user acceptance and more feasibility. After the feedback and requirements the system will go for fine tuning and hope within 3-4 months it will run smoothly in IIT while the students, teachers and staffs will be the prime beneficiaries of this project.