Student Information System – IIUI
(Student ERP)

Developed By:
Muhammad Imran
(2540-FBAS/BSSE/F14)

Waqar Kaleem Khan
(2433-FBAS/BSSE/F14)

Supervised By
Mr. Zulqarnain Hasmi

Department of Computer Science &
Software Engineering
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad

بسم الله الرحمن الرحيم

Department of Computer Science and Software Engineering International Islamic University Islamabad

Dated: _________

FINAL APPROVAL

It is certified that we have read the project report submitted by **Muhammad Imran 2540-FBAS/BSSE/F14 and Waqar Kaleem Khan 2433-FBAS/BSSE/F14** and it is our judgment that this project is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the BS degree in Software Engineering.

Committee

External Examiner

Dr. Ahsan Qureshi, Lecturer

Department of Computer Science and Software Engineering

Faculty of Basic and Applied Science

International Islamic University, Islamabad ______________________________

Internal Examiner

Dr. Ali Daud, Lecturer

Department of Computer Science and Software Engineering

Faculty of Basic and Applied Science

International Islamic University, Islamabad ______________________________

Dr. Ahsan Qureshi, Lecturer

Department of Computer Science and Software Engineering

Faculty of Basic and Applied Science

International Islamic University, Islamabad ______________________________

Supervisor

Mr. Zulqarnain Hashmi, Lecturer

Department of Computer Science and Software Engineering

Faculty of Basic and Applied Science

International Islamic University, Islamabad ______________________________

ABSTRACT

STUDENT ERP is a web base application designed for the IIUI student. There are two interfaces in this website one is for student and another for the admin. This web base application will allow student to do online registration of the courses and can take their fees challan online. The system also allow the student to add and drop of the course. There is also an admin role in this web application system. Admin will make a profile for the student after confirming the admission in the university and do auto course registration for the student of the first semester and generate the student card automatically from the user profile. This system also generate roll no slip for those students who have clear in attendances which can be controlled by admin and in the last results can be handled by admin and students can see their results from the student side.

ACKNOWLEDGEMET

We bestow all praises and appreciation to Almighty Allah, the most Merciful, Who gave us the understanding, courage and patience to complete this project.

We respect and thank **Mr. Zulqarnain Hashmi,** for providing us an opportunity to do the project work in IIUI and giving us all support and guidance which made us complete the project duly. We are extremely thankful to him for providing such a nice support and guidance, although he had busy schedule managing the corporate affairs.

DECLARATION

We as a result of this document proclaim that this system, except as otherwise indicated, neither as a whole nor as a part has been copied out from anything. It is further proclaim that we have developed this software entirely on the basis of our hardworking made under the sincere guidance of our teachers and supervisor.

No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

DEDICATION

We dedicate this project to a school boy Aitzaz Hassan Bangash Shaheed from Hangu District, Pakistan. Who died while preventing a suicide bomber from entering his school at Hangu, on 6 Jan 2014 and saved about 2,000 students lives.

DISSERTATION

A dissertation submitted to the

Department of Computer Science & Software Engineering,

International Islamic University Islamabad

As a partial fulfillment of requirements, for the award of the degree

BS in Software Engineering

PROJECT IN BRIEF

Project Title:	***Student ERP (Student Information System - IIUI)***
Objective:	Online Course Registration, Fee and Results
Undertaken By:	
Supervised By:	Mr Zulqarnain Hasmi Department of Computer Science and Software Engineering, International Islamic University, Islamabad.
Date Started:	8,Aug 2018
Date Completed:	15, Nov 2018
Technologies:	Tools used: • PHP(Laravel Framework) • PHP (Custom) • MySQL • HTML 5 • CSS 3 • Bootstrap 4 • JavaScript
System Used:	Dell (Intel Core I-3)
OS	WIN 10

Table of Contents

Chapter 1

INTRODUCTION

Introduction

Simply it’s an online student information system with some extra features. This system is a phase after admission system which is currently launched by International Islamic University, Islamabad. In this system admin can create new session for the student and then offer/add courses for the current session and after adding courses admin creates profiles for those students who are newly admitted to university or those students who are already in university. Thereupon registered students will add/drop and handle subjects from online portal, after course registration students print their fee slip from the online portal and paid their fees. On the other hand admin handle fees of those students who paid their fees. Admin handle attendance status at the end of classes, after attendance roll number slip will generate for the student in online portal. And in the end results will announce for the students by admin.

1.1. Project Motivation

The motivation for designing this application came because offering online course registration system to the students of IIUI that they are easily registered their courses, print their fees slip from online system easily and want to be convenient and accessible for the students. With online student information system, the students makes everything clear on their end. Everything is in writing, and there's no mix up.

1.2. Scope

Student Information System will be web based application whose main language of programming will be PHP (Laravel Framework).This web base application will support student profiles creation, session creation and offer courses for the students, simply a process after admission system for IIUI . Its main aim is to simplify and improve the efficiency of the student information system process for both admin and student. Students will also be able to add and drop those subjects which are offer for least semester than student’s semester and be able to have a visual confirmation that the subjects\course are selected correctly. The scope for a product customization option is an added service or feature that we provide allowing student to add/drop in their online course registration. It is important to come up with various courses options that help student to select his drop or fail subjects as he/she choose.

1.3. Existing Software’s

There is no existing system for online course registration, fee and results in IIUI. But there is online admission system which is the current existing system which is the existing system of student ERP system. In admission system students are registered first from the online portal, they give their roll number slip from the online system and so on. We add those students to our system which are got admissions in the university. And add new features to the system according to the policies and rules of the university.

1.4 Objectives

The proposed system aims to facilitate the admin to create a session with courses and fee schema and student to its confidence on the program office or other offices in the university, we are providing them each facility to save their time.

The main objectives of this student information system are:

Choose Your Own Subjects – This system will help students to joining auto and manual courses, the courses are allotted to the student by admin from admin side. So, the student will also pick exactly the courses which he/she wants to join in current semester. This will surely enhance the image of the Student ERP system and student satisfaction will be more.

Better Knowledge- This system will provide student all the details of his courses, fee status, his/her attendance status and results. This confirmation will help students to check their work on right time.

Know Start/End Time- This system will show the time by which the course registration, fee last date, 2nd installment last date etc. to the student. Student can only manage their course registration and fees payment in the given last date which is provided by admin from the admin side.

Reduce Paper Work- As most of the things will be performed online, it will reduce the usage of paper for the Student ERP. Such as course registration is totally online, results, roll number slip and fee.

Improves Efficiency- This system will make things easier for students that this process is totally different and easier than paper work.

1.5 Problem Statement

The problem statement of online student information system project is:

The problem of	Students are not able to do online course registration
Affects	Students, organization
the impact of which is	• Waste of time and cost. • Student dissatisfaction.
a successful solution would be	The web application will provide for students to do online course registration and fees and can add drop the courses online and can get their results online.

Table 1.5: Problem statement of online ordering System

1.6 Overview

General overview of system features is comprises of:

What is accessible in courses: through this feature student can know which subject is offer in current session, and also add/drop subject on behalf of current semester. (For example: If semester of the student is 3 then he/she can access to list all courses of semester 1 and 2 in the current session).

Limitation of courses: there is only 19 credit hours per semester can student add, if the credit hours is above than maximum limits, the system could response. Student can also add minimum subjects as he/she required.

Fees: after course registration fee will be automatically generate for the student, and student will print fee slip from online system. And will paid his fees before due date. After payment admin enters those student fees records whom paid their fees. And hereafter there is a message display in student side for confirmation of fee payment.

Results and Attendance: this feature will help admin to clear or prevent students according to classes. So if the student is clear, roll number slip will automatically show in their screen on behalf of courses. After this exams will be held and added from the admin side and showed in the student side.

Data base: contain record of students, courses, fees and admins to provide some extra package for our regular student, and bonus package for high performance administrators.

Update: admin can add/edit/delete different fields into the system. He can add/edit/delete new student profiles, their images, personal and other details. He can add/edit/delete courses, sessions, fee schema and much more in this system. It's the admin who add new students account into the system. Admin has rights to add/edit/delete courses, fee schema and session etc. for students.

Chapter 2

SYSTEM ANALYSIS

PART 1(PROBLEM ANALYSIS)

2 **Problem Analysis**

Problem analysis means the process of understanding real world problems, user’s needs and proposing efficient solutions to fulfill that needs and requirements. The aim of this process is to get better understanding of the problem being solved, before the development begins.

2.1 **Existing Systems**

Online student systems are increasing universities scope for the student’s efficiency and effectiveness (course registration, fees and results) are growing in profit as it reduces both admin and students involved. Online Student ERP systems also provide quick response to student courses are more visual through graphical interfaces. Courses can be update in every session by admins. Therefore students on the other hand cannot update drop or add new courses from the current session. It can be automatically assigned to them.

Above all it save time of both admin can offer all courses before due date of courses registration and place fee schema in no time with just few inputs (There is same fee schema for different programs in other universities such as NUST, COMSATS) etc.

In this era of technology where our university IIUI hasn’t any course registration and fee system, and there is lots of problems occur for our students and employees in program offices. Paper work can be reduced and fee process is less depend upon how u utilize technology to satisfy your student and admin (program office) if the student and admin is satisfied it’s your plus point the goal of project is to satisfy their customer providing more ease and facilities.

2.2 **Failures of Existing Systems**

Although these online student information systems are very effective for growing paper work and to manage rush, but these systems fails when it comes to both manual and auto course registration providing facility. As with increasing diversity and technology every student’s want to customize their semester courses particularly according to their choice and especially when it comes to credit hours, and current other universities student systems can’t manage this diversity even if we provide hundreds of courses and only auto allotted to students this can’t handle diversity and also not a better approach.

Similarly, in organizations where employees are being paid on hourly basis. If an employee continuously wasting his time on using social networking sites or his personal tasks in job timings, then it will lead towards loss for a company. For monitoring purpose some companies also hire supervisors that are allocated to monitor the activities of employees on computer systems. In this scenarios, there is a high risk that a supervisor can be bribed.

2.3 Project Idea

Student ERP is an online order placing application for course registration and fee payment. It provides the facility to its students to make their own courses according to his/her credit hours which max 19. It's a web application provide all courses in limited time and it calculates the fees after course registration according to the quantity of subjects you added**.**

2.4 Proposed System

Student ERP system (Student Information System) is an online course, fee and results system with a little more fun features.

Through Student ERP Application:

- Sign up by admin.
- Then admin create session with due dates for courses, fees and installment.
- After session creation, admin added courses program vise.
- Admin adds fee schema by session (which will later creates fee slip for students according to the fee schema).
- Admin creates profiles of those students who are newly got admissions in university and also those students who are already in university.(greater than 1st semester students).
- Automatically generated password will be sended to students email which can be change later by students.
- Students login and add drop courses before due date.
- After course registrtion fee slip is automatically generated for the student according to registered courses.
- After fee payment by student, admin enters those students fee who can paid their fees.
- After clearance of fees and in the end of classes, admin enter attendance of registerd students.
- If the student's attendance is clear, there is automatically roll number slip is generated for the student side.
- Before results admin create new session with due dates for courses, fees and installment.
- After session creation, admin added courses program vise.
- Admin adds fee schema by session.
- Above three lines are repeating because if results is added new session is automatically assigned to the students.

- In the last admin enters results of the students.

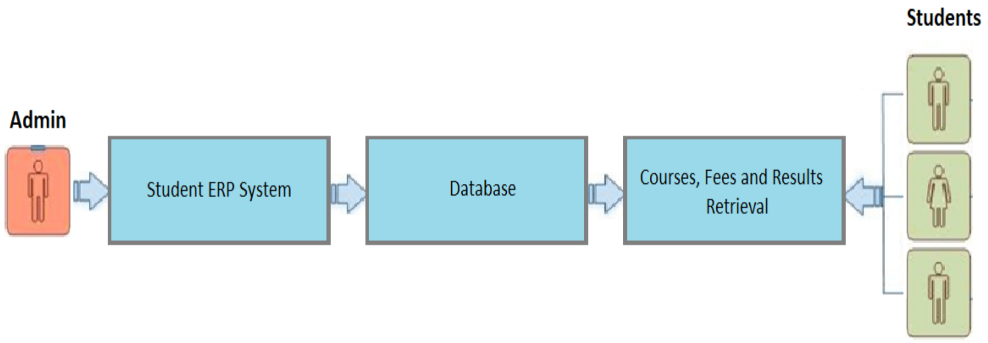


Figure 2.1: Graphical Model Student ERP Application

2.5 Project Scope

Student Information System will be web based application whose main language of programming will be PHP (Laravel Framework).This web base application will support student profiles creation, session creation and offer courses for the students, simply a process after admission system for IIUI . Its main aim is to simplify and improve the efficiency of the student information system process for both admin and student. Students will also be able to add and drop those subjects which are offer for least semester than student's semester and be able to have a visual confirmation that the subjects\course are selected correctly. The scope for a product customization option is an added service or feature that we provide allowing student to add/drop in their online course registration. It is important to come up with various courses options that help student to select his drop or fail subjects as he/she choose.

2.6 Software Process Scope

The process followed for this project is research and development. The reason behind choosing this process is that the features we introduced was totally new to work on, in field of online course registration. Agile technique is also followed to review the project requirements and accept the changes needed after each module is developed. The main goal is to ensure the production of high quality software that meets the user needs.

2.7 Operating Environment

It is necessary to consider the operating environment for the product been developed. By doing this the development team and the end users are able to deal with product in a good way. Some of the major requirements for operating this system “Student ERP System” are as following: internet **connection** to connect online. **Operating systems** Windows 7, Windows 8, Windows 8.1, android and IOS etc. A server with enough capability to deal with multiple customer requests**.** Hardware Specification are a **Processor** of 2.0 GHz, **RAM with** 2 GB and a **Hard** Disk: 50 GB

Frameworks used to carry out this project includes Visual Code studio, MySQL, Windows 7/8, notepad++, and, Photoshop, Star UML, Paint 3D, PHP through wamp and Laravel through composer.

2.8 Software Modules

Software modules are the division of project in order to distribute software requirements for increments. The need of these modules is to get all the requirements done on the basis of priority level. Following are the modules of the Student ERP System: On the basis of the nature of the project software modules are further divided into two parts

Client side module

Server side module

2**.8.1 Client side modules Web Application**

Client’s provided with web application through which student can interact with the server. Can easily perform any function or activity.

2.8.2 Selection of courses

For this purpose student is provided with a complete course schema according to their semester and program contains complete range of courses that can be update by the admin on adding or removing any course. (Student can only add or drop subjects on behalf of their program and semester during a limited time given from admin side).

2.8.3 Data base

A data base in web application is provided to make sure that customer have access to its data online.

2.8.4 Tracking

After course registration student can keep track of its fees submission status and the roll number.

If the student submit his/her fees on time the admin update the status of the student the he/she submit the fees and wait for the end of semester and then he/she can get there roll number slip for final exam online and can print from online portal and after finishing exam when admin update the result he she can get there transcript from online portal with also all previous results.

2.8.5 Server Side Module Web Application

Server control all the traffic of students request through a web application interface.

2.8.6 Response Client's Request

Administration is responsible for management of all student's requests. In this module admin validate/verified students request and then send response accordingly

2.8.7 Throw Various Fees Schema

In this module admin offer different fee schema offers for different programs**.**

2.8.8 Update Settings Option

If administration make any changes in students record or add student fees status or add results or attendance of student the status will be available for students.

2.8.9 Centralize Database

MySQL is used to maintain all records of the students, their fees status, courses register in current semester, attendance status, and result of the students.

2.8.10 Update settings option

Admin has given a role to add new students, change password of current user, add session, update/edit /delete any student record.

2.8.11 Server GUI

Server GUI is the main module that integrates all the previous modules. In this module front hand was developed to interact with the web application. It is user-friendly and kept simple for better understanding of the system.

2.9 Actor Goal List

Actor	Goal
Admin	• Sign up • Login to the system. • Add new session • Add courses by program for the session • Add fees schema for the session • Add students • Add and update student status • Monitor student's system. • View student's system details. • Remove students. • View previous records of any student.
Client	• Starts from logging into application • Update some fields in personal details • Add and drop courses • View fees status • Print roll number slip • View result • Print challan form • And have option to submit full fees or in instalment.

Table 2-10 Actor Goal List

2.10 Software Requirements Specification

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete **Student ERP** by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs while defining high-level product features. The detailed requirements of the **Student ERP** are provided in this document.

2.10.1 Purpose

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

2.10.2 Scope

The document covers the features of this project. It facilitate both students and university any type of university can use this web application by making little changes they need in customization.

- Add new students after the confirmation of admission.
- Manage the information of students.
- Manage the students fee information
- Manage students course registration information
- Manage students attendance information
- Manage student result information
- Editing add and updating of students
- Show records of students courses register is current semester, results, university fees.
- Shows the information and description of online orders and order status.

PART 2

SYSTEM ANALYSIS

2.11. System Analysis

System Analysis is a software Engineering task that bridges the gap between system level requirement engineering and the software design.

2.11.1. Specific Functional Requirements

Functional requirements of the system define the internal workings of the software. It also defines the capabilities and functions that a system must be able to perform successfully. The functional requirements of Student ERP are as follows:

Student:

- Should login to perform further actions.
- Should have availability of email and password issue by university.
- Change password.
- Can view courses register in current semester
- Can view and edit (some fields) in personal details.
- Can view fees status.
- Can view results of all previous semester.
- Add and drop courses
- Instalment of fees(in two section)
- Print fees slip
- Print transcript

Admin:

- Can login through authentic details.
- Add new Students.
- Change password of students.
- Remove students.
- Add fees Schema for new semester.
- Should verify and validate student details.
- Should response appropriately to every student request
- Can also keep track of every student.
- Can update or change student’s details.
- Create session for new semester.
- Update or Add student results.

- Update students fees details

2.11.2 Non-Functional Requirements

In systems engineering and **requirements** engineering, a **non-functional requirement** is a **requirement** that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with **functional requirements** that define specific behavior or functions. Some of the non-functional requirements include:

All of the application data is stored in a MySQL, and therefore MySQL Database must also be installed on the host computer. As with PHP (Laravel Framework), this software is freely available and can be installed and run under most operating systems.

The server hardware can be any computer capable of running both the web and database servers and handling the expected traffic. For a small scale University/college that is not expecting to see much web traffic, an average personal computer may be appropriate. Once the site starts generating more hits, though, it will likely be necessary to upgrade to a dedicated host to ensure proper performance. The exact cutoffs will need to be determined through a more thorough stress testing of the system.

Constraints

Hardware Limitations: The minimum hardware requirement for the system is 2 GB of Ram and a 500MB hard-disc drive.

Others: The application should be built using PHP (Laravel) and JavaScript inscribed in HTML, CSS, Bootstrap and it should, initially, be accessible through the visual code/sublime IDE and later published on a server

2.11.2.1 Performance Requirements

Reliability

Operations should run smoothly

There should be minimum Service outage.

Availability

Full time high availability of the server to students and database to the admin.

Data Security

Student’s information are just shared with admin system.

Data must be highly secured.

2.12 User Characteristics

Users (admin) belong to Information Technology (IT) or administration team.

Chapter 3

SYSTEM DESIGN

3. System Design

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a **system** to satisfy specified requirements. **Systems design** could be seen as the application of **systems** theory to product development.

3.1 Use Case Diagram

The use case model captures the requirements of a system. Use cases are a means of communicating with users and other stakeholders what the system is intended to do.

A use case diagram shows the interaction between the system and entities external to the system. These external entities are referred to as actors. Actors represent roles which may include human users, external hardware or other systems. A use case is a single unit of meaningful work. It provides a high-level view of behavior observable to someone or something outside the system. The notation for a use case is an ellipse.

Use case diagram of Student ERP System consist of 2 actors and 15 use cases. Each use case is linked with any of the 2 actors. It described the interaction between the system and actors. Boundary of the system "Student ERP" is clearly mentioned in the following figure 4.1.

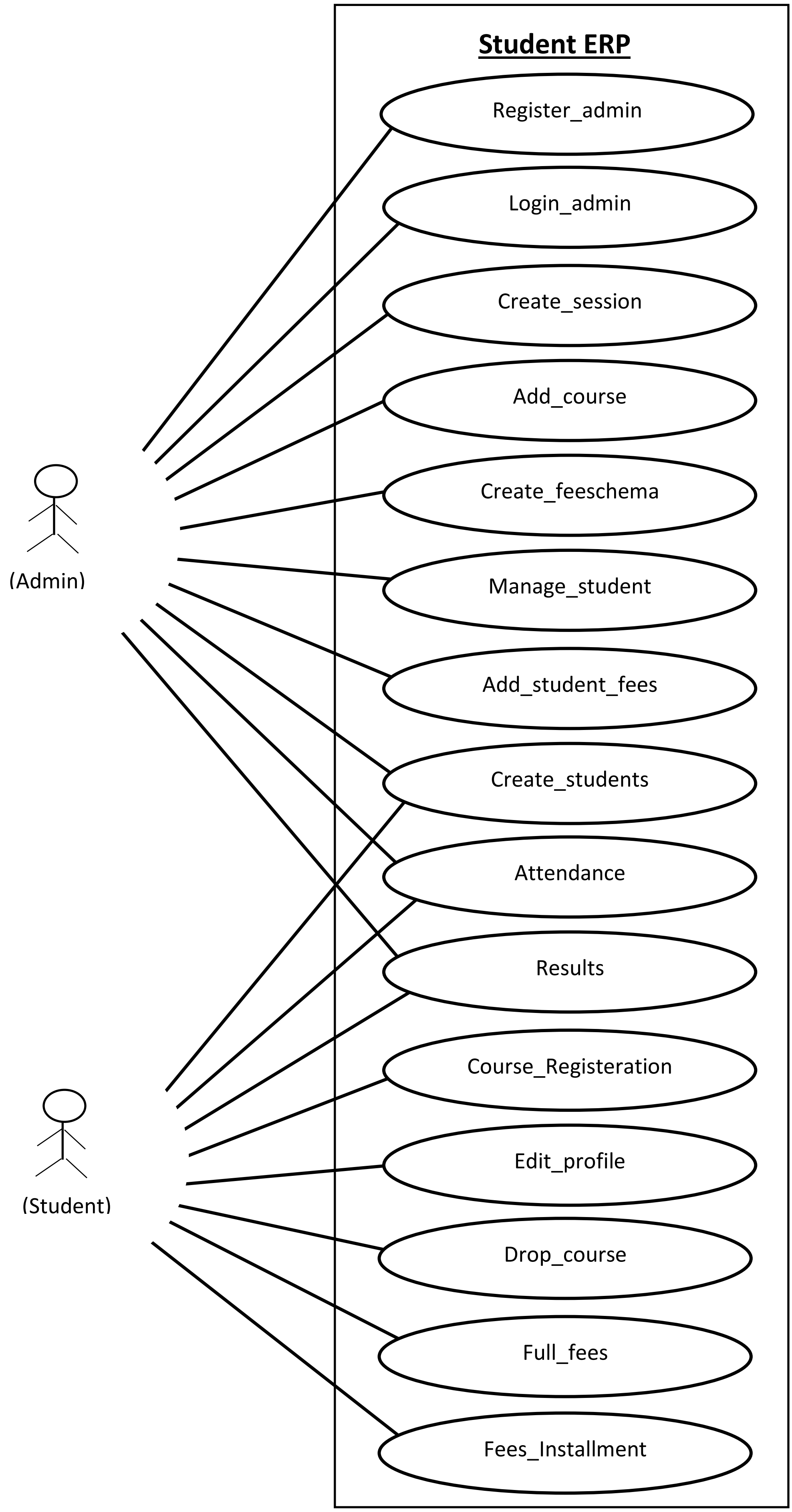


Figure4. 1 Use Case Diagram

3.1.1 Use Case 1:

UC-001 register_admin	
Scope	Student ERP (online course registration and fee system)
Primary actor	Admin
Functional scope	Create account for admin
Pre-condition	None
Post condition	Create new admin
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) Admin adds name, email password. 3) Admin enters random code	2) System sends 6 digits random code to given email by admin for verification. 4) System verify code and create admin
Failure scenario	
Actors Action And Intention	**System Responsibility**
Line 1: Unique email Line 3: Admin enter invalid code	System display error message for invalid code

3.1.2 Use Case 2:

UC-002 login_admin	
Scope	Student ERP (online course registration and fee system)
Primary actor	Admin
Functional scope	Login admin to the system
Pre-condition	Admin must registered with the system
Post condition	Access admin to the system
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) Admin enter email and password.	2) System verify password and email and access admin to the system
Failure scenario	
Actors Action And Intention	**System Responsibility**
Line 1: Admin enter invalid email or password	System display error message for invalid password and email

3.1.3 Use Case 3:

UC-003 create_session	
Scope	Student ERP (online course registration and fee system)
Primary actor	Admin
Functional scope	Creating session for the students
Pre-condition	Admin must login with the system
Post condition	New session create
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) Admin creates session with name, course registration and fee due date etc.	2) System give you message for session creation
Failure scenario	
Actors Action And Intention	**System Responsibility**
Line 1: Unique session	System display error message if session is already added before

3.1.4 Use Case 4:

UC-004 add_course	
Scope	Student ERP (online course registration and fee system)
Primary actor	Admin
Functional scope	Add new courses
Pre-condition	Session must be created first
Post condition	New courses are add to the system
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) Admin enter courses by session and program	2) System gives message for new course.
Failure scenario	
Actors Action And Intention	**System Responsibility**
Line 1: If admin enters same course for current session	System display error message for courses

3.1.5 Use Case 5:

UC-005 create_feeschema	
Scope	Student ERP (online course registration and fee system)
Primary actor	Admin
Functional scope	Creating fees schema for the students
Pre-condition	Session must be created first
Post condition	New fee schemas are add to the system
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) Admin enters fees schema.	2) System verify schema and responded to admin.
Failure scenario	
Actors Action And Intention	**System Responsibility**
Line 1: If admin enters same fee schema for program and current session (unique)	System display error message

3.1.6 Use Case 6:

UC-006 create_students	
Scope	Student ERP (online course registration and fee system)
Primary actors	Admin, Student
Functional scope	Creating students
Pre-condition	1) Session must be created first 2) Courses must added. 3) Fee schema must added.
Post condition	Students are registered to the system and can also login to the system
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) Admin adds new profiles (name,reg#,email etc.) 4) Student login to the system with auto generated password from his email.	2) System added students profile 3) System sends auto generated password to students email. 5) System login student to the student’s side.
Failure scenario	
Actors Action And Intention	**System Responsibility**
Line 1: Unique registration number, CNIC, phone, email and if date of birth is greater than 2000 and if CNIC is not 13 digits. Line 4: If	System display error message

3.1.7 Use Case 7

UC-007 course_registration	
Scope	Student ERP (online course registration and fees system)
Primary actor	Student
Functional scope	Add/Drop courses for current semester
Pre-condition	Student is admitted and authenticated
Post condition	new course is added or current course is drop as choice
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) student logged-in 3) Select courses from menu 4) Click add course button and add new course	2) Student is identified and authenticated 5) Save courses to student courses schema
Failure scenario	
Actors Action And Intention	**System Responsibility**
Line 3: Student select the course in which he/she already enrolled Line 4: Student credit hours more than 19 in current semester	System display error message

3.1.8 Use Case 8:

UC-008 edit_profile	
Scope	Student ERP (online course registration and fee system)
Primary actor	Student
Functional scope	Edit profile
Pre-condition	Student is admitted and authenticated
Post condition	Save edited details
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) Customize profile	2) Save student information
Failure scenario	
Actors Action And Intention	**System Responsibility**
Line 1: Student enter invalid email or password	System display error message

3.1.9 Use Case 9:

UC-009 drop_course	
Scope	Student ERP (online course registration and fees system)
Primary actor	Student
Functional scope	drop course
Pre-condition	Student is admitted and authenticated
Post condition	Course delete/drop
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) Select courses from menu 2) Click drop course button and confirm the course he is dropping	3) Remove course from student courses schema
Failure scenario	
Actors Action And Intention	**System Responsibility**
Line 4:Student credit hours less than 9 in current semester	System display error message

3.1.10 Use Case 10:

UC-010 full_fees	
Scope	Student ERP (online course registration and fees system)
Primary actor	Student
Functional scope	Full fees
Pre-condition	Student must registered with courses
Post condition	Print fee slip
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) Select fees from menu 3)Click drop full-fees button and print fees slip	2) Calculate total fees for all credit hours 4)Print fees slip
Failure scenario	
Actors Action And Intention	**System Responsibility**
Student enter invalid email or password	System display error message

3.1.11 Use Case 11:

UC-011 fees_instalment	
Scope	Student ERP (online course registration and fees system)
Primary actor	Student
Functional scope	Fees instalment
Pre-condition	Student must registered with courses
Post condition	Divide fees in two instalments
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) Click drop fees instalment button and print fees instalment latter 3)Print on installment button	2) Divide the total fees in two instalment and generate installment letter. 4)Print installment form
Failure scenario	
Actors Action And Intention	**System Responsibility**
None	

3.1.12 Use Case 12:

UC-012 manage_students	
Scope	Student ERP (online course registration and fees system)
Primary actor	Admin
Functional scope	Add, Update, Edit and Delete existing students record
Pre-condition	Student exist
Post condition	Delete student record
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
Admin login Select student Delete student record	Admin is identified and authenticated
Failure scenario	
Actors Action And Intention	**System Responsibility**
Admin enter invalid email or password	System display error message

3.1.13 Use Case 13:

UC-013 add_student_fees	
Scope	Student ERP (online course registration and fees system)
Primary actor	Admin
Functional scope	Update student fee status
Pre-condition	Student submitted fee
Post condition	Update student fee status
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) Select student fee from menu 2) Update student fee status 3) Save student fee status	4) System gives success message and updates student fees.
Failure scenario	
Actors Action And Intention	**System Responsibility**
Line 1: Admin enter invalid email or password	System display error message

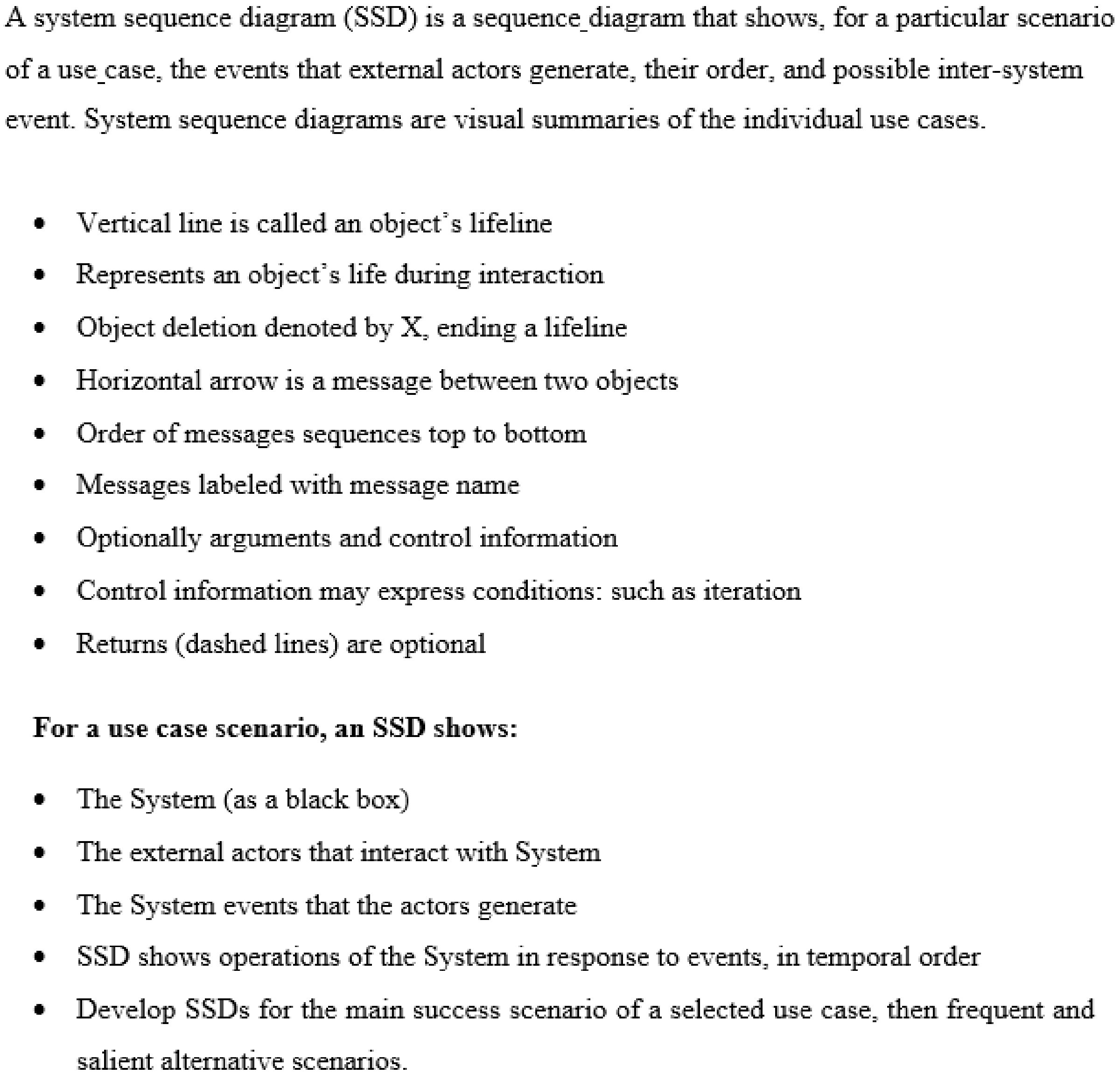
3.1.14 Use Case 14:

UC-014 attendance	
Scope	Student ERP (online course registration and fees system)
Primary actor	Admin, Student
Functional scope	Add student attendance
Pre-condition	Total attendance
Post condition	Save student attendance
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) Admin inserts students attendance 3) Student click prints roll number slip	2) Generate roll number slip for students 4) prints slip
Failure scenario	
Actors Action And Intention	**System Responsibility**
Line 1: (1) student is not registered with the system Line 1: (2) admin enters prevent	1) System display error message 2) Roll no slip not generate

3.1.15 Use Case 15:

UC-015 results	
Scope	Student ERP (online course registration and fees system)
Primary actor	Admin, Student
Functional scope	Add results
Pre-condition	Attendance clear
Post condition	Save students results
Main Success Scenario	
Actors Action And Intention	**System Responsibility**
1) Admin enters results 4) Students can read their results.	2) System save results 3) Show results to students.
Failure scenario	
Actors Action And Intention	**System Responsibility**
Line 1: unique results for per session	System display error message

3.2. **System Sequence Diagram**



A system sequence diagram (SSD) is a sequence diagram that shows, for a particular scenario of a use case, the events that external actors generate, their order, and possible inter-system event. System sequence diagrams are visual summaries of the individual use cases.

- Vertical line is called an object's lifeline
- Represents an object's life during interaction
- Object deletion denoted by X, ending a lifeline
- Horizontal arrow is a message between two objects
- Order of messages sequences top to bottom
- Messages labeled with message name
- Optionally arguments and control information
- Control information may express conditions: such as iteration
- Returns (dashed lines) are optional

For a use case scenario, an SSD shows:

- The System (as a black box)
- The external actors that interact with System
- The System events that the actors generate
- SSD shows operations of the System in response to events, in temporal order
- Develop SSDs for the main success scenario of a selected use case, then frequent and salient alternative scenarios.

3.2.1 Register Admin

To register the system:

1. Click on the “admin” link in welcome page and then click on the “register" link.
2. Enter “name” in name-field.
3. Enter “email” in email-field.
4. Enter “password” and “confirm password”.
5. Then click on the “submit” button.
6. System sends random code of 6 digits on given email.

7. Enter 6 digit random code on code-field.
8. Press “verify” button.

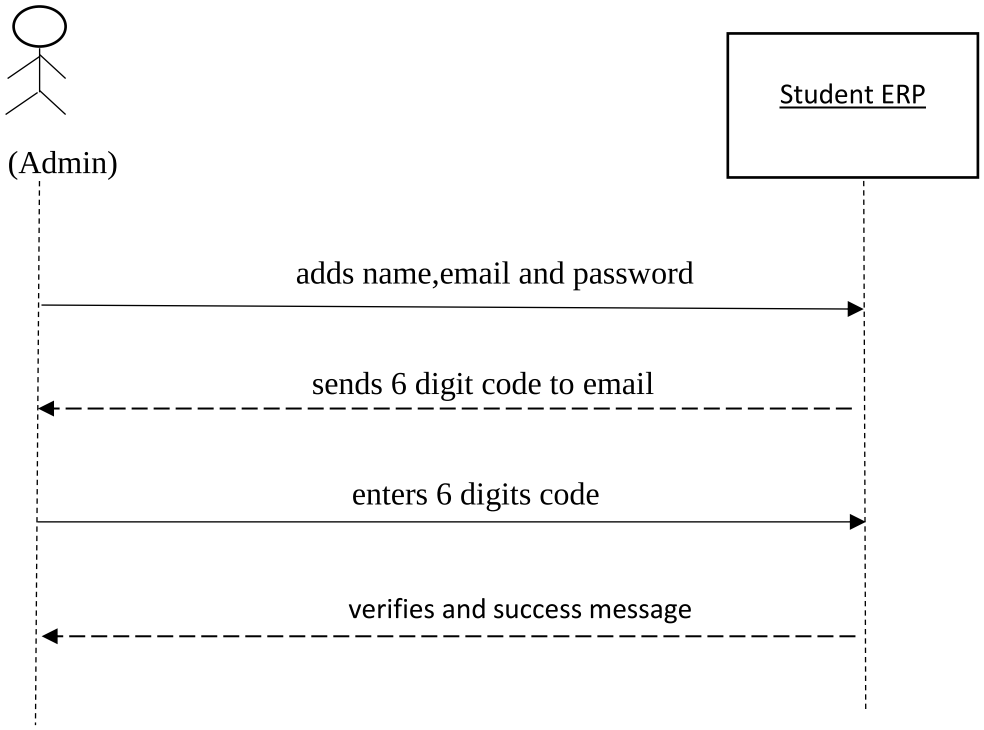

(Admin)
Student ERP
adds name,email and password
sends 6 digit code to email
enters 6 digits code
verifies and success message

Figure 4.2.1 Register Admin

3.2.2 Login Admin

To login admin to the system:

1. Click on "Admin” and then “Login" option in the welcome panel.

2. Enter registered email.

3. Enter registered password.

4. Then click on the ‘login’ button

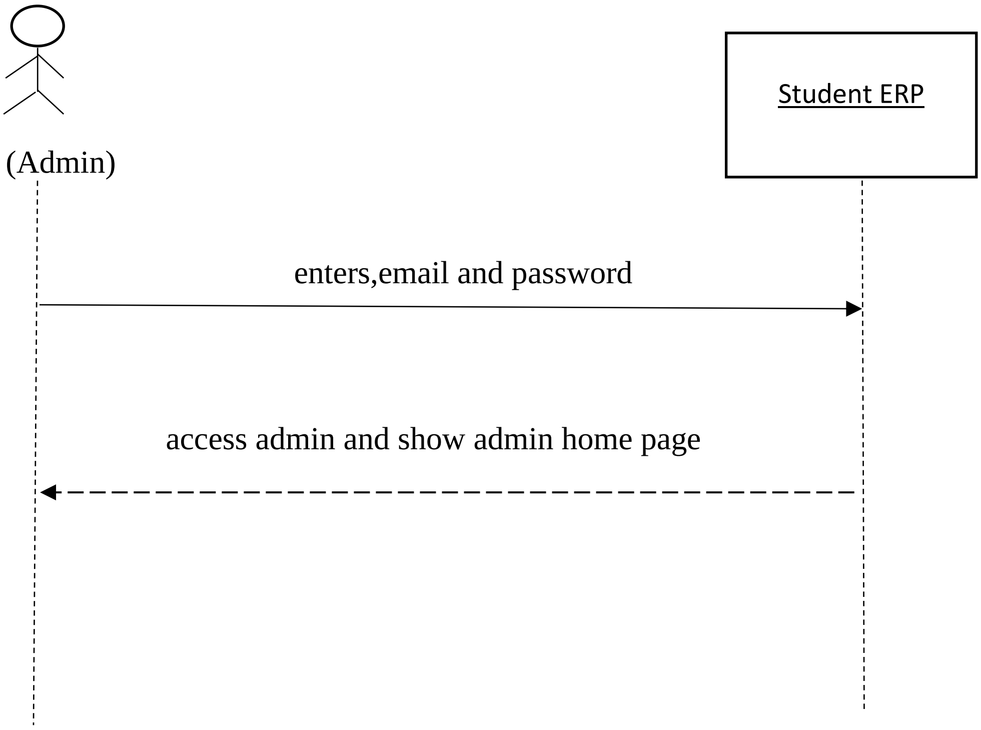

(Admin)
Student ERP
enters,email and password
access admin and show admin home page

Figure 4.2.2 Login Admin

3.2.3 **Create Session**

1. Click on “Session” and then click on below link create session.
2. Add session details (name, course and fee due date etc.).
3. Click on ‘Save’ button

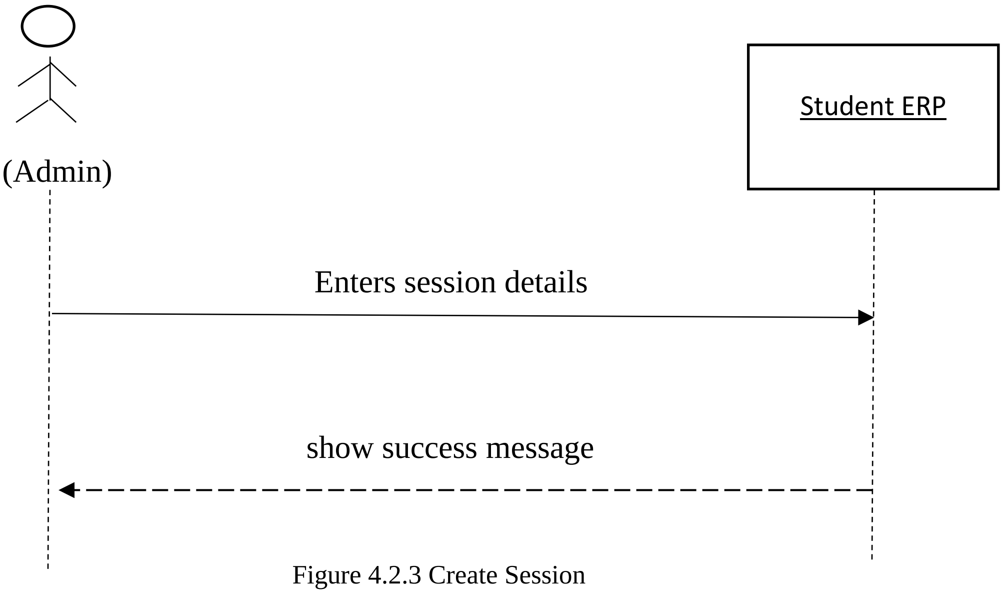


Figure 4.2.3 Create Session

3.2.4 Create Fee Schema

1. Click on Fee Schema link in sidebar and then click on add fee.
2. Add fee schema details. (fees per credit hours, registration fee, others etc.)
3. Click on save button.

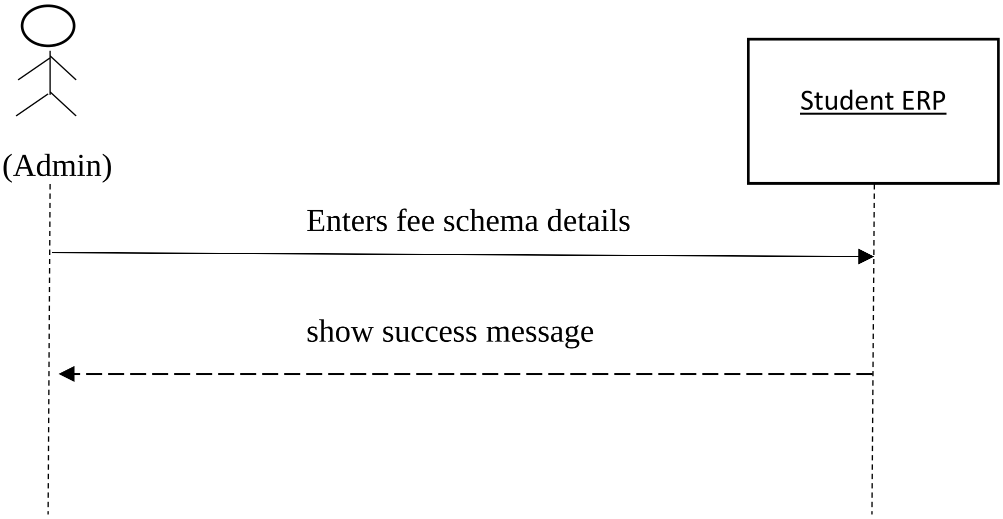


Figure 4.2.4 create fee schema

3.2.5 **Add Course**

It is necessary to add record of all courses.

1. Admin enter course name.
2. Admin enter course code.
3. Admin enter course credit hours.
4. Admin select session, program etc.
5. Then click on “SAVE”.

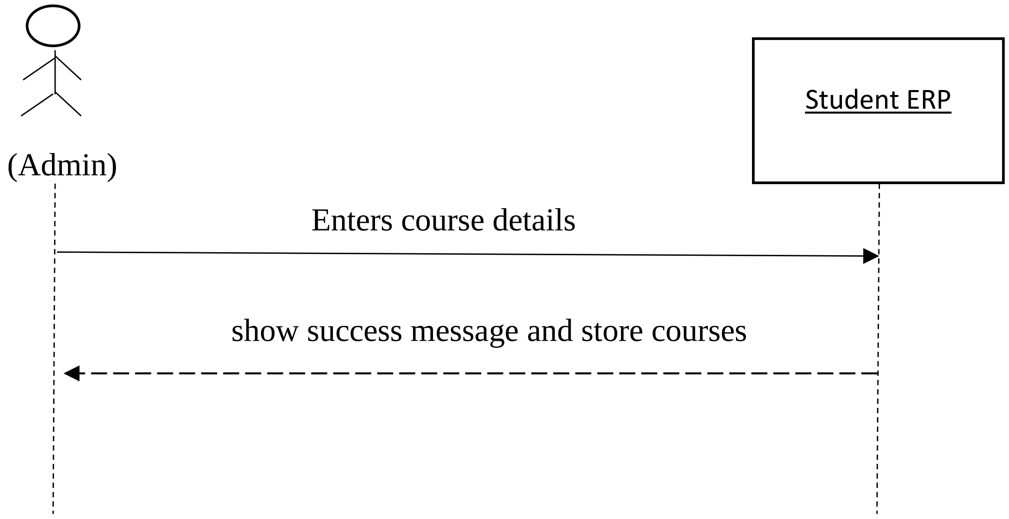


Figure 4.2.5 add course

3.2.6 Create Students

Student ERP needs to create students to connection between client and server.

1. Admin enters student’s personal details. (name, father name, CNIC, phone, address etc.)
2. Admin enters student’s academic details.
3. Admin fill the form and then click on “Save” button.
4. System sends auto generated password to student’s email.
5. Then student enter his/her email and password.
6. Student then click on login to access “Students home”.

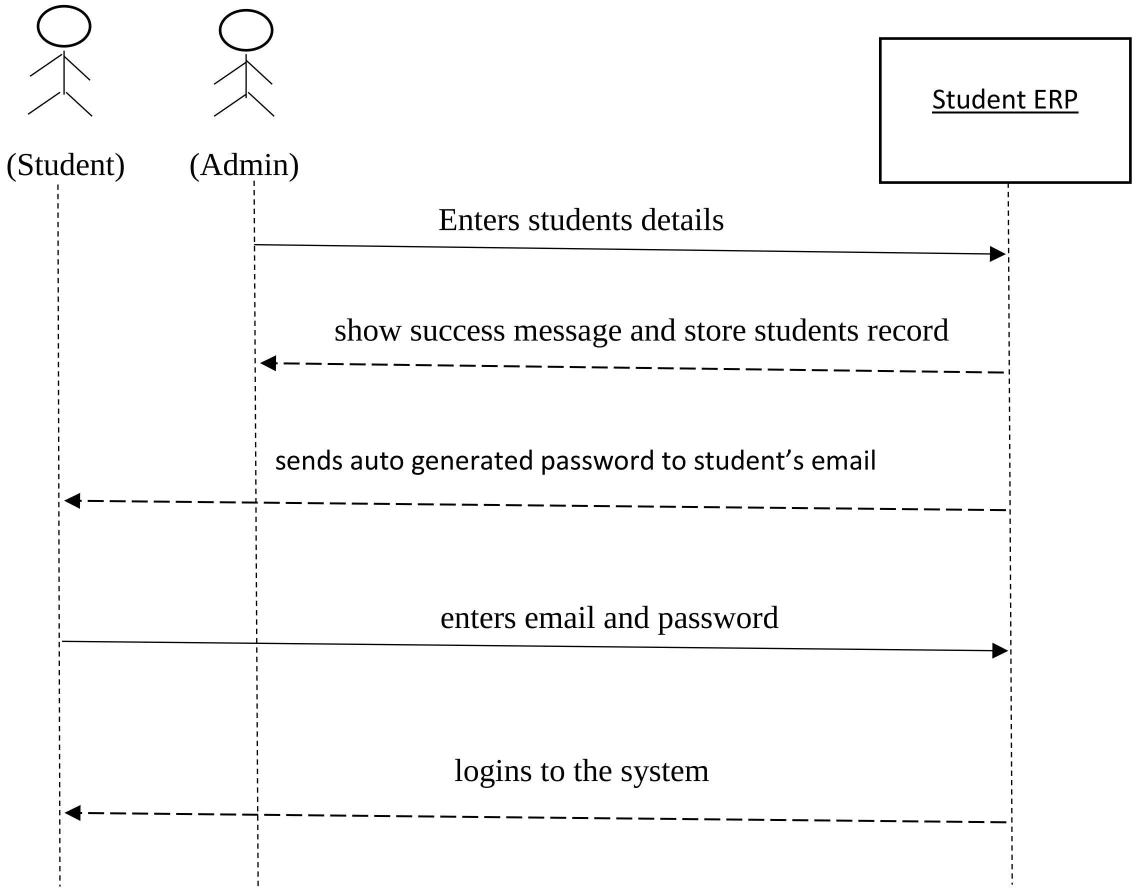

(Student)
(Admin)
Student ERP
Enters students details
show success message and store students record
sends auto generated password to student's email
enters email and password
logins to the system

Figure 4.2.6 Create Students

3.2.7 **Course Registration**

Course registration is editable for students before course registration due date.

1. Student login to the system
2. Student can add courses
3. Student can drop courses

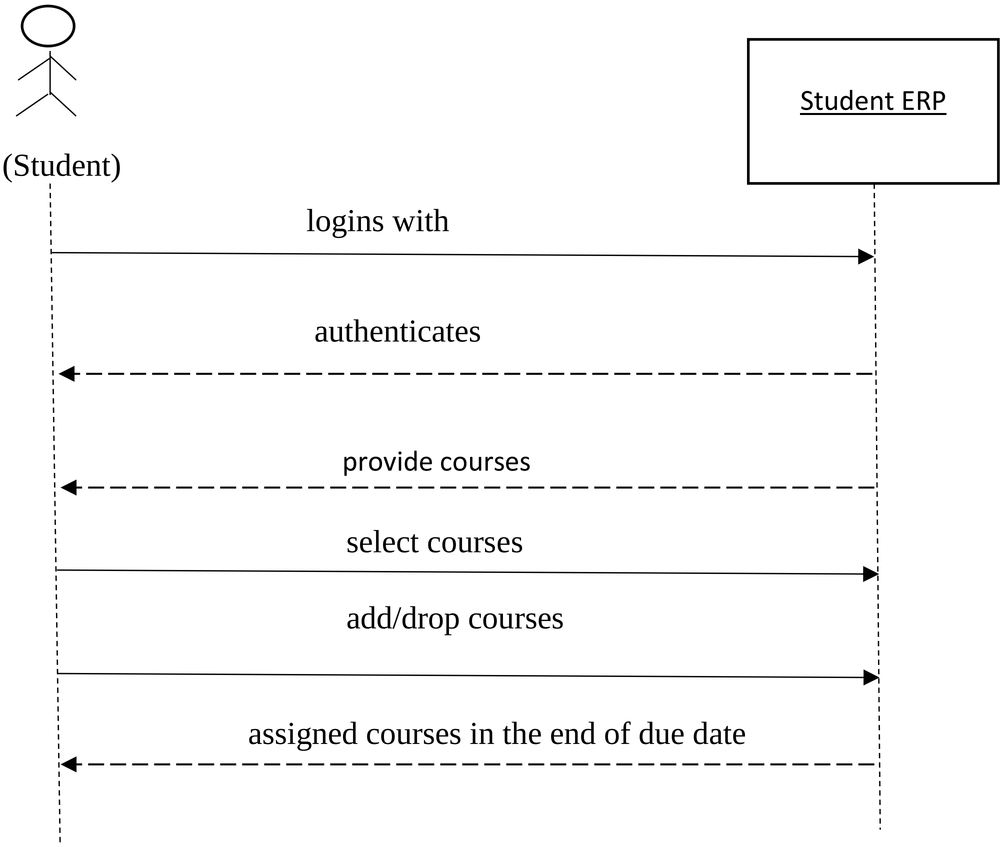


Figure 4-7 Course Registration

3.2.8 Edit Profile

1. Login Student
2. Click on edit profile on home page
3. Student can only edit specific data.
4. Click on update button

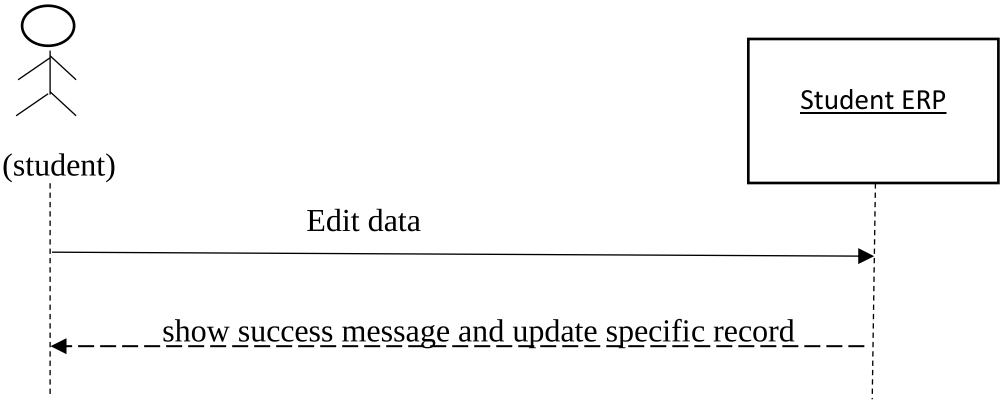


Figure 4-8 Edit Profil

3.2.9 Drop Course

1. Login Student
2. Select courses from menu
3. Click on drop button

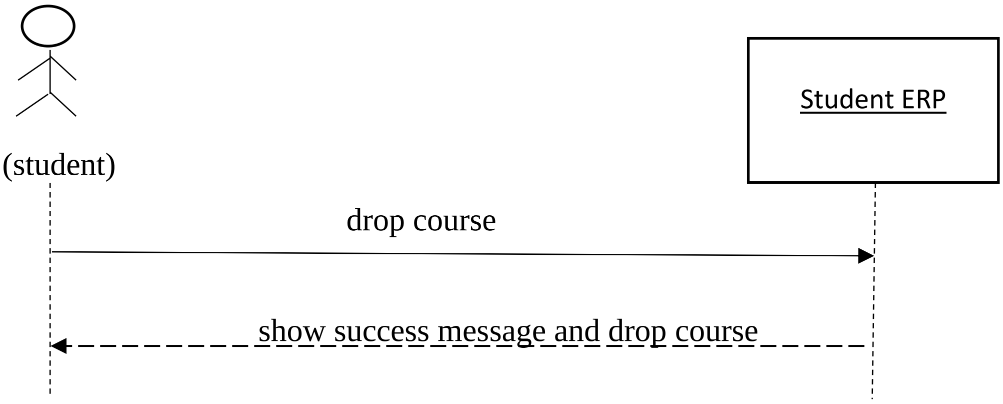


Figure 4-9 Drop Course

3.2.10 Full Fees

1. Login Student
2. Select fees
3. Generate fee after calculating credit hours
4. Print fee slip

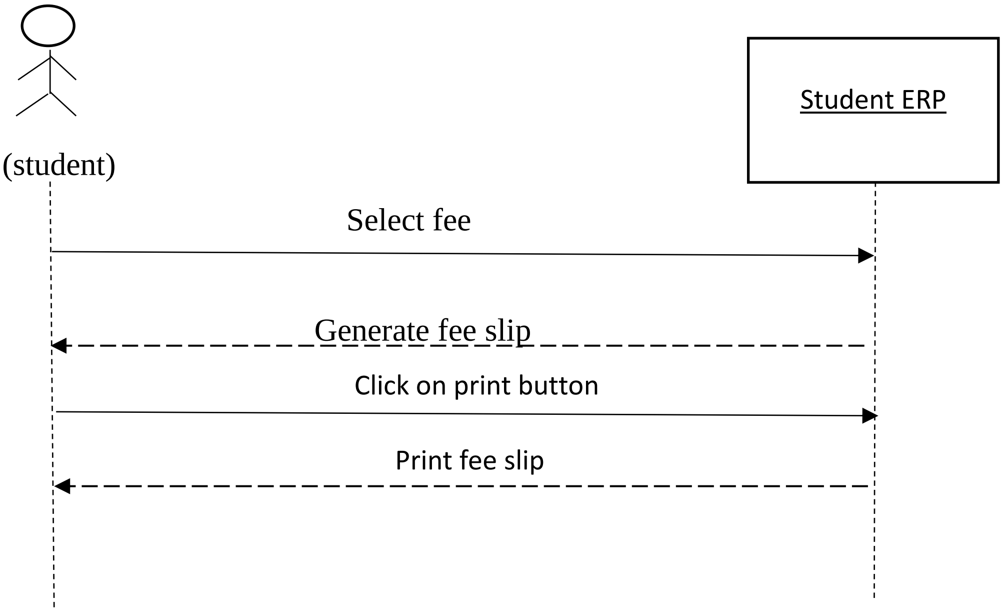


Figure 4-10 Full Fees

3.2.11 Fees Instalment

1. Login Student
2. Click on installment button
3. Generate installment fee
4. Click on print button under installment form

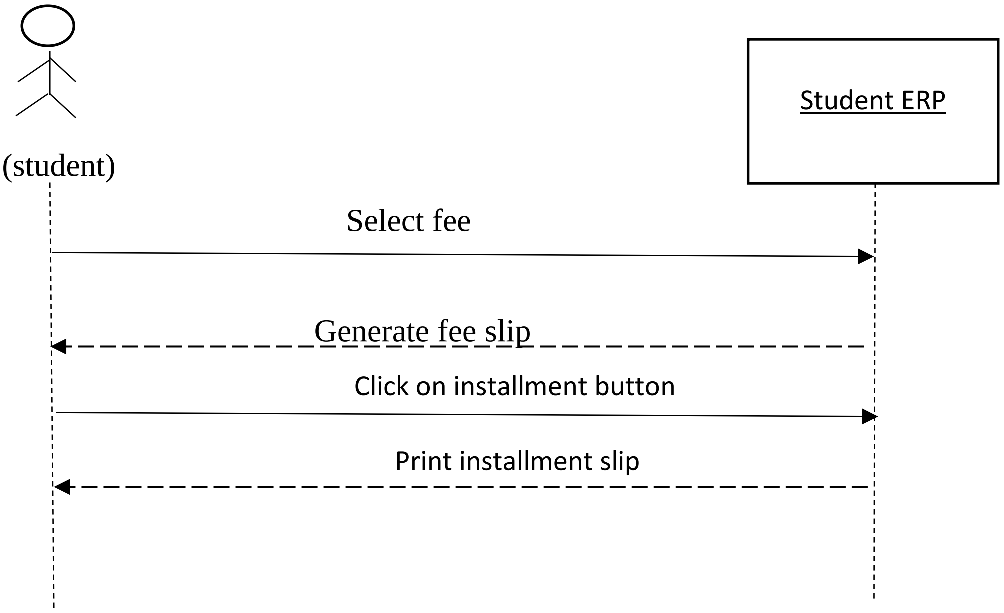


Figure 4-11 Fee Installment

3.2.12 Manage Student

1. Login Admin
2. Select student from the list
3. Click on delete button

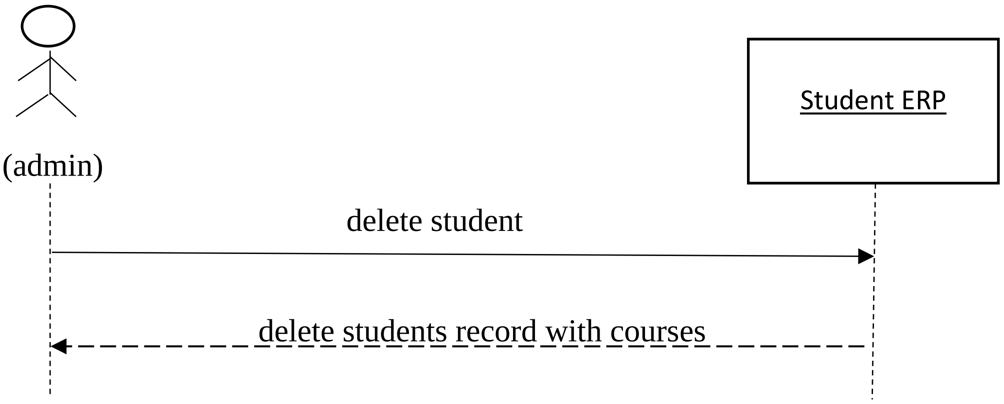


Figure 4-12 Manage Student

3.2.13 Add/Remove/Update Student Fee

1. Login Admin
2. Select student from the list
3. Click on edit button

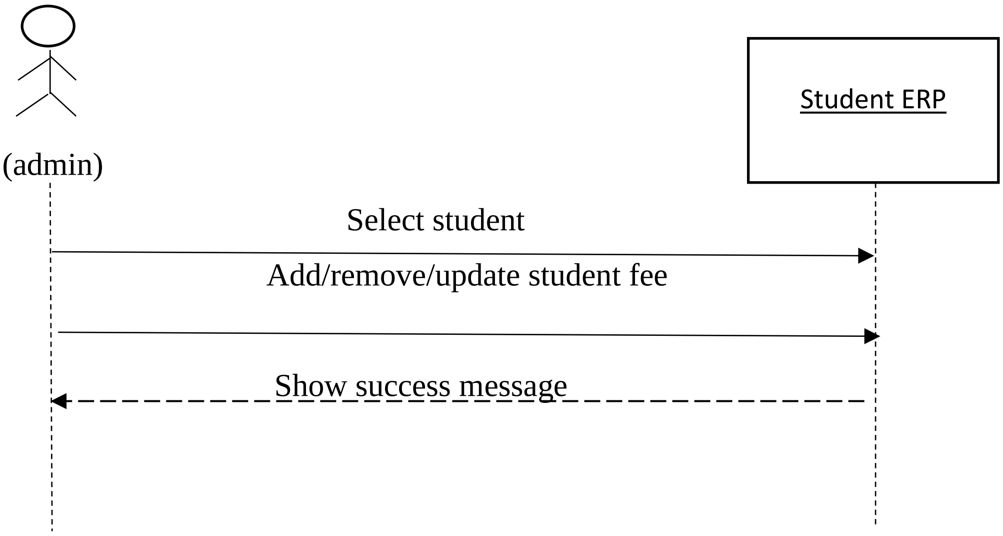


Figure 4-13 Remove Student Fee

3.2.14 Attendance

1. Login Admin
2. Select student from the list
3. Click on attendance
4. Add attendance
5. Login student
6. Click on print button of roll number slip

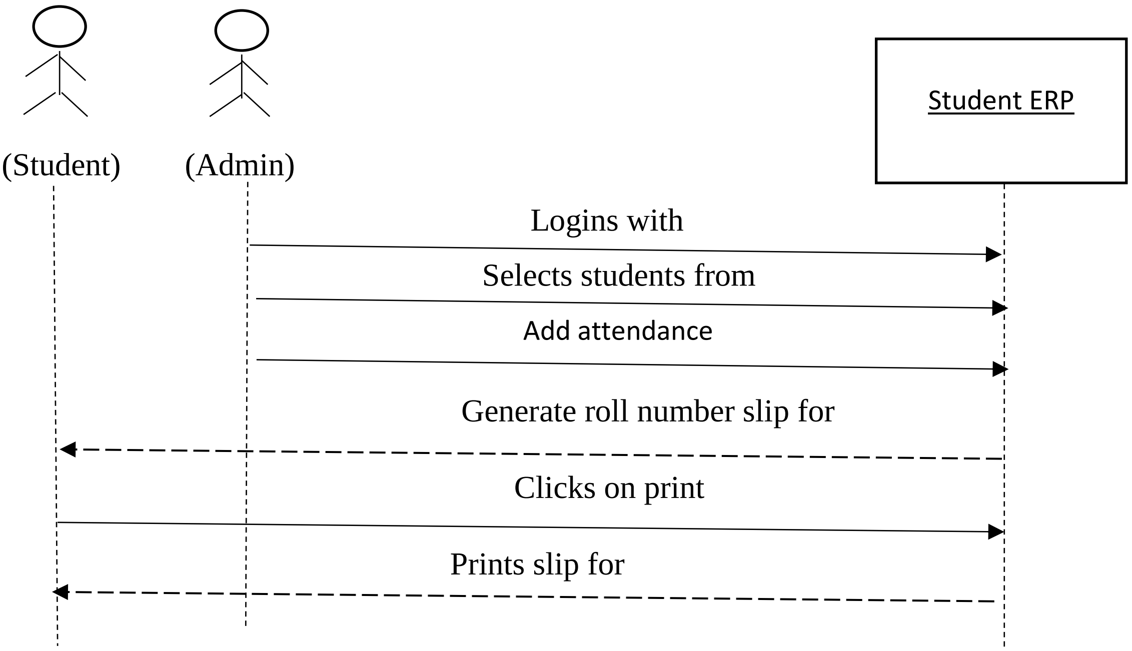


Figure 4.2.14 Attendance

3.2.15 Result

1. Login Admin
2. Select student from the list
3. Add result
4. Login student
5. Click on print button of roll number slip

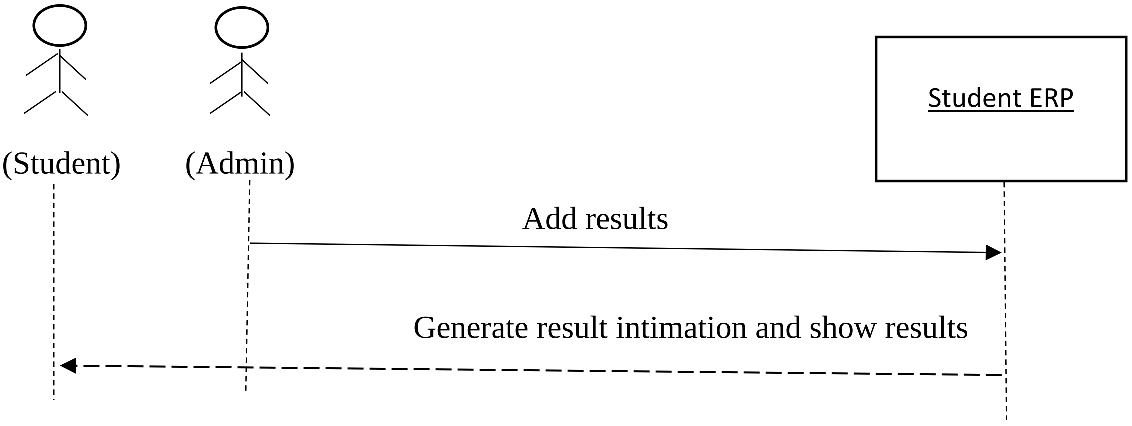
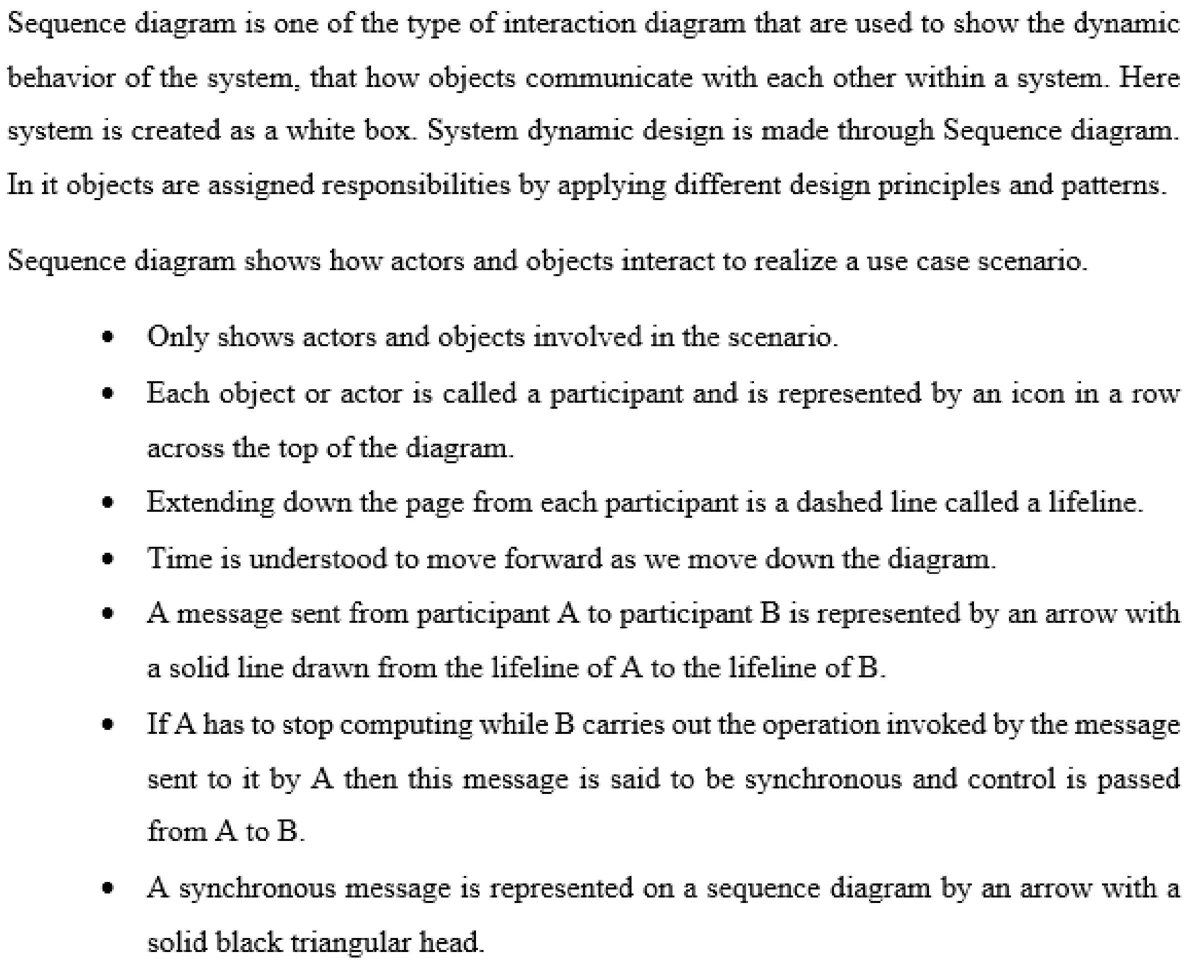


Figure 4.2.15 Result

3.3 Sequence Diagram



Sequence diagram is one of the type of interaction diagram that are used to show the dynamic behavior of the system, that how objects communicate with each other within a system. Here system is created as a white box. System dynamic design is made through Sequence diagram. In it objects are assigned responsibilities by applying different design principles and patterns.

Sequence diagram shows how actors and objects interact to realize a use case scenario.

- Only shows actors and objects involved in the scenario.
- Each object or actor is called a participant and is represented by an icon in a row across the top of the diagram.
- Extending down the page from each participant is a dashed line called a lifeline.
- Time is understood to move forward as we move down the diagram.
- A message sent from participant A to participant B is represented by an arrow with a solid line drawn from the lifeline of A to the lifeline of B.
- If A has to stop computing while B carries out the operation invoked by the message sent to it by A then this message is said to be synchronous and control is passed from A to B.
- A synchronous message is represented on a sequence diagram by an arrow with a solid black triangular head.

3.3.1 Admin Login

To login the system:

1. Click on the "Admin Login" button.
2. Enter “email” in field.
3. Enter the “Password” in field.
4. Then click on the “Login" button.

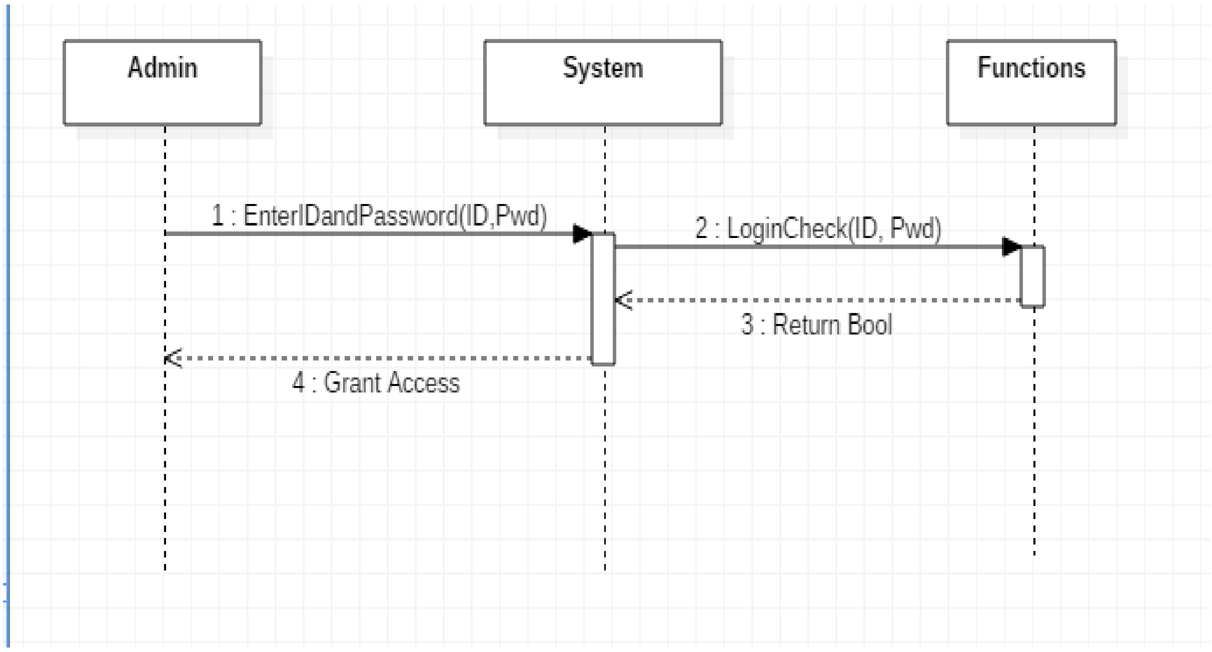


Figure 4.3.1 Admin Login

3.3.2 Student Login

To login the system:

1. Click on the “Login" button.
2. Enter “Email” in field.
3. Enter the “Password” of the client in field.

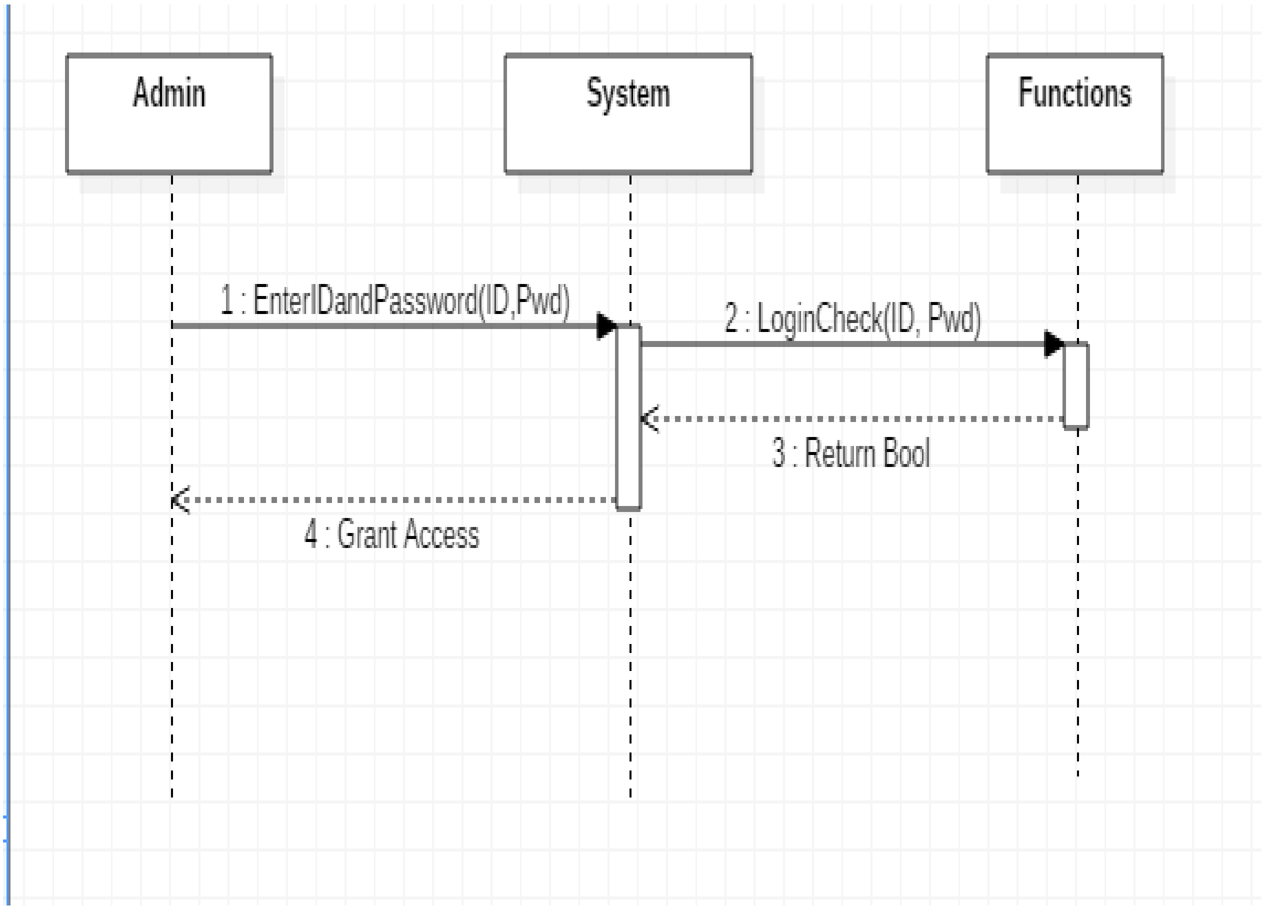


Figure 4.3.2 Student Login

3.3.3 Admin Logout

Admin Logout refrain all accesses from admin.

1. Click “Logout” button
2. Display home page

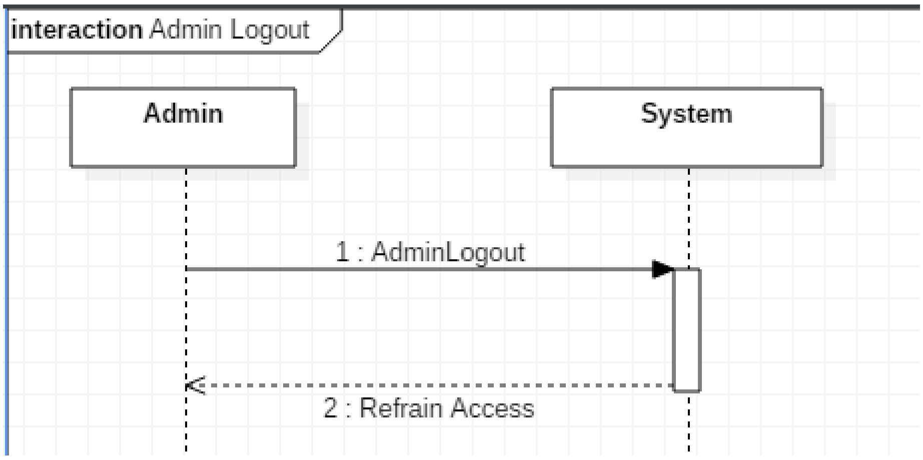


Figure 4.3.5Admin Logout

3.3.4 Build Connection

Desktop Activity Monitor System needs to build connection between client and server, to communicate.

1. Start Server.
2. Accept request
3. Ready to accept more request

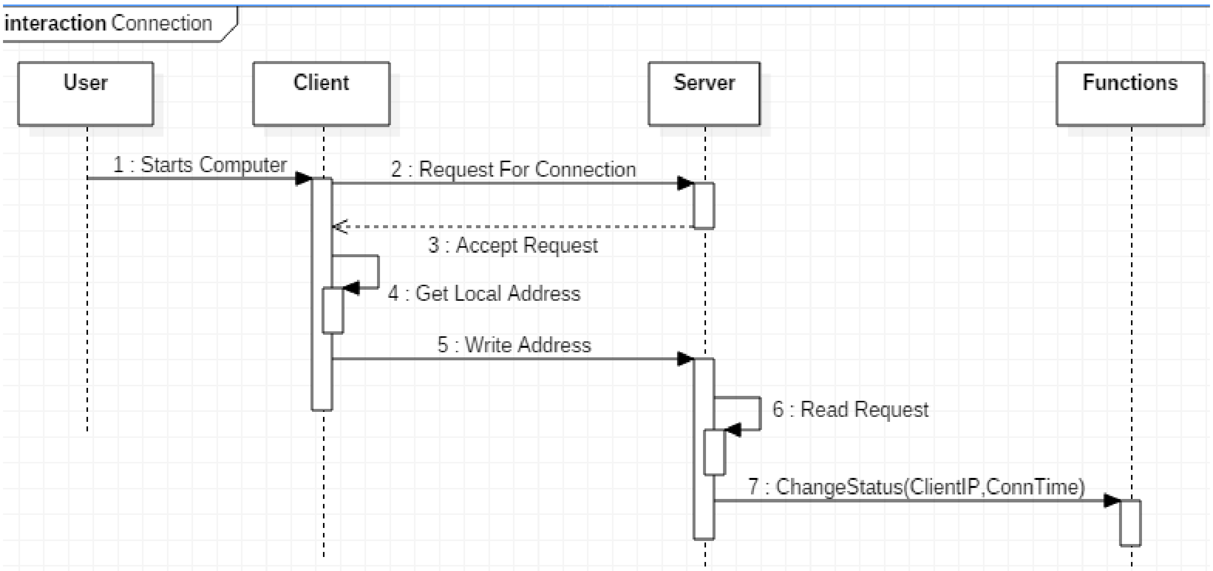
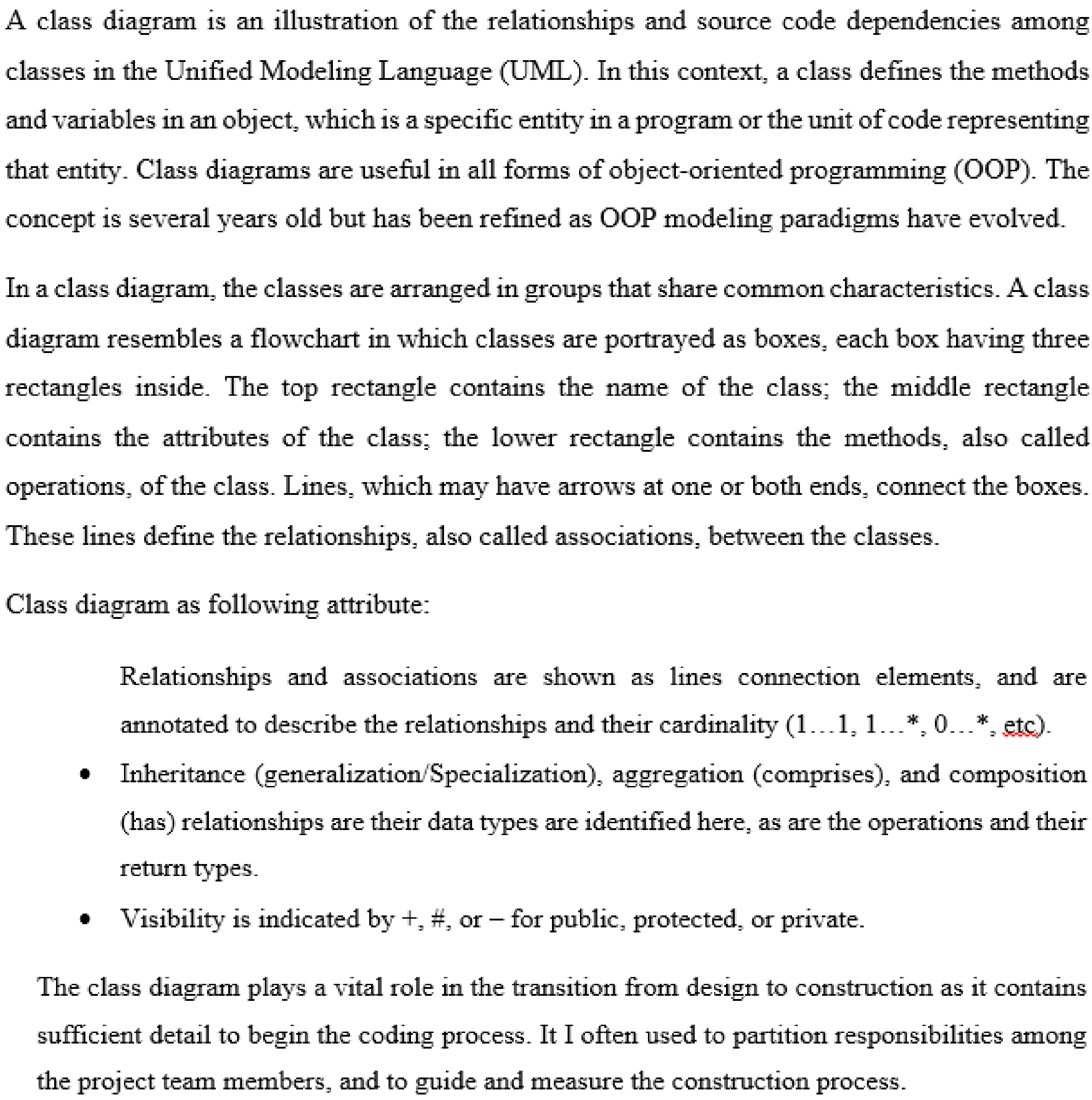


Figure 4.3.6Build Connection

3.4 Class Diagram



A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved.

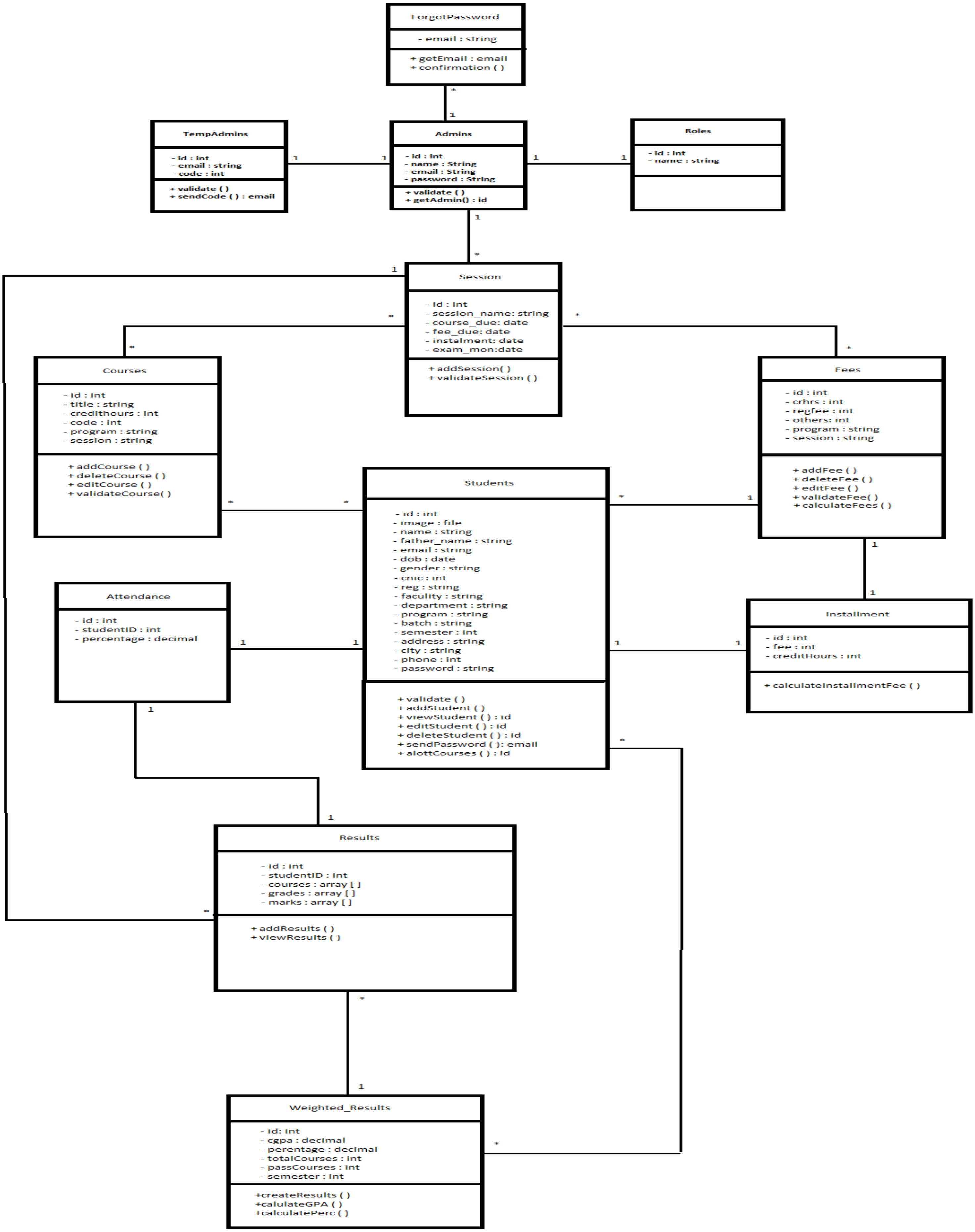
In a class diagram, the classes are arranged in groups that share common characteristics. A class diagram resembles a flowchart in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the attributes of the class; the lower rectangle contains the methods, also called operations, of the class. Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes.

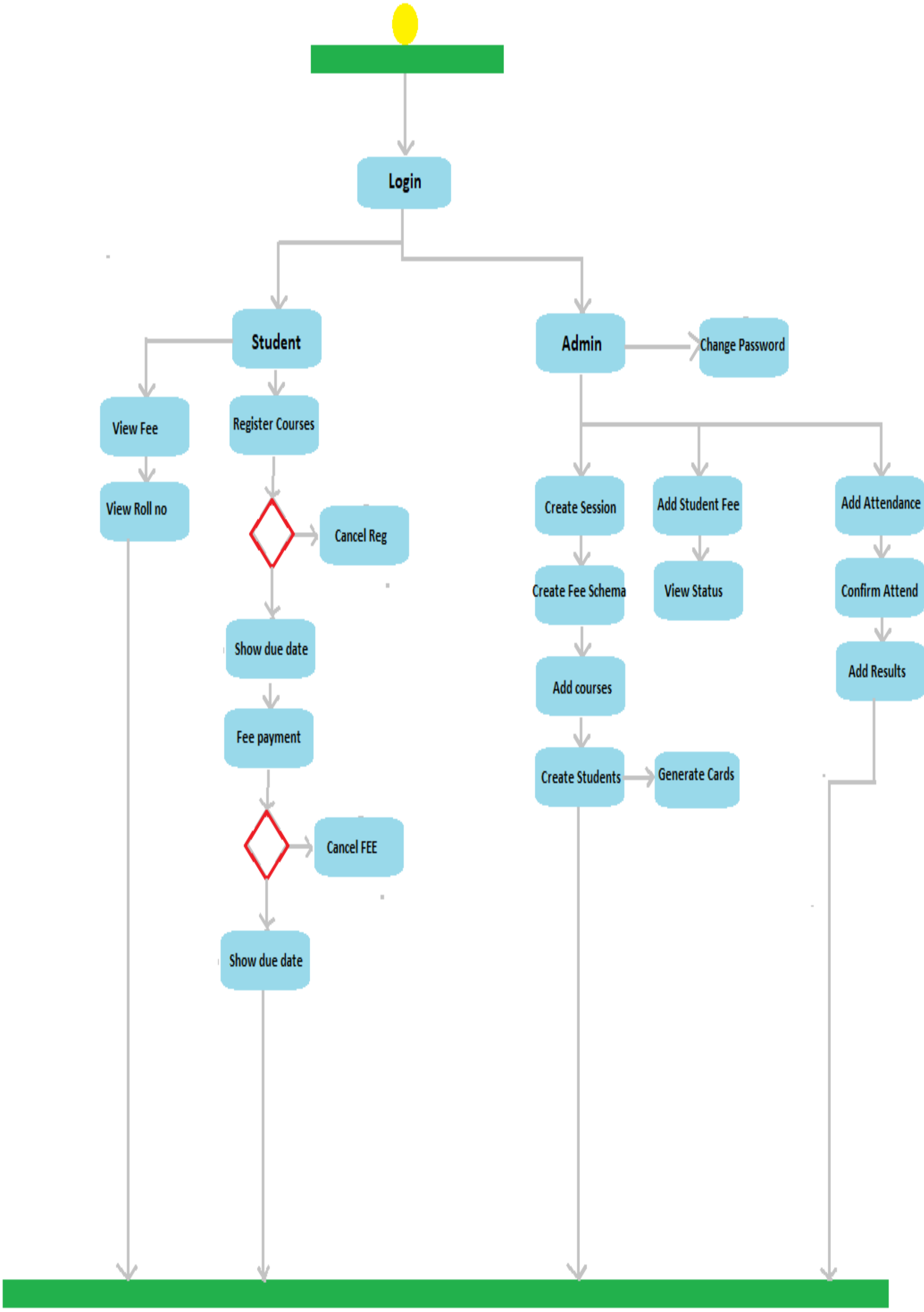
Class diagram as following attribute:

Relationships and associations are shown as lines connection elements, and are annotated to describe the relationships and their cardinality (1…1, 1…*, 0…*, etc).

- Inheritance (generalization/Specialization), aggregation (comprises), and composition (has) relationships are their data types are identified here, as are the operations and their return types.
- Visibility is indicated by +, #, or – for public, protected, or private.

The class diagram plays a vital role in the transition from design to construction as it contains sufficient detail to begin the coding process. It I often used to partition responsibilities among the project team members, and to guide and measure the construction process.


ForgotPassword
- email : string
+ getEmail : email
+ confirmation ( )
*
1
TempAdmins
- id : int
- email : string
- code : int
+ validate ( )
+ sendCode ( ) : email
1
1
Admins
- id : int
- name : String
- email : String
- password : String
+ validate ( )
+ getAdmin() : id
1
1
Roles
- id : int
- name : string
1
*
1
Session
- id : int
- session_name: string
- course_due: date
- fee_due: date
- instalment: date
- exam_mon:date
+ addSession( )
+ validateSession ( )
*
*
*
*
Courses
- id : int
- title : string
- credithours : int
- code : int
- program : string
- session : string
+ addCourse ( )
+ deleteCourse ( )
+ editCourse ( )
+ validateCourse( )
Fees
- id : int
- crhrs : int
- regfee : int
- others: int
- program : string
- session : string
+ addFee ( )
+ deleteFee ( )
+ editFee ( )
+ validateFee( )
+ calculateFees ( )
*
*
*
1
Students
- id : int
- image : file
- name : string
- father_name : string
- email : string
- dob : date
- gender : string
- cnic : int
- reg : string
- faculty : string
- department : string
- program : string
- batch : string
- semester : int
- address : string
- city : string
- phone : int
- password : string
+ validate ( )
+ addStudent ( )
+ viewStudent ( ) : id
+ editStudent ( ) : id
+ deleteStudent ( ) : id
+ sendPassword ( ): email
+ alottCourses ( ) : id
1
1
Attendance
- id : int
- studentID : int
- percentage : decimal
1
1
1
1
Installment
- id : int
- fee : int
- creditHours : int
+ calculateInstallmentFee ( )
1
*
1
Results
- id : int
- studentID : int
- courses : array [ ]
- grades : array [ ]
- marks : array [ ]
+ addResults ( )
+ viewResults ( )
*
*
1
Weighted_Results
- id: int
- cgpa : decimal
- perentage : decimal
- totalCourses : int
- passCourses : int
- semester : int
+createResults ( )
+calulateGPA ( )
+calculatePerc ( )
*

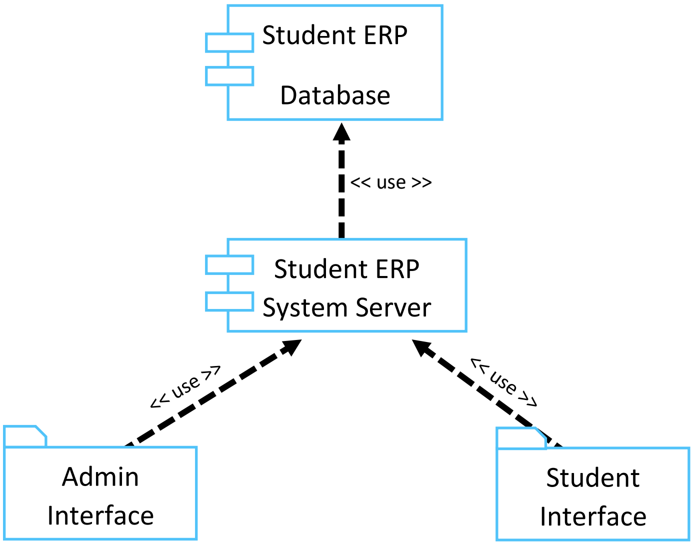

Login
Student
Admin
Change Password
View Fee
Register Courses
View Roll no
Cancel Reg
Show due date
Fee payment
Cancel FEE
Show due date
Create Session
Add Student Fee
Add Attendance
Create Fee Schema
View Status
Confirm Attend
Add courses
Add Results
Create Students
Generate Cards

Chapter 4

IMPLEMENTATION

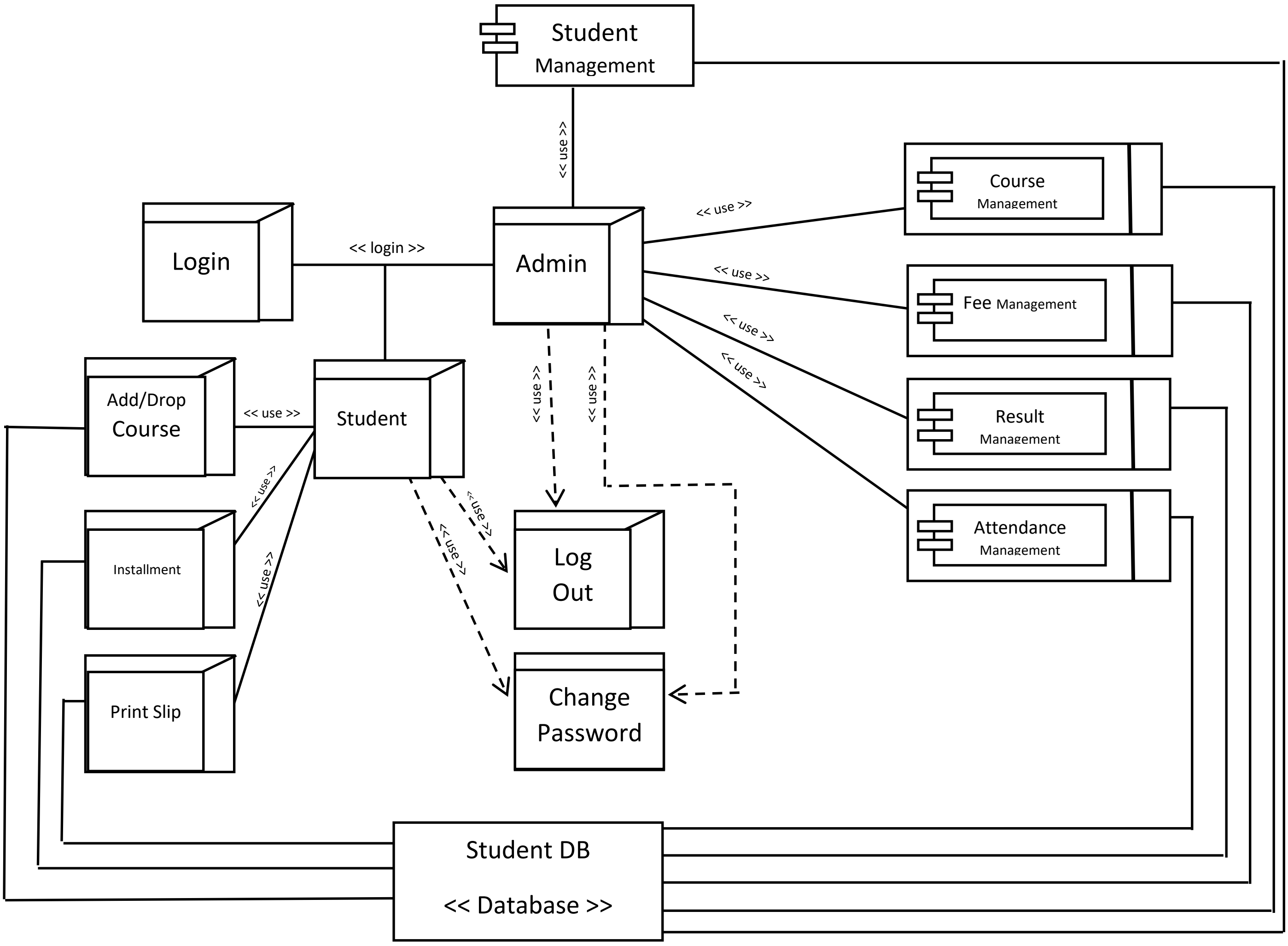
Component Diagram

Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.



Deployment Diagram

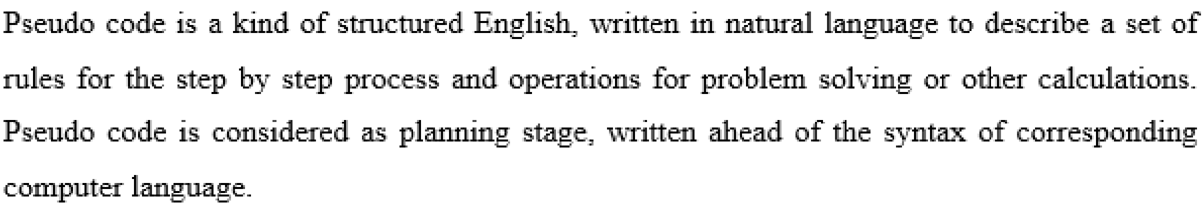
Deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modeling the physical aspects of an object-oriented system. They are often be used to model the static deployment view of a system (topology of the hardware)



4. Implementation

Implementation of the project is given below:

4.1 Pseudo code

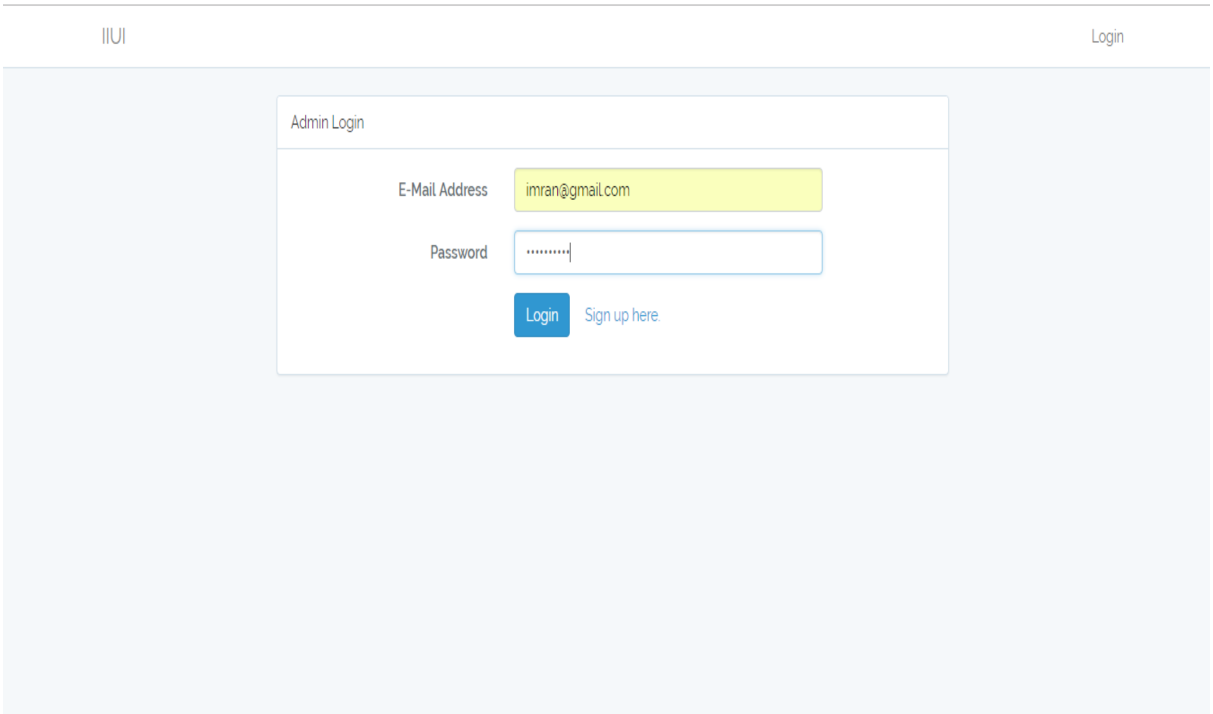


Pseudo code is a kind of structured English, written in natural language to describe a set of rules for the step by step process and operations for problem solving or other calculations. Pseudo code is considered as planning stage, written ahead of the syntax of corresponding computer language.

4.1.1 **Pseudo code for login:**

- Goto login page.
- Have the user fill out the fields.
- Have the page process the fields, send them to mysql database, check them. If the fields check, goto step 5. If not, to step 4.
- Show error message.
- Start session, go to default redirect after login.
- the person do whatever they want.
- the user is on for a prolonged period of time or their computer goes awry.
- what happens next? does the computer automatically log out?
- Complete
- the user logsout
- end session
- complete

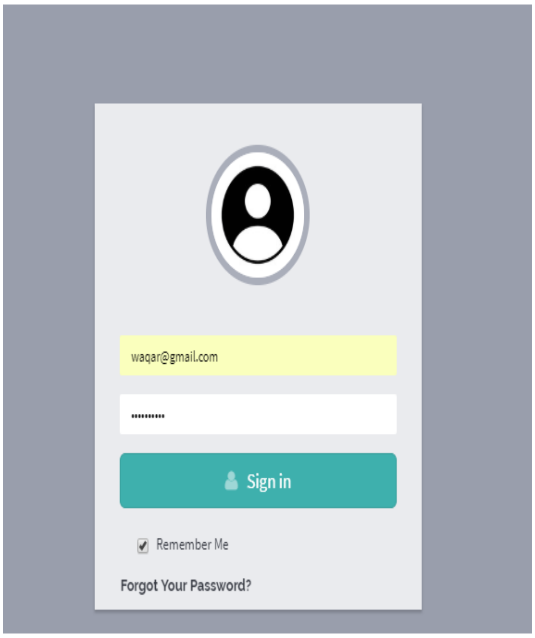
```
GET Username
GET Password
IF (Username = EnteredUsername & Password = EnteredPassword) THEN
    Login Successful
ELSE IF (Username != NULL & Password != NULL) THEN
    Login Failed
ELSE
    Login Failed
ENDIF
```



Admin Login

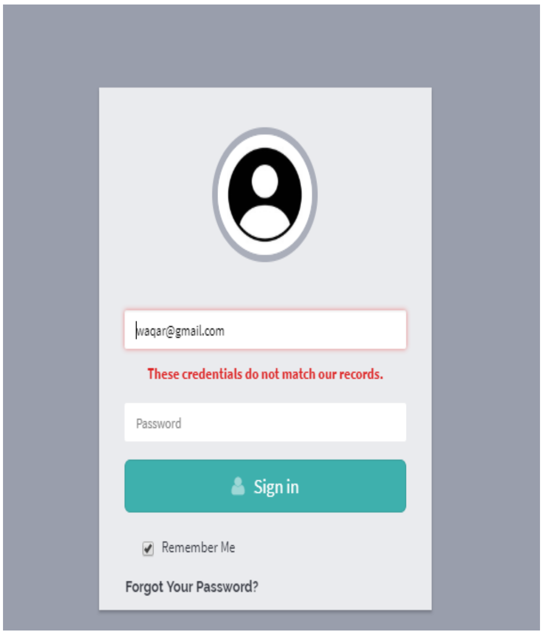
E-Mail Address: imran@gmail.com

Password: ••••••••••



waqar@gmail.com

•••••••••



waqar@gmail.com

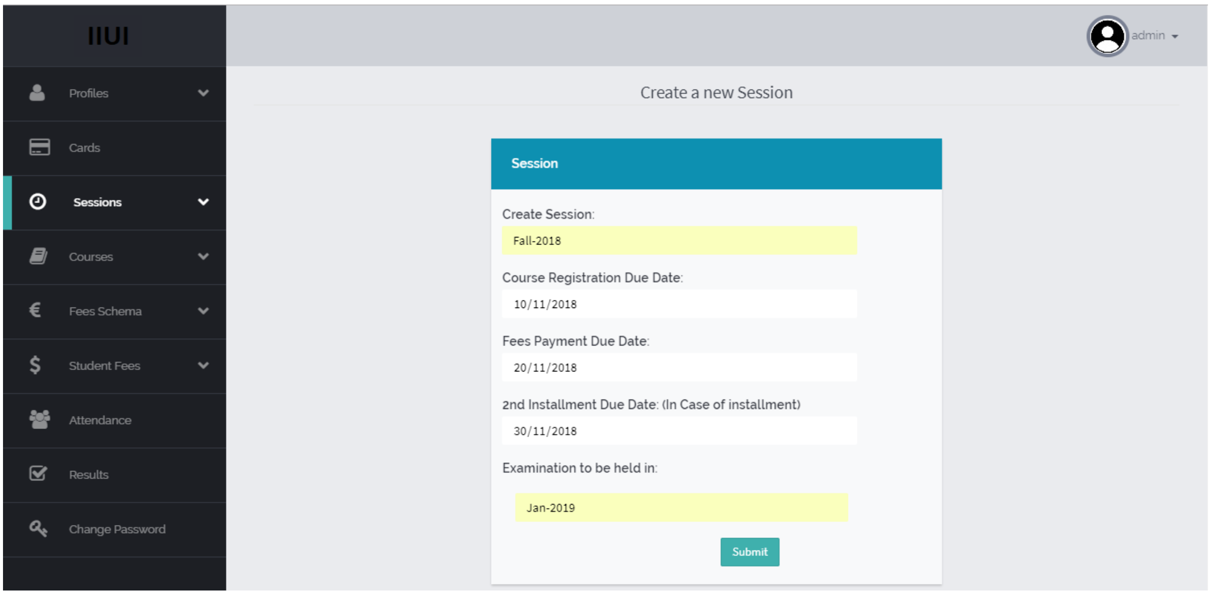
These credentials do not match our records.

4.1.2 Pseudo code for Creating Session:

- Click on the Create Session link inside admin home

- Insert name, Due date and more details of the session
- Click [Create Session].

```
CREATE Textfield_Session_Name
CREATE Textfield_Course_Duedate
CREATE Textfield_Fee_Duedate
CREATE Textfield_InstallmentFee_Duedate
On_Mouse_Click
        GET SessionName
        GET Course_Duedate
        GET Fee_Duedate
        IF (SessionName ==NULL OR Course_Duedate == NULL OR Fee_Duedate ==Null) THEN
        Error ‘SESSIONNAME and Others is required’
        ELSE IF (SNAME == DB.SNAME) THEN
        Error SESSION already exists’
        ELSE
        Session successfully added
        CREATE New Command
        GET Sessions
        ADD Command
Command ID will be auto generated
DISPLAY New Command Added Successfully
IF (Exception)
DISPLAY Error message
```



Create a new Session

Session

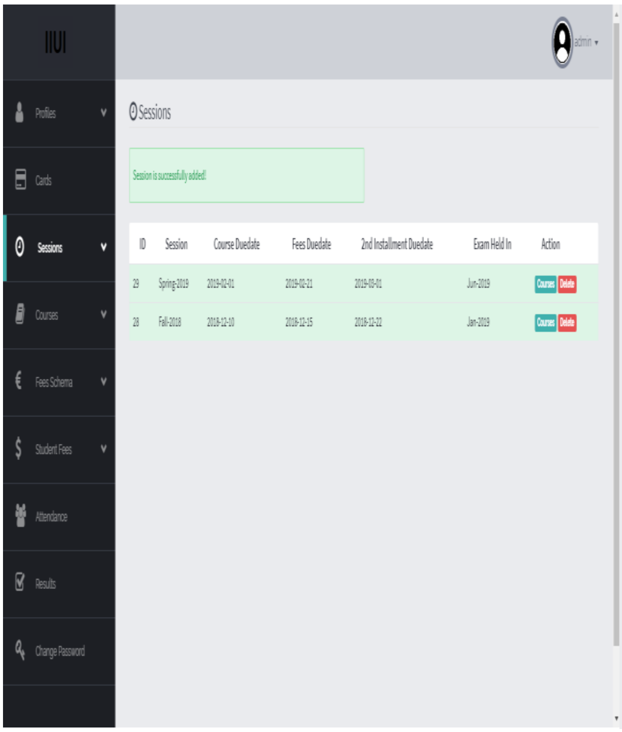
Create Session: Fall-2018

Course Registration Due Date: 10/11/2018

Fees Payment Due Date: 20/11/2018

2nd Installment Due Date: (In Case of installment) 30/11/2018

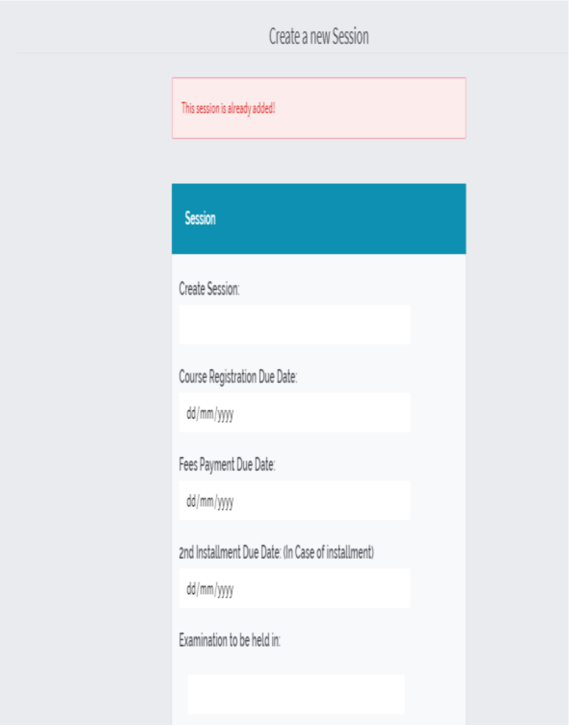
Examination to be held in: Jan-2019



Sessions

Session is successfully added!

ID	Session	Course Duedate	Fees Duedate	2nd Installment Duedate	Exam Held In	Action
29	Spring-2019	2019-02-01	2019-02-21	2019-03-01	Jun-2019	Courses Delete
28	Fall-2018	2018-12-10	2018-12-15	2018-12-22	Jan-2019	Courses Delete



Create a new Session

This session is already added!

Session

Create Session:

Course Registration Due Date: dd/mm/yyyy

Fees Payment Due Date: dd/mm/yyyy

2nd Installment Due Date: (In Case of installment) dd/mm/yyyy

Examination to be held in:

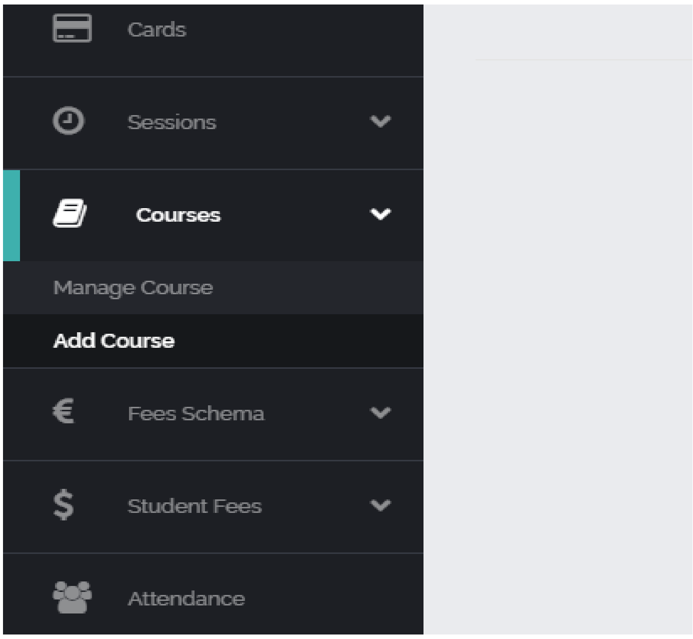
4.1.3 Pseudo code for Adding a new course:

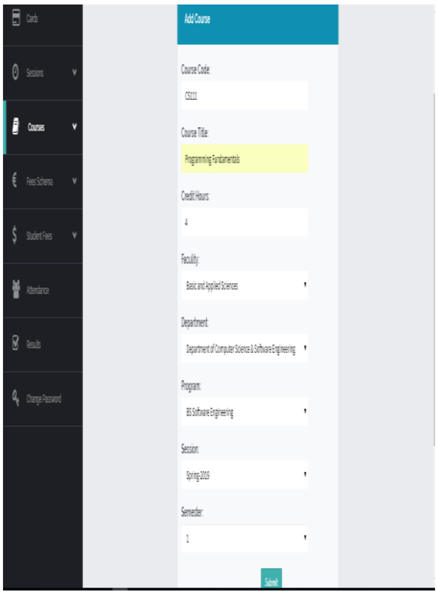
- Click on courses from admin side
- The *Add New Course*
- Enter the new course's title
- Enter the new course's code
- Enter the new course's credit hours
- Select the new course's program
- Select the new course's session
- Click [add course]

```
CREATE TextField_UniqueID (Auto)
CREATE Textfield_CourseTitle
CREATE Textfield_CourseCode
CREATE Textfield_CourseCRHRS
CREATE Textfield_CourseProgram
CREATE Textfield_CourseSession


On_Mouse_Click
        GET  UniqueID
        GET CourseName
        IF (CourseName ==NULL OR  UniqueID == NULL) THEN
        Error ‘CourseID and CourseNAME is required’
        ELSE IF (CNAME == DB.CNAME) THEN
        Error ‘COURSE’ already exists’
        ELSE
Course successfully added
        CREATE New Command
        GET Courses
        ADD Command
Command ID will be auto generated
DISPLAY New Command Added Successfully
IF (Exception)
```

DISPLAY Error message





Add Course

Course Code: CS111

Course Title: Programming Fundamentals

Credit Hours: 4

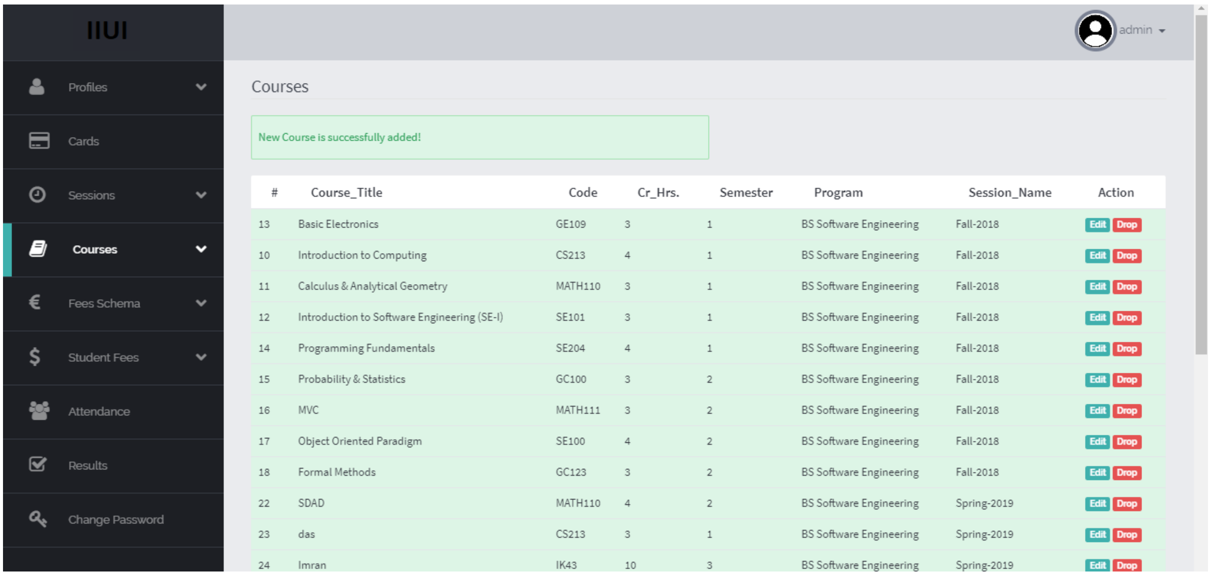
Faculty: Basic and Applied Sciences

Department: Department of Computer Science & Software Engineering

Program: BS Software Engineering

Session: Spring-2019

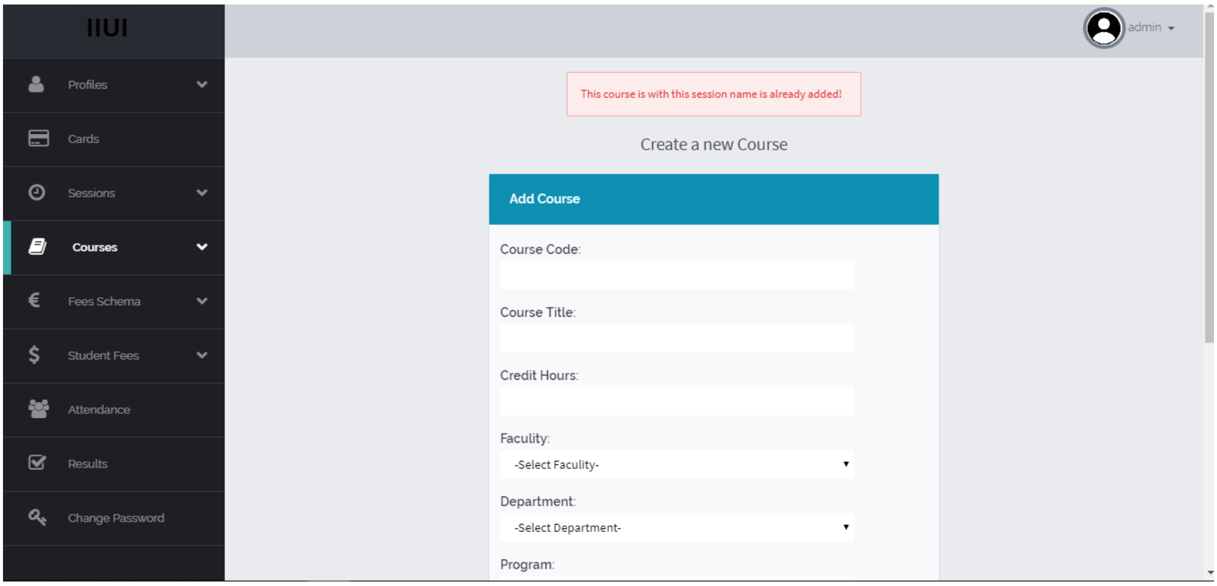
Semester: 1



Courses

New Course is successfully added!

#	Course_Title	Code	Cr_Hrs.	Semester	Program	Session_Name	Action
13	Basic Electronics	GE109	3	1	BS Software Engineering	Fall-2018	Edit Drop
10	Introduction to Computing	CS213	4	1	BS Software Engineering	Fall-2018	Edit Drop
11	Calculus & Analytical Geometry	MATH110	3	1	BS Software Engineering	Fall-2018	Edit Drop
12	Introduction to Software Engineering (SE-I)	SE101	3	1	BS Software Engineering	Fall-2018	Edit Drop
14	Programming Fundamentals	SE204	4	1	BS Software Engineering	Fall-2018	Edit Drop
15	Probability & Statistics	GC100	3	2	BS Software Engineering	Fall-2018	Edit Drop
16	MVC	MATH111	3	2	BS Software Engineering	Fall-2018	Edit Drop
17	Object Oriented Paradigm	SE100	4	2	BS Software Engineering	Fall-2018	Edit Drop
18	Formal Methods	GC123	3	2	BS Software Engineering	Fall-2018	Edit Drop
22	SDAD	MATH110	4	2	BS Software Engineering	Spring-2019	Edit Drop
23	das	CS213	3	1	BS Software Engineering	Spring-2019	Edit Drop
24	Imran	IK43	10	3	BS Software Engineering	Spring-2019	Edit Drop



This course is with this session name is already added!

Create a new Course

Add Course

Course Code:

Course Title:

Credit Hours:

Faculty: -Select Faculty-

Department: -Select Department-

Program:

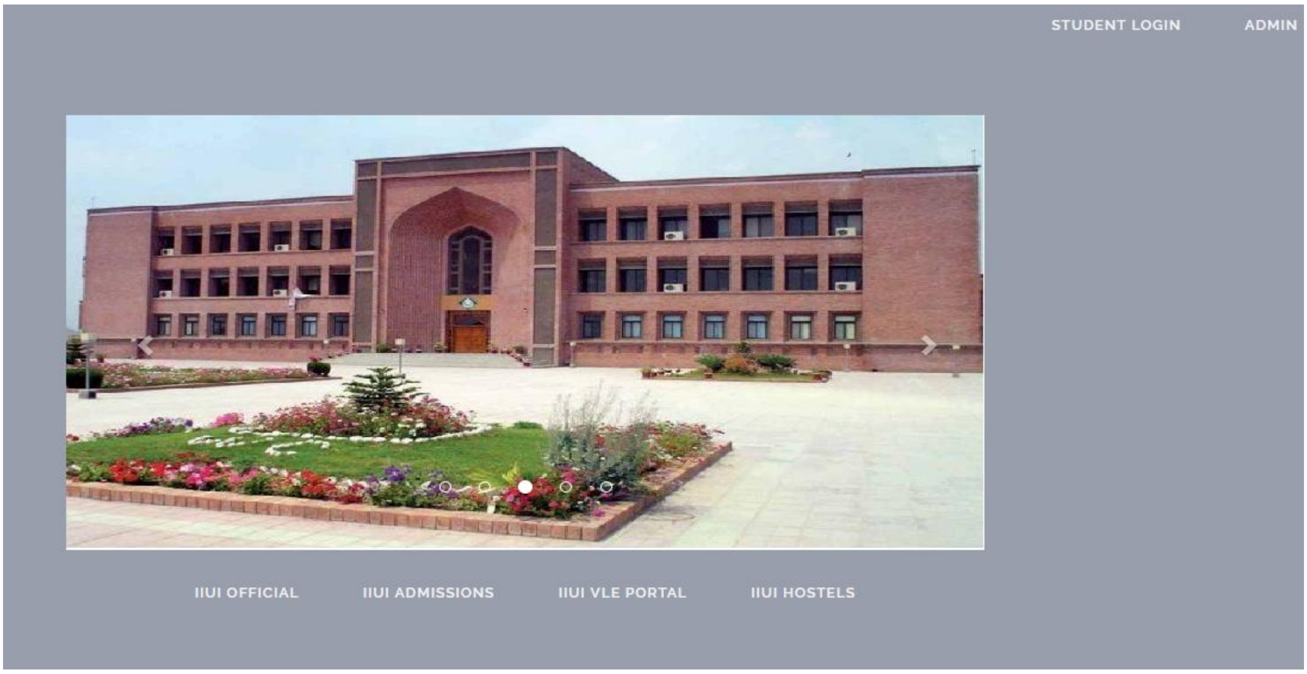
4.1.4 Pseudo code for Admin Logout:

- Goto login page.
- Have the user fill out the fields.
- Have the page process the fields, send them to mysql database, check them. If the fields check, goto step 5. If not, to step 4.
- Show error page.
- Start session, go to default redirect after login.
- the person do wahtever they wnat.
- the user is on for a prolonged period of time or their computer goes awry.
- what happens next? does the computer automaticlly log out?
- Complete
- the user logsout
- end session
- complete

```
CREATE Button_Admin_Logout
On Button_Recordings Click
        SET Access IS NULL
IF (Exception)
DISPLAY Error message
```

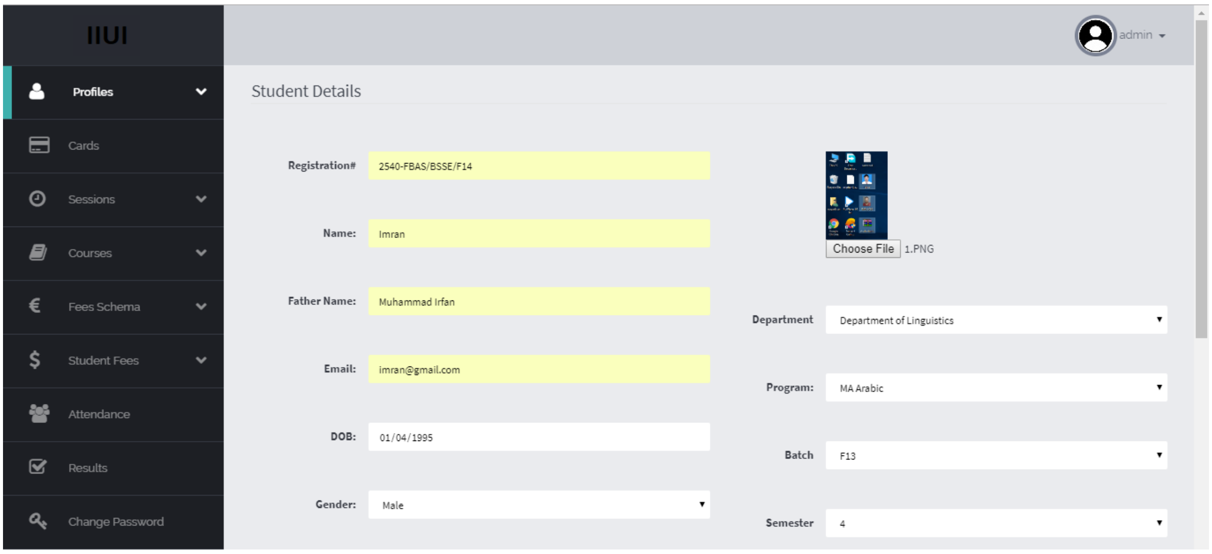
4.1.5 Pseudo code for Go To Home:

```
CREATE Button_HOME
On Button_Recordings Click
        GO TO Home Panel
        DISPLAY Application Details
IF (Exception)
DISPLAY Error message
```



4.1.6 Pseudo code for Manage Students:

```
CREATE Button_Change_Settings
CREATE Button_Add_New_Client
CREATE Button_Change_Password
CREATE Button_Delete_Client
DISPLAY Button_Add_New_Client IN Panel
DISPLAY Button_Change_Password IN Panel
DISPLAY  Button_Delete_Client IN Panel
IF (Exception)
DISPLAY Error message
```



Student Details

Registration#: 2540-FBAS/BSSE/F14

Name: Imran

Father Name: Muhammad Irfan

Email: imran@gmail.com

DOB: 01/04/1995

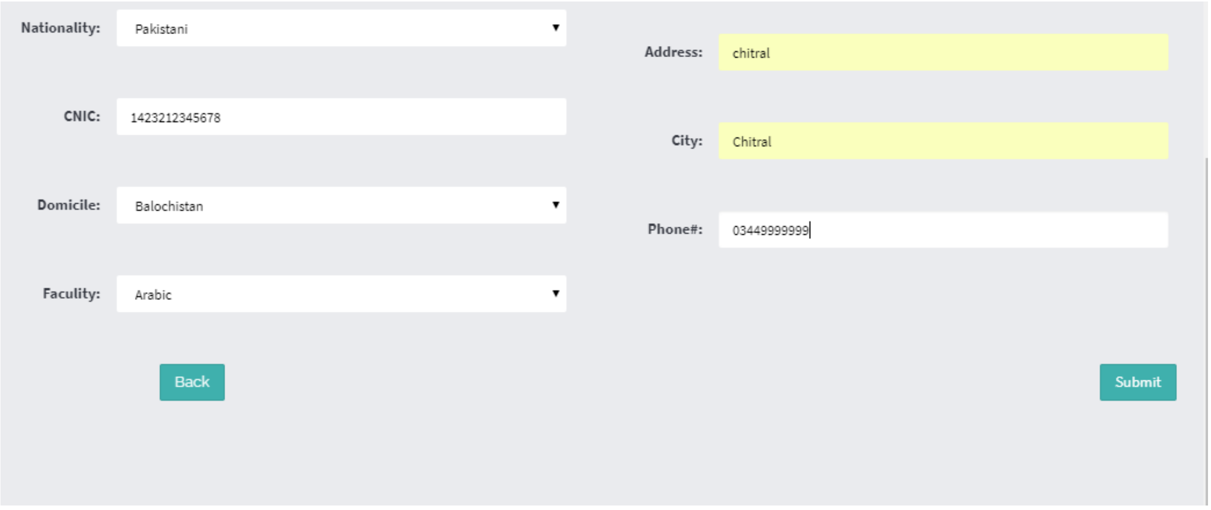
Gender: Male

Department: Department of Linguistics

Program: MA Arabic

Batch: F13

Semester: 4



Nationality: Pakistani

CNIC: 1423212345678

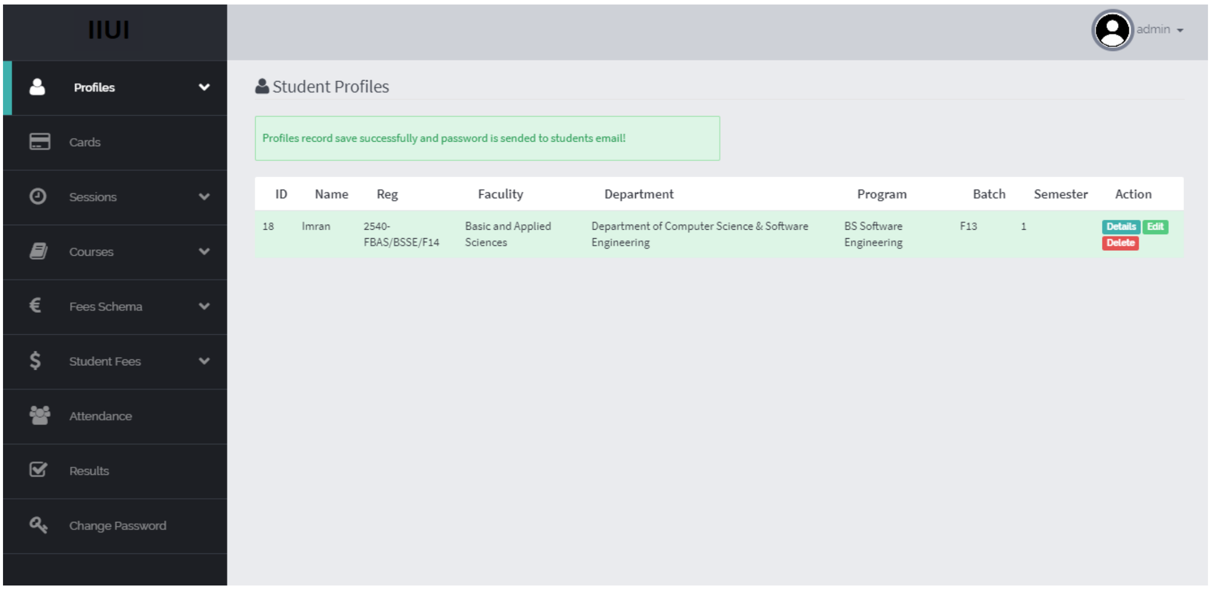
Domicile: Balochistan

Faculty: Arabic

Address: chitral

City: Chitral

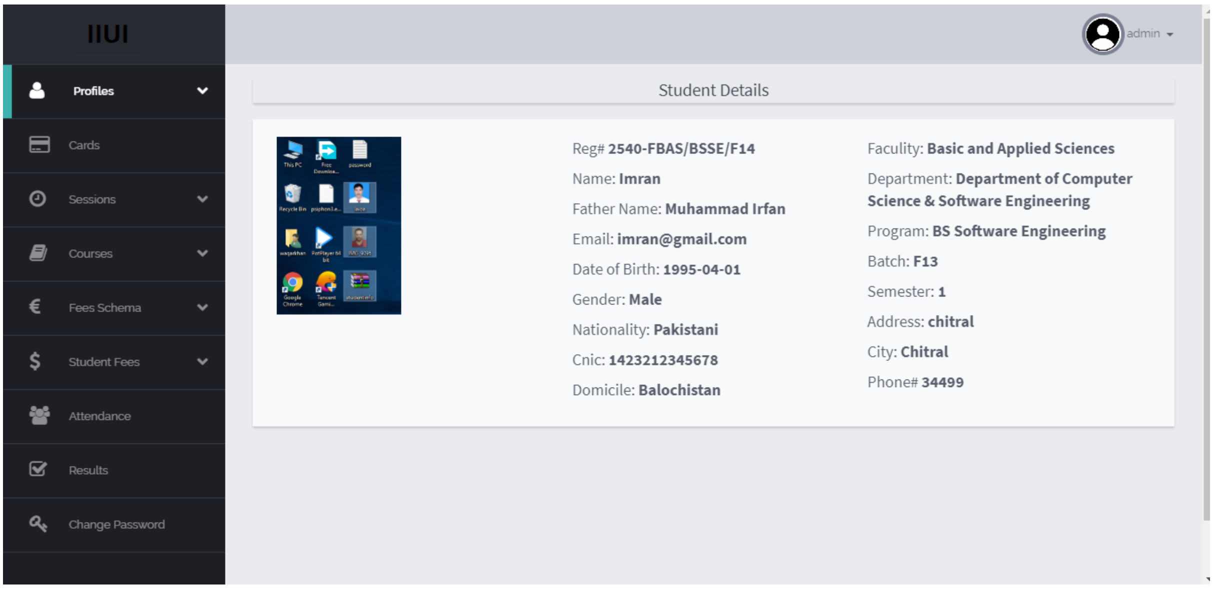
Phone#: 03449999999



Student Profiles

Profiles record save successfully and password is sended to students email!

ID	Name	Reg	Faculty	Department	Program	Batch	Semester	Action
18	Imran	2540-FBAS/BSSE/F14	Basic and Applied Sciences	Department of Computer Science & Software Engineering	BS Software Engineering	F13	1	Details Edit Delete



Student Details

Reg# **2540-FBAS/BSSE/F14**

Name: **Imran**

Father Name: **Muhammad Irfan**

Email: **imran@gmail.com**

Date of Birth: **1995-04-01**

Gender: **Male**

Nationality: **Pakistani**

Cnic: **1423212345678**

Domicile: **Balochistan**

Faculty: **Basic and Applied Sciences**

Department: **Department of Computer Science & Software Engineering**

Program: **BS Software Engineering**

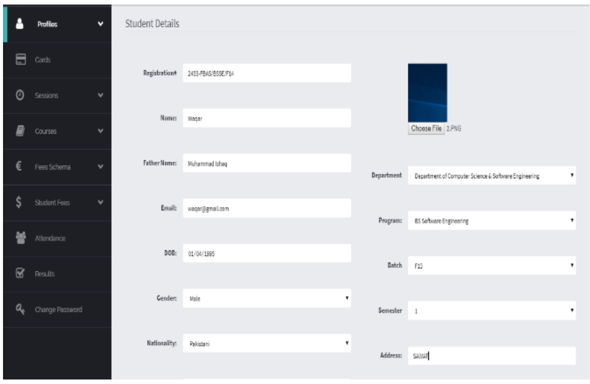
Batch: **F13**

Semester: **1**

Address: **chitral**

City: **Chitral**

Phone# **34499**



Student Details

Registration# 2433-FBAS/BSSE/F14

Name: Waqar

Father Name: Muhammad Ishaq

Email: waqar@gmail.com

DOB: 01/04/1995

Gender: Male

Nationality: Pakistani

2.PNG

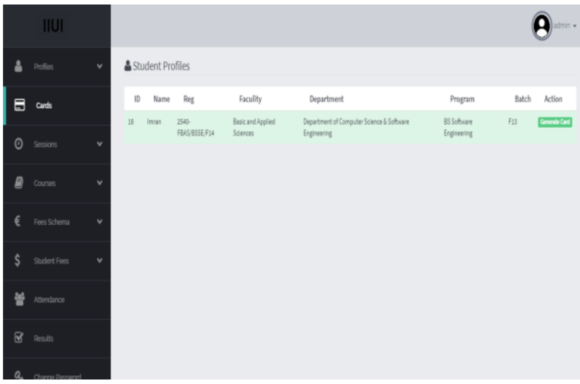
Department Department of Computer Science & Software Engineering

Program: BS Software Engineering

Batch F13

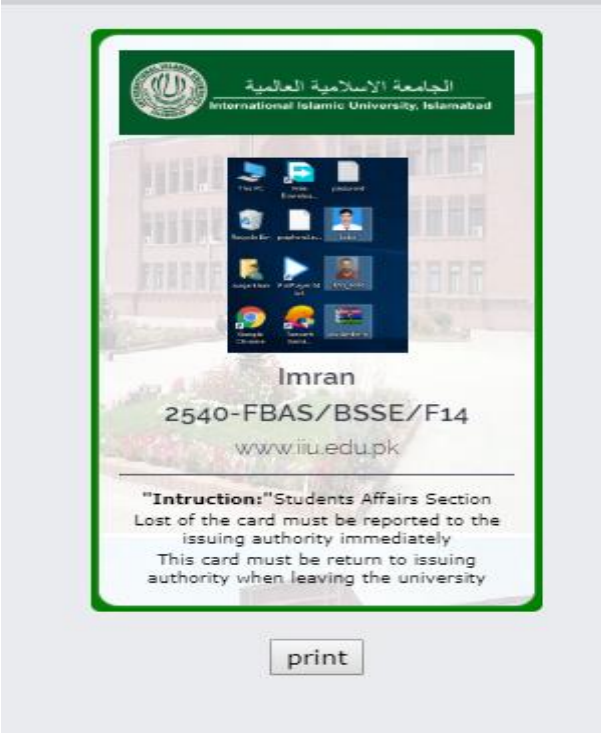
Semester 1

Address: SAWAT



Student Profiles

ID	Name	Reg	Faculty	Department	Program	Batch	Action
18	Imran	2540-FBAS/BSSE/F14	Basic and Applied Sciences	Department of Computer Science & Software Engineering	BS Software Engineering	F13	Generate Card



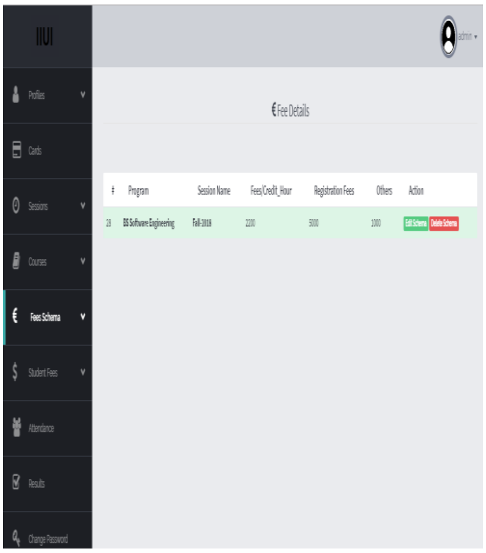
الجامعة الاسلامية العالمية
International Islamic University, Islamabad

Imran

2540-FBAS/BSSE/F14

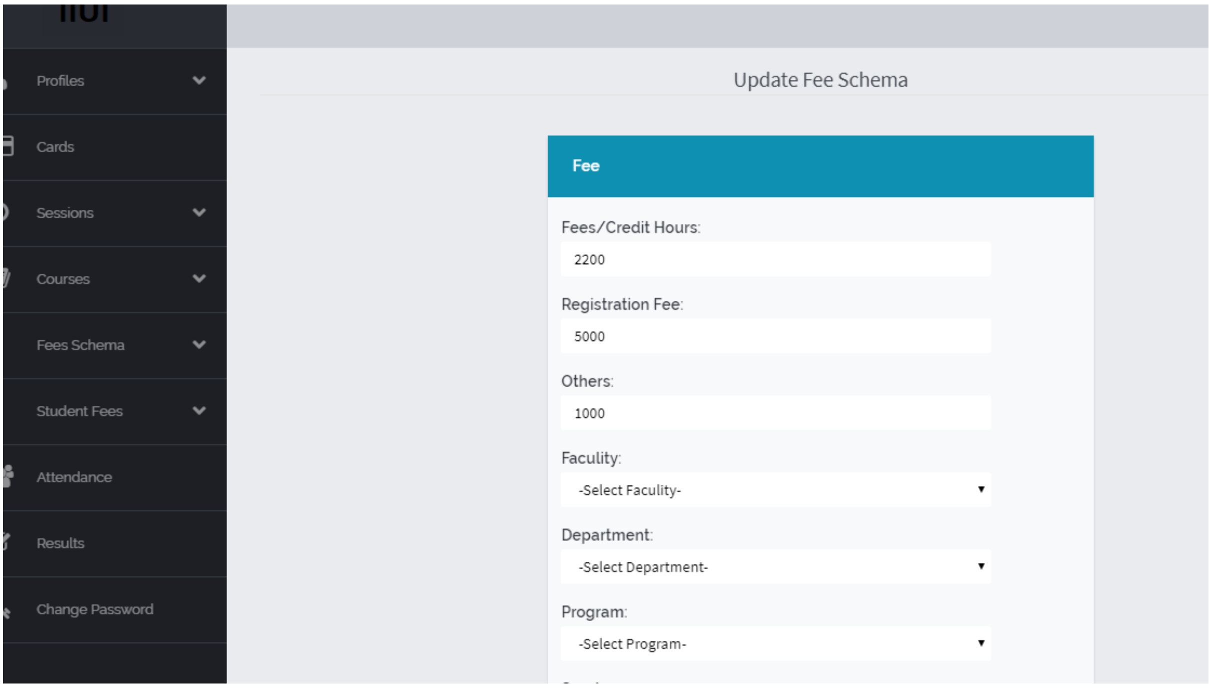
www.iiu.edu.pk

"Intruction:"Students Affairs Section
Lost of the card must be reported to the issuing authority immediately
This card must be return to issuing authority when leaving the university



Fee Details

#	Program	Session Name	Fees/Credit_Hour	Registration Fees	Others	Action
28	BS Software Engineering	Fall-2018	2200	5000	1000	Edit Schema Delete Schema



Update Fee Schema

Fee

Fees/Credit Hours: 2200

Registration Fee: 5000

Others: 1000

Faculty: -Select Faculty-

Department: -Select Department-

Program: -Select Program-

Chapter 5

TESTING

5. Testing

Software testing is an analysis conducted to provide information about quality of product with respect to the context in which it is intended to operate. Testing is the process of executing program with the intent of finding an error.

5.1 Test Cases

A test case is a set of conditions or variables under which a tester determine whether an application or software is working correctly or not.

5.1.1 Register Admin

1. Goto login page.
2. Click on register link
3. Have the user fill out the fields.
4. Email is sended to the admins email
5. Insert 6 digits unique random code
6. System verify and generate new ADMIN

Test Case ID:	**TC-001**	
Associated Use Case:	Register Admin	
Functionality to be Tested:	Test Successful Register	
Actor:	Admin	
Pre-Conditions:	Server GUI is launched successfully	
Post Conditions:	Successful Registration	
Steps:	**Expected Result**	**Actual Result**
1.Enter name, email and password 2.Tap register 3.Enter random code 4.Tap verify	Register new admin to the system	Admin Registered Successfully

Test Case Status:	**Fail**

Table 5.1.1Test Case : Register Admin

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-001	Validate user and register to the system	Name and email cannot be empty	1. Do not enter the value as "12345" ,"/*--+- in the name field" 2. Do not enter the value as Null Characters or " " blank space 3. Enter the value as "a to z" in the first name field 4. Entering Infinite entry in the field	Checking the functionality of the text box " Name" ,"Email"	1) Should show the error message "Invalid entry. First Name accepts only characters". 2) Should show the error message "Please enter your Name". 3) Should accept the value and show the pop up message as "Please enter your email". 5) Should able to accept up to "40" Characters	Sucessulfuly register	Fail	Registration successful

5.1.2 Login Admin

7. Goto login page.
8. Have the user fill out the fields.
9. Have the page process the fields, send them to mysql database, check them. If the fields check, goto step 5. If not, to step 4.
10. Show error page.
11. Start session, go to default redirect after login.
12. the person do wahtever they wnat.
13. the user is on for a prolonged period of time or their computer goes awry.
14. what happens next? does the computer automaticlly log out?
15. Complete
16. the user logsout
17. end session
18. complete

Test Case ID:	**TC-002**	
Associated Use Case:	Admin Login	
Functionality to be Tested:	Test Successful Login	
Actor:	Admin	
Pre-Conditions:	Server GUI is launched successfully	
Post Conditions:	Successful login	
Steps:	**Expected Result**	**Actual Result**
1.Enter email and password 2.Taps Login button	Grant Access to the system	Grant accessed
Test Case Status:	**Fail**	

Table 5.1.2Test Case : Login Admin

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-002	Validate if the user is register to the system	Name and email cannot be empty	1. Do not enter the value as "12345" ,"/*--+- in the name field" 2. Do not enter the value as Null Characters or " " blank space 3. Enter the value as "a to z" in the first name field 4. Entering Infinite entry in the field	Checking the functionality of the text box " Name" ,"Email"	1) Should show the popup error message "Invalid entry. First Name accepts only characters". 2) Should show the popup error message "Please enter your Name". 3) Should accept the value and show the pop up message as "Please enter your email". 5) Should able to accept up to "40" Characters	Sucessulfuly login	Fail	Login successful

5.1.3 Create Session

19. Goto Create Session
20. Have the admin fill out the fields.
21. Tap add session button
22. Showing completion message
23. complete

Test Case ID:	**TC-003**	
Associated Use Case:	Create Session	
Functionality to be Tested:	Test Successful Create session	
Actor:	Admin	
Pre-Conditions:	Admin login to the system	
Post Conditions:	Successful session creation	
Steps:	**Expected Result**	**Actual Result**
1.Enter Session Name, Course Due date, Fee Due date, Installment Due date etc. 2.Taps add session button	Create a new session for the students	Session created
Test Case Status:	**Fail**	

Table 5.1.3Test Case : Create Session

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-003	Validate if the session is create for the students	All fields cannot be empty	1. Do not enter the value as Null Characters or “ “ blank space 2. Select date values in fields	Checking the functionality of the all text boxes	1) Should show the error message ”This session is already”. If session is already exist. 2) Should show the error message "Please enter Session Name". 3) Should accept the value and show the pop up message as "Please enter this field ".	Sucessulfuly session creation	Fail	Create session successful

5.1.4 Add Course

24. Goto Courses
25. Click Add Course
26. Gives input fields
27. Tap on add new course
28. Showing completion message
29. Complete

Test Case ID:	**TC-004**	
Associated Use Case:	Add course	
Functionality to be Tested:	Test Successful adding new course	
Actor:	Admin	
Pre-Conditions:	Admin login to the system and session must be created first	
Post Conditions:	Successfully new course is added	
Steps:	**Expected Result**	**Actual Result**
1.Enter Course Title, Course Code, Course CRHRS, Program and select session from the list which is created before courses. 2.Taps add course button	Create a new course schema for the students	Course added
Test Case Status:	**Fail**	

Table 5.1.4 Test Case : Add Course

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-004	Validate if the course is create for the students	All fields cannot be empty	1. Do not enter the value as Null Characters or “ “ blank space 2. Select session from the fields which is first added by admin	Checking the functionality of the all text boxes	1) Should show the error message ”This course is already added for this session”. If course is already exist with the same session. 2) Should show the error message "Please enter course Name".	Success fully course is added	Fail	Create courses successful

5.1.5 Create Fee Schema

30. Goto Fees
31. Click Add Fee Schema
32. Gives input fields
33. Tap on add new Fee Schema
34. Showing completion message
35. Complete

Test Case ID:	**TC-005**	
Associated Use Case:	Create Fee Schema	
Functionality to be Tested:	Test Successful adding new fee schema	
Actor:	Admin	
Pre-Conditions:	Admin login to the system and session must be created first	
Post Conditions:	Successfully new fee schema is added	
Steps:	**Expected Result**	**Actual Result**
1.Enter CRS, Registration fees, others, Program and select session from the list which is created before courses. 2.Taps add fee button	Create a new fee schema for the students	Fee Schema is added
Test Case Status:	**Fail**	

Table 5.1.5 Test Case : Create fee_sechema

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-005	Create new fee schema by program	All fields cannot be empty	1. Do not enter the value as Null Characters or " " blank space 2. Select session from the fields which is first added by admin	Checking the functionality of the all text boxes	1) Should show the error message "This Fee Schema is already added for this session". If course is already exist with the same session and program. 2) Should show the error message "Please enter course Name".	Success fully course is added	Fail	Create courses successful

5.1.6 Create students

36. Go to profile
37. Click create profile
38. Gives input fields
39. Add image
40. Tap on add new student
41. Showing completion message
42. complete

Test Case ID:	**TC-006**
Associated Use Case:	Create Student
Functionality to be Tested:	Test Successful adding creating new student profile
Actor:	Admin
Pre-Conditions:	Admin login to the system and session must be created first

Post Conditions:	Successfully new student is added	
Steps:	**Expected Result**	**Actual Result**
1.Taps add student button	Create a new student	New student is added
Test Case Status:	**Fail**	

Table 5.1.6Test Case : Create Students

Test Case #	Test Scenari o	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-006	Create new student by admin	All fields cannot be empty	1. Do not enter the value as Null Characters or " " blank space 2. Select session from the fields which is first added by admin	Checki ng the functio nality of the all text boxes	1) Should show the error message "This Fee Schema is already added for this session". If course is already exist with the same session and program. 2) Should show the error message "Please enter course Name".	Success fully student is added	Fail	Create students successful

5.1.7 Course Registration

43. Go to courses

44. Click Add courses

45. Add course

46. Click add button

47. Showing completion message

48. complete

Test Case ID:	**TC-007**	
Associated Use Case:	Course registration	
Functionality to be Tested:	Test Successful adding or dropping new course	
Actor:	Student	
Pre-Conditions:	Student is admitted and authenticated	
Post Conditions:	new course is added or current course is drop as choice	
Steps:	**Expected Result**	**Actual Result**
1.Taps add course 2. Select the course want to enrolled 3. Click add course	Add and drop of courses	Course registration
Test Case Status:	**Fail**	

Table 6.1.7 Test Case : Course Registration

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-007	Course registration	All fields cannot be empty	1. Do not enter the value as Null Characters or " " blank space 2. Select session from the fields which is first added by admin	Checking the functionality of the all text boxes	1) Should show the error message "This Course is already added for this session". If course is already exist with the same session and program. 2) Should show the error message "Please enter course Name".	Successfully add or drop course	Fail	Course registration

5.1.8 Course Registration

49. Go to courses
50. Click Add courses
51. Add course
52. Click add button
53. Showing completion message
54. Complete

Test Case ID:	**TC-008**
Associated Use Case:	Edit profile
Functionality to be Tested:	Edit profile
Actor:	Student
Pre-Conditions:	Student is admitted and authenticated
Post Conditions:	Save edited details

Steps:	**Expected Result**	**Actual Result**
1.Click edit profile button 2. Edit field want to edit 3. Click on update button	Edit profile	Edit and update of profile
Test Case Status:	**Fail**	

Table 5.1.8Test Case : Course Registration

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-008	Edit profile	All fields cannot be empty	1. Do not enter the value as Null Characters or " " blank space 2. Select session from the fields which is first added by admin	Checking the functionality of the all text boxes	1) Should show the error message "Please enter Name". if empty	Success fully edit and update student profile data	Fail	Edit profile

5.1.9 Drop Course

55. Go to courses
56. Click the course want to drop
57. Click on drop button
58. Showing completion message
59. complete

Test Case ID:	**TC-009**	
Associated Use Case:	Drop course	
Functionality to be Tested:	drop course	
Actor:	Student	
Pre-Conditions:	Student is admitted and authenticated	
Post Conditions:	Course delete/drop	
Steps:	**Expected Result**	**Actual Result**
1.Click on courses 2. Select course want to drop 3. Click drop button	Drop course	Drop and delete course from student record
Test Case Status:	**Fail**	

Table 5.1.9 Test Case : Drop Course

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-009	Drop course	All fields cannot be empty	1. Do not enter the value as Null Characters or " " blank space 2. Select session from the fields which is first added by admin	Checking the functionality of the all text boxes	1) Should show the message "course is successfully dropped".	Success fully drop or delete course from student record	Fail	Drop course

5.1.10 Full Fees

60. Go to fees
61. Click on full fee
62. Print fees invoice
63. complete

Test Case ID:	**TC-010**	
Associated Use Case:	Full fees	
Functionality to be Tested:	Full fees	
Actor:	Student	
Pre-Conditions:	Student must registered with courses	
Post Conditions:	Print fee slip	
Steps:	**Expected Result**	**Actual Result**

1.Click fees from menu 2. Click full fees button 3. Print fees invoice	Full fees	Print full invoice
Test Case Status:	**Fail**	

Table 5.1.10Test Case : Full Fees

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-010	Full fees	All fields cannot be empty	1. Do not enter the value as Null Characters or " " blank space 2. Select session from the fields which is first added by admin	Checking the functionality of the all text boxes	1) Should show the error message "This Fee Schema is already added for this session". If course is already exist with the same session and program. 2) Should show the error message "Please enter course Name".	Success fully print full fees challan	Fail	Full fees challan

5.1.11 Fees Installment

64. Go to fees
65. Click on installment
66. Print installment application
67. complete

Test Case ID:	**TC-011**	
Associated Use Case:	Fees installment	
Functionality to be Tested:	Fees installment	
Actor:	Student	
Pre-Conditions:	Student must registered with courses	
Post Conditions:	Divide fees in two instalments	
Steps:	**Expected Result**	**Actual Result**
1.Click fees from menu 2. Click full installment button 3. Print installment application	Fees installment	Print installment application
Test Case Status:	**Fail**	

Table 5.1.11 Test Case : Fees Installment

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-011	Fees installment	All fields cannot be empty	1. Do not enter the value as Null Characters or " " blank space 2. Select session from the fields which is first added by admin	Checking the functionality of the all text boxes	1) Should show the error message "This Fee Schema is already added for this session". If course is already exist with the same session and program. 2) Should show the error message "Please enter course Name".	Successfully print installment application	Fail	Two installment application

5.1.12 Manage Students

68. Admin login
69. Click manage students from menu
70. Edit or delete records
71. Showing completion message
72. Complete

Test Case ID:	**TC-012**
Associated Use Case:	Manage student
Functionality to be Tested:	Add, Update, Edit and Delete existing students record
Actor:	Admin
Pre-Conditions:	Student exist
Post Conditions:	Delete student record

Steps:	**Expected Result**	**Actual Result**
1.Click manage students from menu 2. Click update delete button 3. Update or delete student record	Manage students	Add, Update, Edit and Delete existing students record
Test Case Status:	**Fail**	

Table 5.1.12 Test Case : Manage Students

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-012	Manage student	All fields cannot be empty	1. Do not enter the value as Null Characters or " " blank space 2. Select session from the fields which is first added by admin	Checking the functionality of the all text boxes	1) Should show the error message "This Fee Schema is already added for this session". If course is already exist with the same session and program. 2) Should show the error message "Please enter course Name".	Successfully edit or delete record of existing students	Fail	Edit or delete student record

5.1.13 Add student fee

73. Admin login
74. Click fees from menu
75. Click add fees
76. Add fees of student
77. Click add
78. Showing completion message
79. Complete

Test Case ID:	**TC-013**	
Associated Use Case:	Add student fee	
Functionality to be Tested:	Update student fee status	
Actor:	Admin	
Pre-Conditions:	Student submitted fee	
Post Conditions:	Update student fee status	
Steps:	**Expected Result**	**Actual Result**
1.Click fees from menu 2. Click add fees 3. Add student fees	Add student fees	Update student fees status
Test Case Status:	**Fail**	

Table 5.1.13 Test Case : Add Student Fees

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-013	Add student fees	All fields cannot be empty	1. Do not enter the value as Null Characters or “ “ blank space 2. Select session from the fields which is first added by admin	Checking the functionality of the all text boxes	1) Should show the error message ”This Fee Schema is already added for this session”. If course is already exist with the same session and program. 2) Should show the error message "Please enter course Name".	Successfully update status of student fees record	Fail	Update student fees status

5.1.14 Attendance

80. Admin login
81. Click attendance from menu
82. Add student attendance
83. Showing completion message
84. complete

Test Case ID:	**TC-014**	
Associated Use Case:	Attendance	
Functionality to be Tested:	Add student attendance	
Actor:	Admin	
Pre-Conditions:	Total attendance	
Post Conditions:	Save student attendance	
Steps:	**Expected Result**	**Actual Result**

1.Click attendance from menu 2. Click add attendance 3. Add student attendance	Add student attendance	Add student attendance
Test Case Status:	**Fail**	

Table 5.1.14 Test Case : Attendance

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-014	Attendance	All fields cannot be empty	1. Do not enter the value as Null Characters or " " blank space 2. Select session from the fields which is first added by admin	Checking the functionality of the all text boxes	1) Should show the error message "This Fee Schema is already added for this session". If course is already exist with the same session and program. 2) Should show the error message "Please enter course Name".	Successfully add student attendance	Fail	Add student attendance

5.1.15 Result

85. Admin login
86. Click result from menu
87. Add student marks in courses he enrolled
88. Click add result
89. Showing completion message
90. Complete

Test Case ID:	**TC-015**	
Associated Use Case:	Result	
Functionality to be Tested:	Add student result	
Actor:	Admin	
Pre-Conditions:	Attendance clear	
Post Conditions:	Save students results	
Steps:	**Expected Result**	**Actual Result**
1.Click results from menu 2. Click add result 3. Add student marks in courses field who is register 4. Add/Save result	Add student attendance	Add student attendance
Test Case Status:	**Fail**	

Table 5.1.15 Test Case : Result

Test Case #	Test Scenario	Test Cases	Test Steps	Test Data	Expected Result	Actual Result	Pass /Fail	Remarks
TC-015	Results	All fields cannot be empty	1. Do not enter the value as Null Characters or “ “ blank space 2. Select session from the fields which is first added by admin	Checking the functionality of the all text boxes	1) Should show the error message ”This Fee Schema is already added for this session”. If course is already exist with the same session and program. 2) Should show the error message "Please enter course Name".	Successfully add student results	Fail	Add student result

Chapter 6

CONCLUSION

6. Conclusion

The following results have been achieved after completing the system and relate back to the system's objective. Should allow users to browse through different course categories: This is achieved through an easy to use graphical interface menu options. Should allow users to add courses to the list and view detailed information about the session in short period of time which is given before by admin: The users can add limit of 19 credit hours per session or semester only. Once courses is added, user is presented with detailed fees to review and print fee slip from the portal. Should allow the user to see Success message after fee payment: This is achieved when user successfully paid his her fees. The user is given the attendance details and results along with success message.

6.1 Further enhancements

The following section describes the work that will be implemented with future releases of the software. Enhance User Interface by adding more user interactive features. Provide Courses of the Session to Home Page

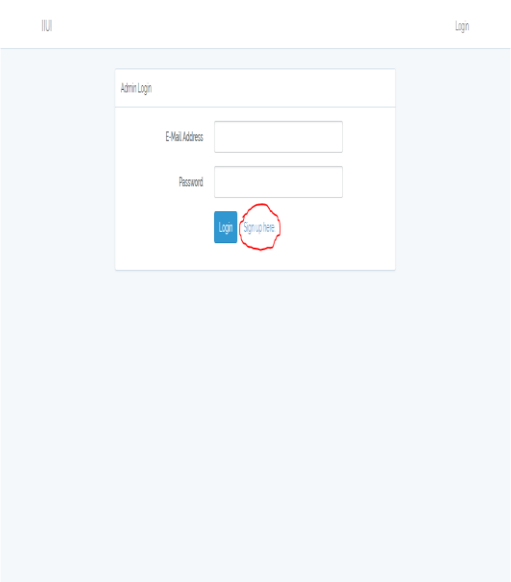
- Allow to add/drop courses in a period of time
- Course Options: Add Courses for current and less than current semester
- Allotment Process Estimate: Courses should be automatically allotted to the student on the end of course registration due date
- Fee Ready notification: Automatically generate fees and gives option for fee slip

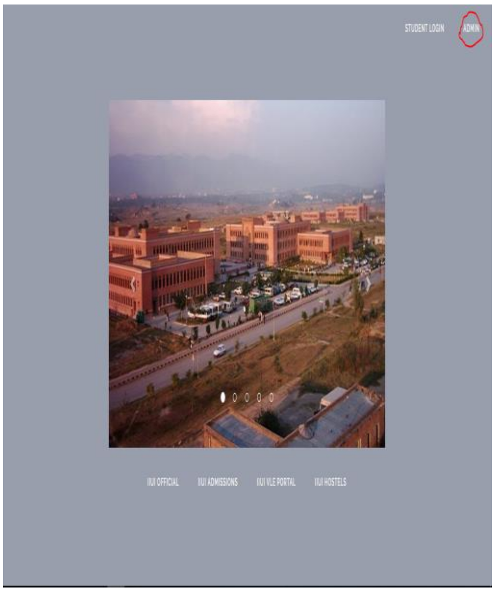
APPENDIX A

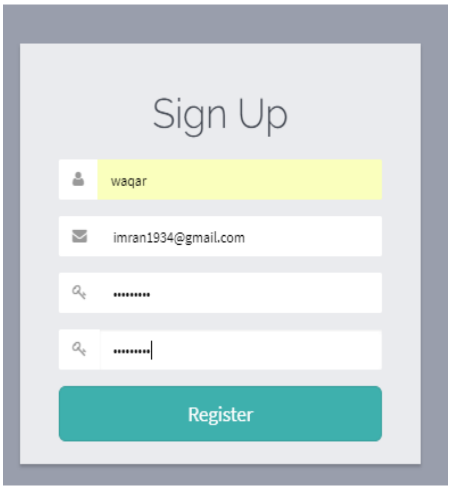
USER MANUAL

2 User Manual

A user guide or user's guide, also commonly known as a manual, is a technical communication document intended to give assistance to people using a particular system. It is usually written by a technical writer, although user guides are written by programmers, product or project managers, or other technical staff, particularly in smaller companies. User guides are most commonly associated with electronic goods, computer hardware and software.







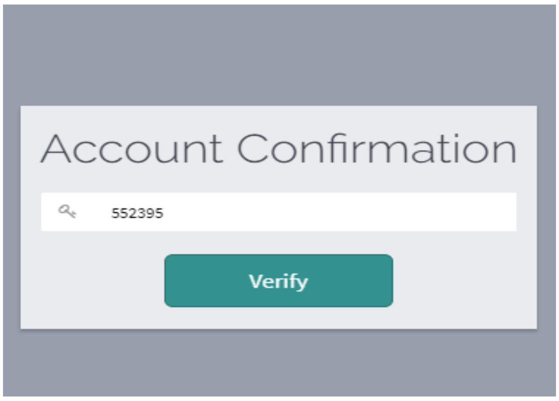
Sign Up

waqar

imran1934@gmail.com

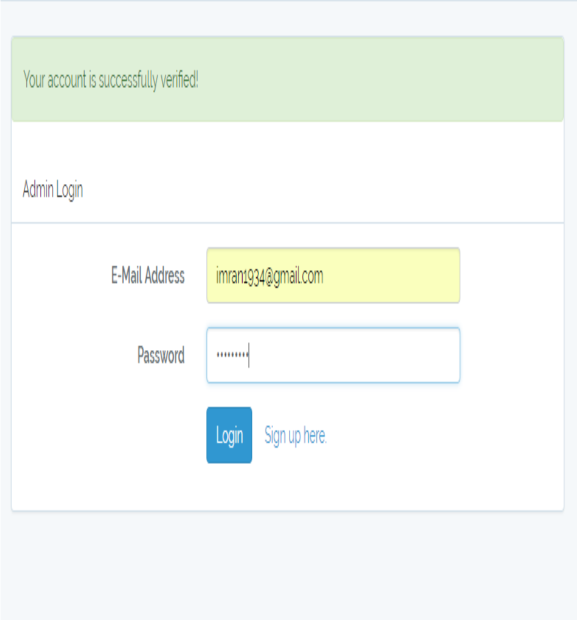
•••••••••

•••••••••



Account Confirmation

552395

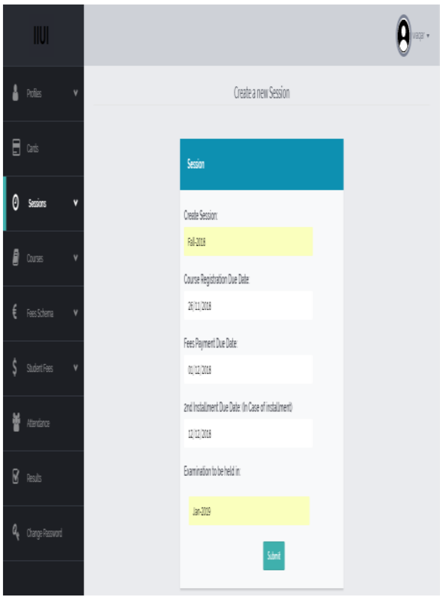


Your account is successfully verified!

Admin Login

E-Mail Address: imran1934@gmail.com

Password: ••••••••



Create a new Session

Session

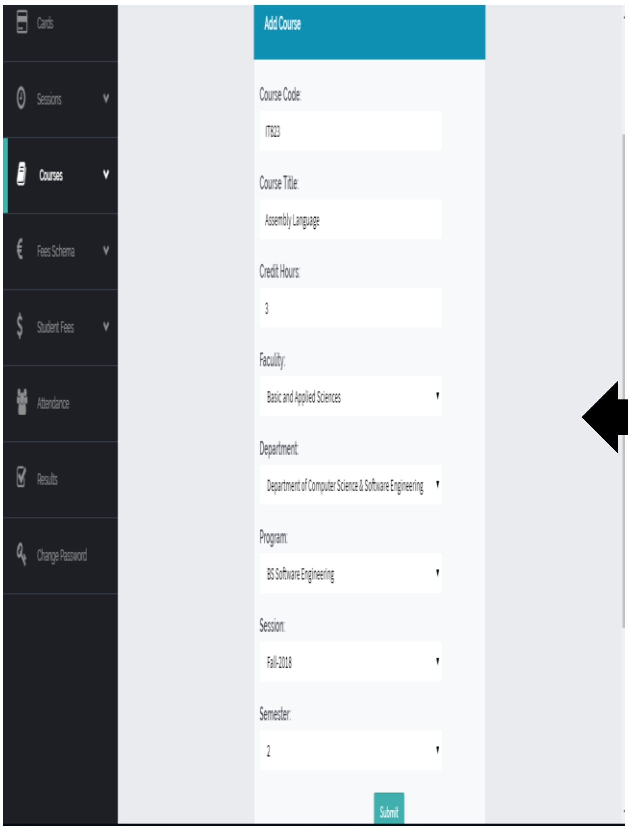
Create Session: Fall-2018

Course Registration Due Date: 26/11/2018

Fees Payment Due Date: 01/12/2018

2nd Installment Due Date: (In Case of installment) 12/12/2018

Examination to be held in: Jan-2019



Add Course

Course Code: IT823

Course Title: Assembly Language

Credit Hours: 3

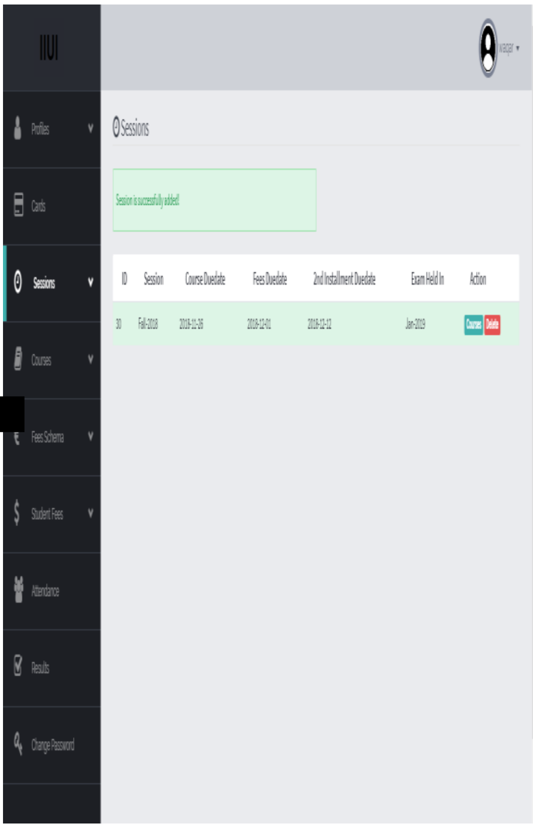
Faculty: Basic and Applied Sciences

Department: Department of Computer Science & Software Engineering

Program: BS Software Engineering

Session: Fall-2018

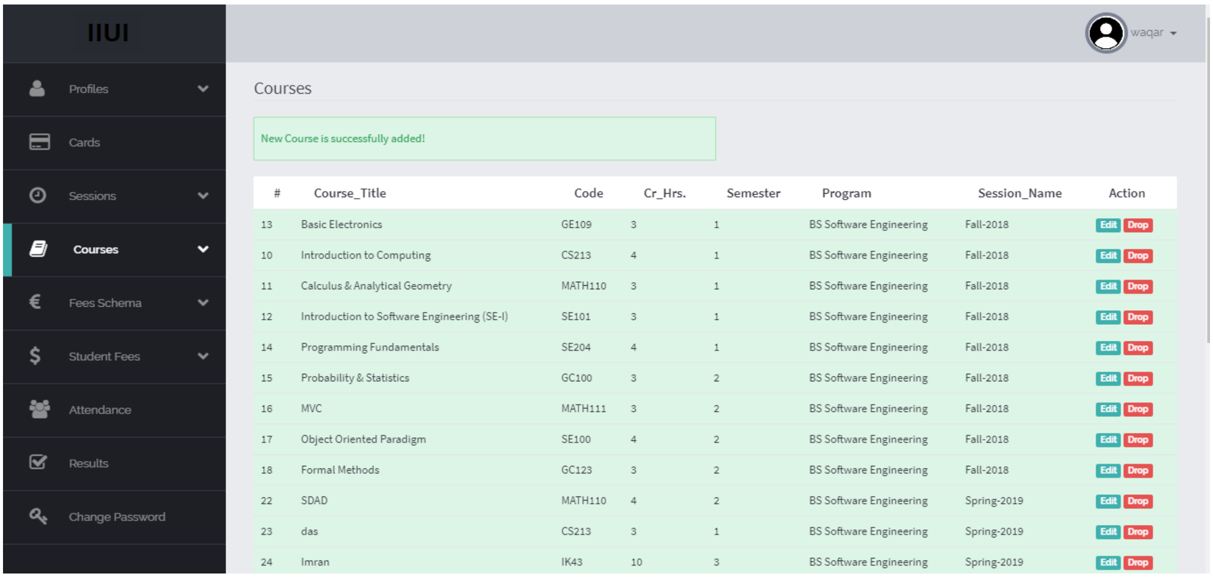
Semester: 2



Sessions

Session is successfully added!

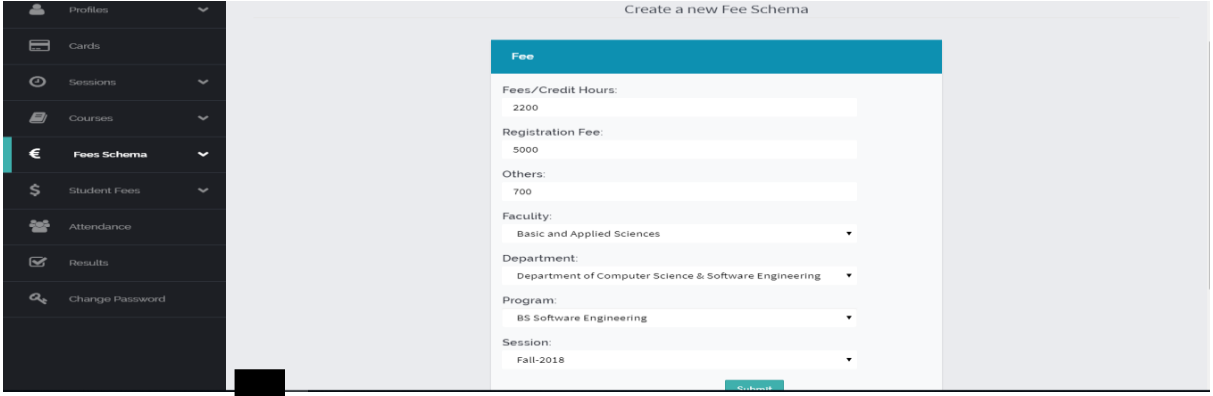
ID	Session	Course Duedate	Fees Duedate	2nd Installment Duedate	Exam Held In	Action
30	Fall-2018	2018-11-26	2018-12-01	2018-12-12	Jan-2019	Courses Delete



Courses

New Course is successfully added!

#	Course_Title	Code	Cr_Hrs.	Semester	Program	Session_Name	Action
13	Basic Electronics	GE109	3	1	BS Software Engineering	Fall-2018	Edit Drop
10	Introduction to Computing	CS213	4	1	BS Software Engineering	Fall-2018	Edit Drop
11	Calculus & Analytical Geometry	MATH110	3	1	BS Software Engineering	Fall-2018	Edit Drop
12	Introduction to Software Engineering (SE-I)	SE101	3	1	BS Software Engineering	Fall-2018	Edit Drop
14	Programming Fundamentals	SE204	4	1	BS Software Engineering	Fall-2018	Edit Drop
15	Probability & Statistics	GC100	3	2	BS Software Engineering	Fall-2018	Edit Drop
16	MVC	MATH111	3	2	BS Software Engineering	Fall-2018	Edit Drop
17	Object Oriented Paradigm	SE100	4	2	BS Software Engineering	Fall-2018	Edit Drop
18	Formal Methods	GC123	3	2	BS Software Engineering	Fall-2018	Edit Drop
22	SDAD	MATH110	4	2	BS Software Engineering	Spring-2019	Edit Drop
23	das	CS213	3	1	BS Software Engineering	Spring-2019	Edit Drop
24	Imran	IK43	10	3	BS Software Engineering	Spring-2019	Edit Drop



Create a new Fee Schema

Fees/Credit Hours: 2200

Registration Fee: 5000

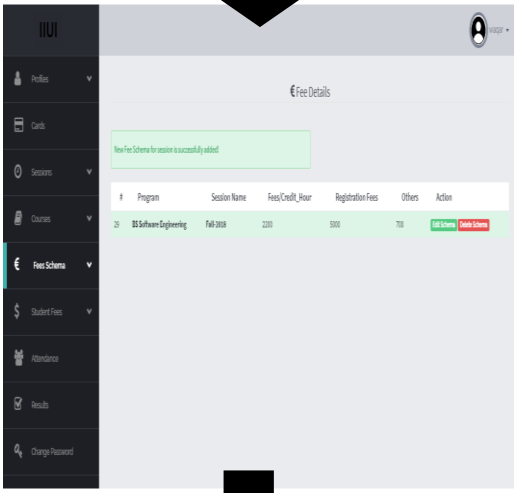
Others: 700

Faculty: Basic and Applied Sciences

Department: Department of Computer Science & Software Engineering

Program: BS Software Engineering

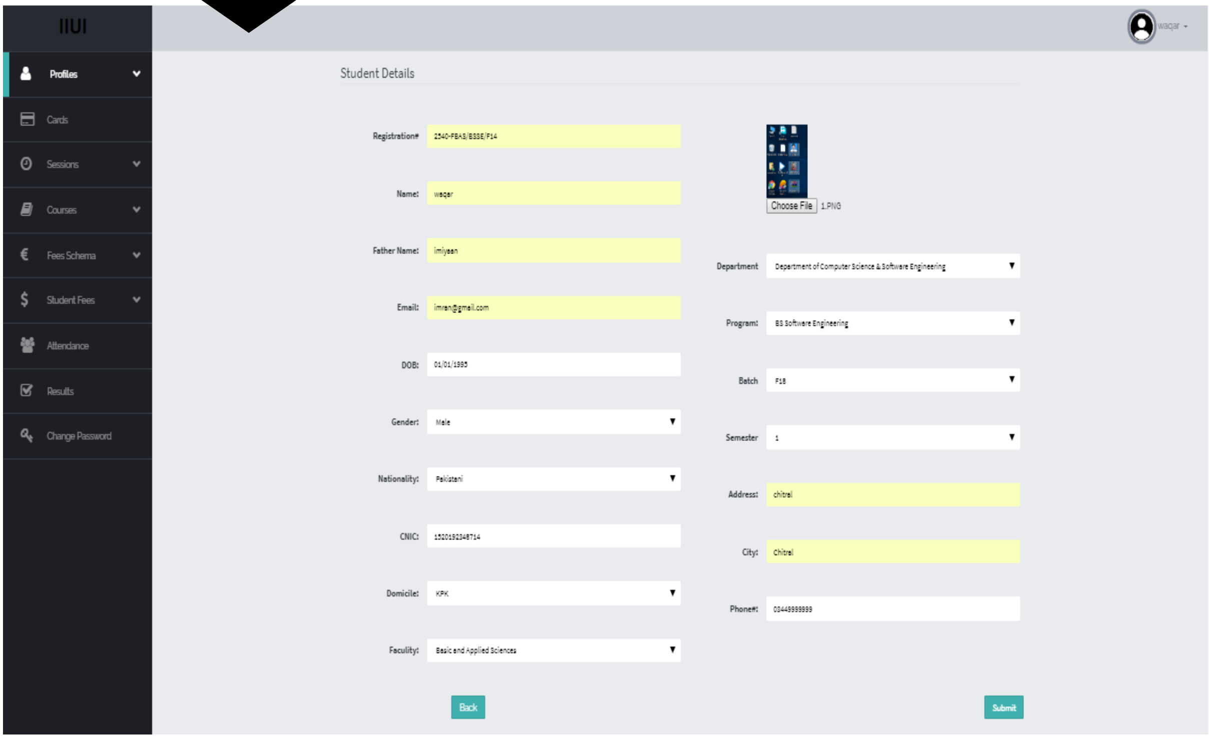
Session: Fall-2018



€ Fee Details

New Fee Schema for session is successfully added!

#	Program	Session Name	Fees/Credit_Hour	Registration Fees	Others	Action
29	BS Software Engineering	Fall-2018	2200	5000	700	Edit Schema Delete Schema



Student Details

Registration# 2540-FBAS/BSSE/F14

Name: waqar

Father Name: imiyaan

Email: imran@gmail.com

DOB: 01/01/1995

Gender: Male

Nationality: Pakistani

CNIC: 1520192348714

Domicile: KPK

Faculty: Basic and Applied Sciences

Department Department of Computer Science & Software Engineering

Program: BS Software Engineering

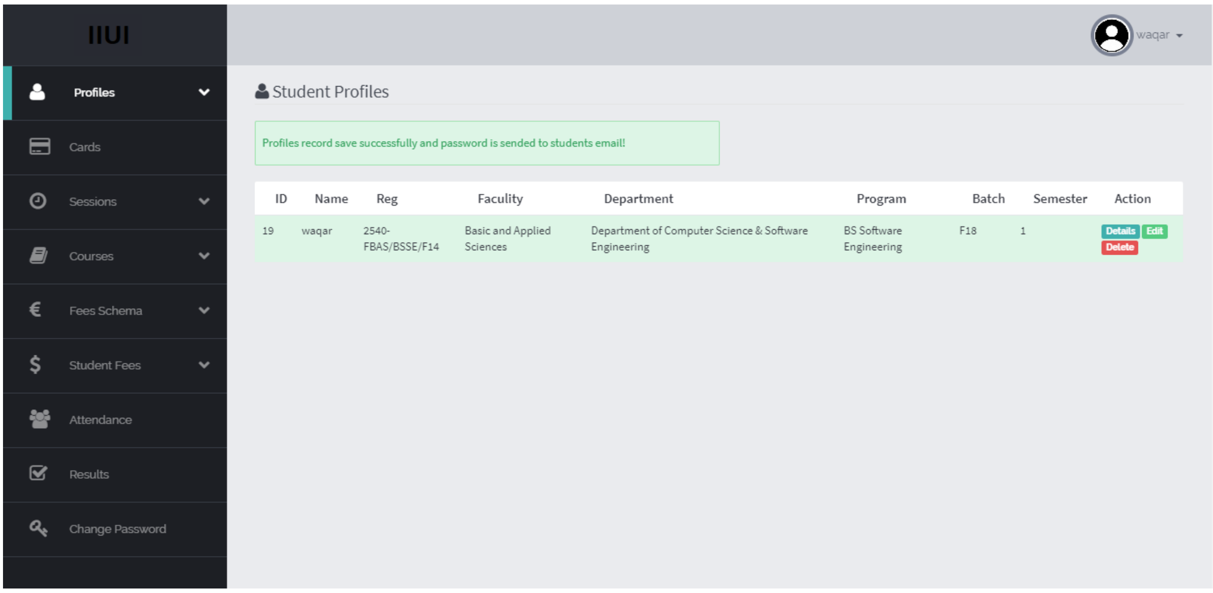
Batch F18

Semester 1

Address: chitral

City: Chitral

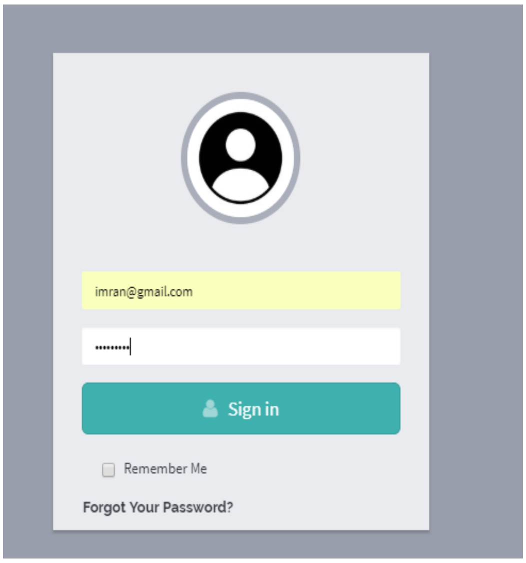
Phone#: 03449999999



Student Profiles

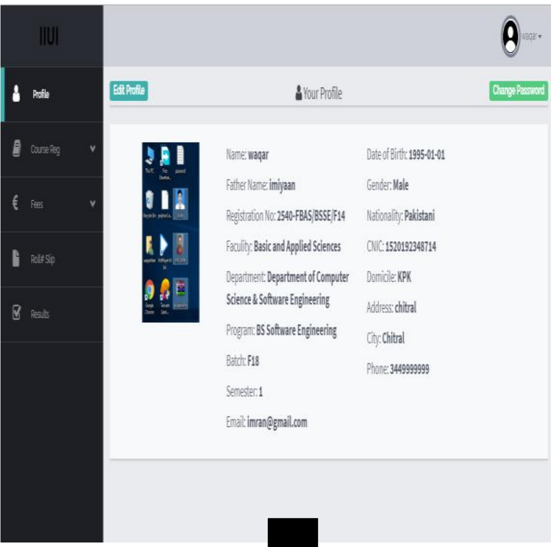
Profiles record save successfully and password is sended to students email!

ID	Name	Reg	Faculity	Department	Program	Batch	Semester	Action
19	waqar	2540-FBAS/BSSE/F14	Basic and Applied Sciences	Department of Computer Science & Software Engineering	BS Software Engineering	F18	1	Details Edit Delete



imran@gmail.com

•••••••••



Your Profile

Name: waqar

Father Name: imiyaan

Registration No: 2540-FBAS/BSSE/F14

Faculty: Basic and Applied Sciences

Department: Department of Computer Science & Software Engineering

Program: BS Software Engineering

Batch: F18

Semester: 1

Email: imran@gmail.com

Date of Birth: 1995-01-01

Gender: Male

Nationality: Pakistani

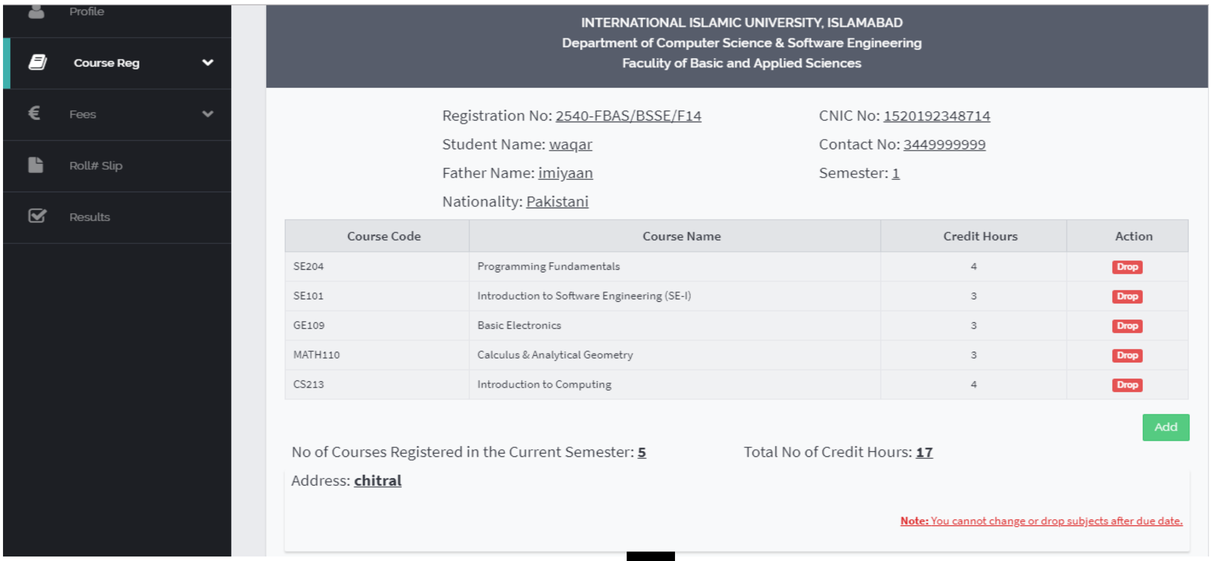
CNIC: 1520192348714

Domicile: KPK

Address: chitral

City: Chitral

Phone: 3449999999



INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD
Department of Computer Science & Software Engineering
Faculty of Basic and Applied Sciences

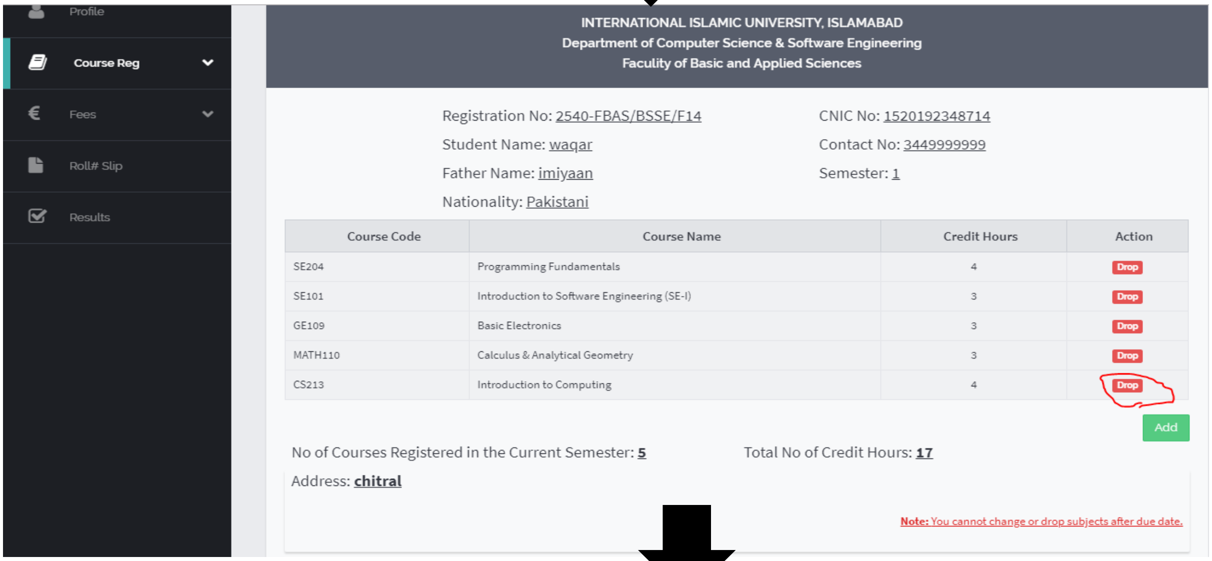
Registration No: 2540-FBAS/BSSE/F14 CNIC No: 1520192348714
Student Name: waqar Contact No: 3449999999
Father Name: imiyaan Semester: 1
Nationality: Pakistani

Course Code	Course Name	Credit Hours	Action
SE204	Programming Fundamentals	4	Drop
SE101	Introduction to Software Engineering (SE-I)	3	Drop
GE109	Basic Electronics	3	Drop
MATH110	Calculus & Analytical Geometry	3	Drop
CS213	Introduction to Computing	4	Drop

No of Courses Registered in the Current Semester: **5** Total No of Credit Hours: **17**

Address: **chitral**

Note: You cannot change or drop subjects after due date.



INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD
Department of Computer Science & Software Engineering
Faculty of Basic and Applied Sciences

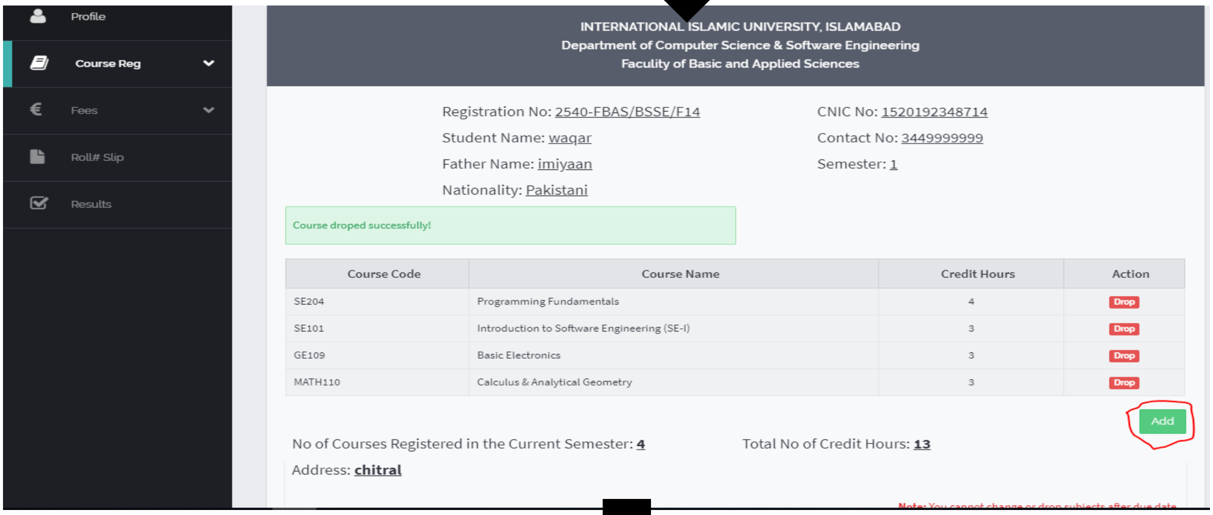
Registration No: 2540-FBAS/BSSE/F14 CNIC No: 1520192348714
Student Name: waqar Contact No: 3449999999
Father Name: imiyaan Semester: 1
Nationality: Pakistani

Course Code	Course Name	Credit Hours	Action
SE204	Programming Fundamentals	4	Drop
SE101	Introduction to Software Engineering (SE-I)	3	Drop
GE109	Basic Electronics	3	Drop
MATH110	Calculus & Analytical Geometry	3	Drop
CS213	Introduction to Computing	4	Drop

No of Courses Registered in the Current Semester: **5** Total No of Credit Hours: **17**

Address: **chitral**

Note: You cannot change or drop subjects after due date.



INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD
Department of Computer Science & Software Engineering
Faculty of Basic and Applied Sciences

Registration No: 2540-FBAS/BSSE/F14 CNIC No: 1520192348714
Student Name: waqar Contact No: 3449999999
Father Name: imiyaan Semester: 1
Nationality: Pakistani

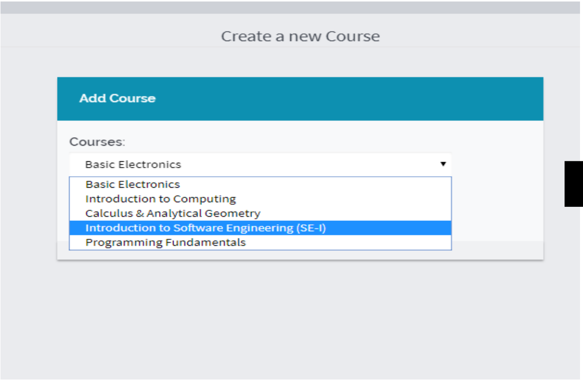
Course droped successfully!

Course Code	Course Name	Credit Hours	Action
SE204	Programming Fundamentals	4	Drop
SE101	Introduction to Software Engineering (SE-I)	3	Drop
GE109	Basic Electronics	3	Drop
MATH110	Calculus & Analytical Geometry	3	Drop

No of Courses Registered in the Current Semester: **4** Total No of Credit Hours: **13**

Address: **chitral**

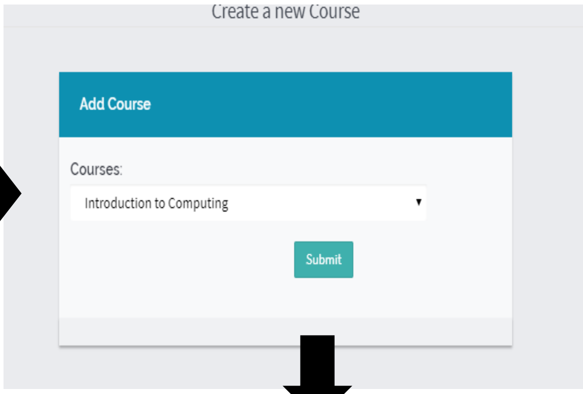
Note: You cannot change or drop subjects after due date.


Create a new Course

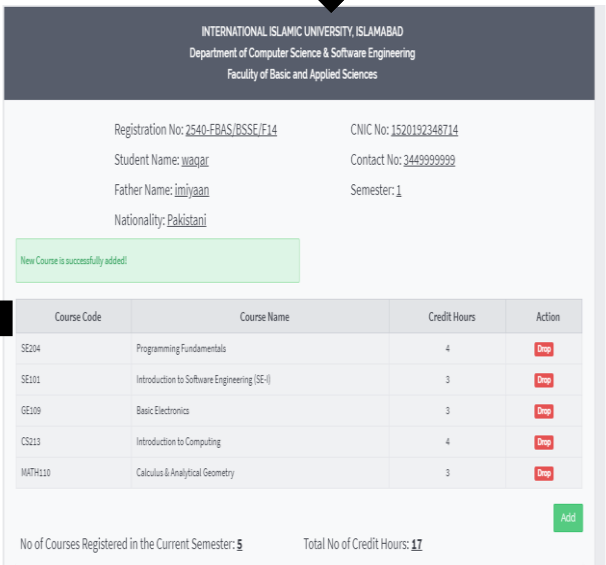
Courses:

Basic Electronics

- Basic Electronics
- Introduction to Computing
- Calculus & Analytical Geometry
- Introduction to Software Engineering (SE-I)
- Programming Fundamentals


Create a new Course

Courses: Introduction to Computing


INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD
Department of Computer Science & Software Engineering
Faculty of Basic and Applied Sciences

Registration No: 2540-FBAS/BSSE/F14 CNIC No: 1520192348714

Student Name: waqar Contact No: 3449999999

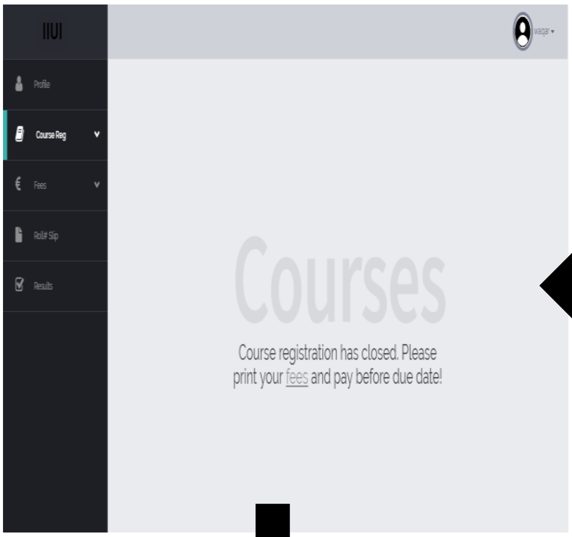
Father Name: imiyaan Semester: 1

Nationality: Pakistani

New Course is successfully added!

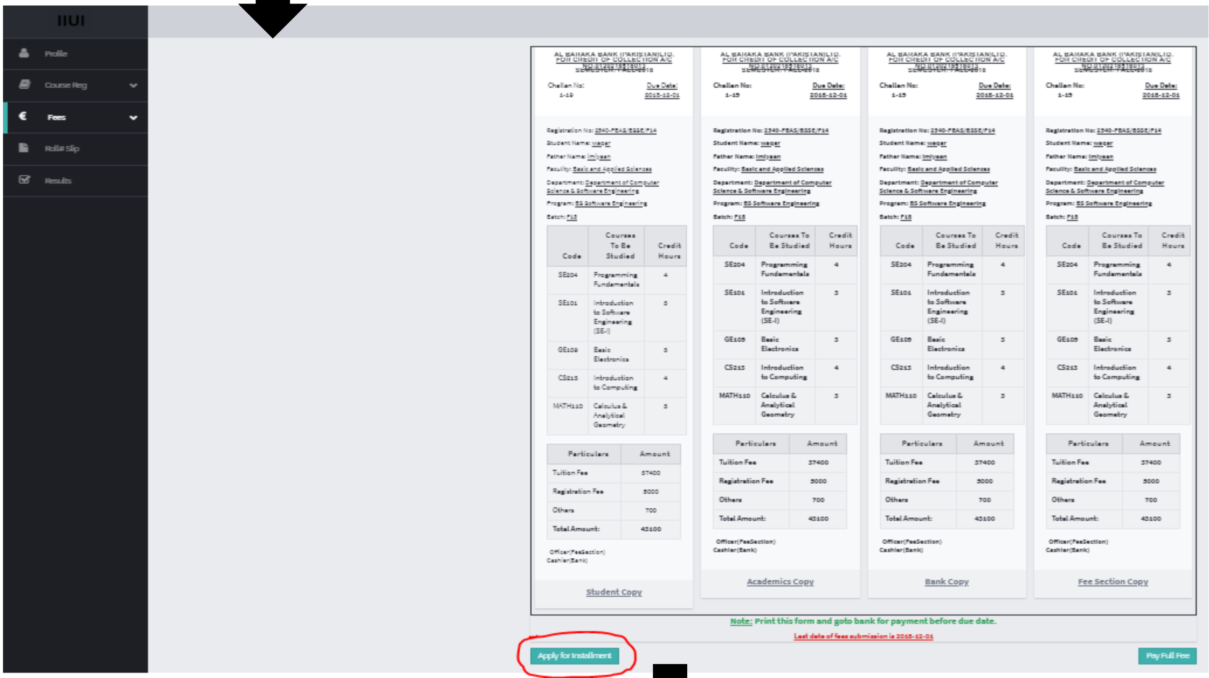
Course Code	Course Name	Credit Hours	Action
SE204	Programming Fundamentals	4	Drop
SE101	Introduction to Software Engineering (SE-I)	3	Drop
GE109	Basic Electronics	3	Drop
CS213	Introduction to Computing	4	Drop
MATH110	Calculus & Analytical Geometry	3	Drop

No of Courses Registered in the Current Semester: 5 Total No of Credit Hours: 17



Courses

Course registration has closed. Please print your fees and pay before due date!



AL BARAKA BANK (PAKISTAN)LTD.
FOR CREDIT OF COLLECTION A/C
NO [illegible]
SEMESTER: FALL-2018

Challan No: 1-19 Due Date: 2018-12-01

Registration No: 2540-FBAS/BSSE/F14
Student Name: waqar
Father Name: imiyaan
Faculty: Basic and Applied Sciences
Department: Department of Computer Science & Software Engineering
Program: BS Software Engineering
Batch: F18

Code	Courses To Be Studied	Credit Hours
SE204	Programming Fundamentals	4
SE101	Introduction to Software Engineering (SE-I)	3
GE109	Basic Electronics	3
CS213	Introduction to Computing	4
MATH110	Calculus & Analytical Geometry	3

Particulars	Amount
Tuition Fee	37400
Registration Fee	5000
Others	700
Total Amount:	43100

Officer(FeeSection)
Cashier(Bank)

Student Copy

AL BARAKA BANK (PAKISTAN)LTD.
FOR CREDIT OF COLLECTION A/C
NO [illegible]
SEMESTER: FALL-2018

Challan No: 1-19 Due Date: 2018-12-01

Registration No: 2540-FBAS/BSSE/F14
Student Name: waqar
Father Name: imiyaan
Faculty: Basic and Applied Sciences
Department: Department of Computer Science & Software Engineering
Program: BS Software Engineering
Batch: F18

Code	Courses To Be Studied	Credit Hours
SE204	Programming Fundamentals	4
SE101	Introduction to Software Engineering (SE-I)	3
GE109	Basic Electronics	3
CS213	Introduction to Computing	4
MATH110	Calculus & Analytical Geometry	3

Particulars	Amount
Tuition Fee	37400
Registration Fee	5000
Others	700
Total Amount:	43100

Officer(FeeSection)
Cashier(Bank)

Academics Copy

AL BARAKA BANK (PAKISTAN)LTD.
FOR CREDIT OF COLLECTION A/C
NO [illegible]
SEMESTER: FALL-2018

Challan No: 1-19 Due Date: 2018-12-01

Registration No: 2540-FBAS/BSSE/F14
Student Name: waqar
Father Name: imiyaan
Faculty: Basic and Applied Sciences
Department: Department of Computer Science & Software Engineering
Program: BS Software Engineering
Batch: F18

Code	Courses To Be Studied	Credit Hours
SE204	Programming Fundamentals	4
SE101	Introduction to Software Engineering (SE-I)	3
GE109	Basic Electronics	3
CS213	Introduction to Computing	4
MATH110	Calculus & Analytical Geometry	3

Particulars	Amount
Tuition Fee	37400
Registration Fee	5000
Others	700
Total Amount:	43100

Officer(FeeSection)
Cashier(Bank)

Bank Copy

AL BARAKA BANK (PAKISTAN)LTD.
FOR CREDIT OF COLLECTION A/C
NO [illegible]
SEMESTER: FALL-2018

Challan No: 1-19 Due Date: 2018-12-01

Registration No: 2540-FBAS/BSSE/F14
Student Name: waqar
Father Name: imiyaan
Faculty: Basic and Applied Sciences
Department: Department of Computer Science & Software Engineering
Program: BS Software Engineering
Batch: F18

Code	Courses To Be Studied	Credit Hours
SE204	Programming Fundamentals	4
SE101	Introduction to Software Engineering (SE-I)	3
GE109	Basic Electronics	3
CS213	Introduction to Computing	4
MATH110	Calculus & Analytical Geometry	3

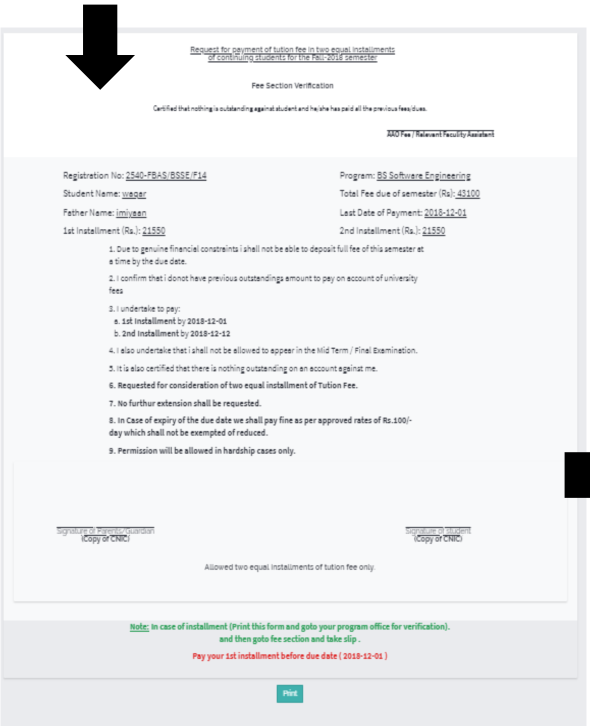
Particulars	Amount
Tuition Fee	37400
Registration Fee	5000
Others	700
Total Amount:	43100

Officer(FeeSection)
Cashier(Bank)

Fee Section Copy

Note: Print this form and goto bank for payment before due date.

Last date of fees submission is 2018-12-01



Request for payment of tution fee in two equal installments of continuing students for the Fall-2018 semester

Fee Section Verification

Certified that nothing is outstanding against student and he/she has paid all the previous fees/dues.

AAO Fee / Relevant Faculty Assistant

Registration No: 2540-FBAS/BSSE/F14
Program: BS Software Engineering
Student Name: waqar
Total Fee due of semester (Rs): 43100
Father Name: imiyaan
Last Date of Payment: 2018-12-01
1st Installment (Rs.): 21550
2nd Installment (Rs.): 21550

1. Due to genuine financial constraints i shall not be able to deposit full fee of this semester at a time by the due date.
2. I confirm that i donot have previous outstandings amount to pay on account of university fees
3. I undertake to pay:
   a. **1st Installment** by **2018-12-01**
   b. **2nd Installment** by **2018-12-12**
4. I also undertake that i shall not be allowed to appear in the Mid Term / Final Examination.
5. It is also certified that there is nothing outstanding on an account against me.
6. **Requested for consideration of two equal installment of Tution Fee.**
7. **No furthur extension shall be requested.**
8. **In Case of expiry of the due date we shall pay fine as per approved rates of Rs.100/- day which shall not be exempted of reduced.**
9. **Permission will be allowed in hardship cases only.**

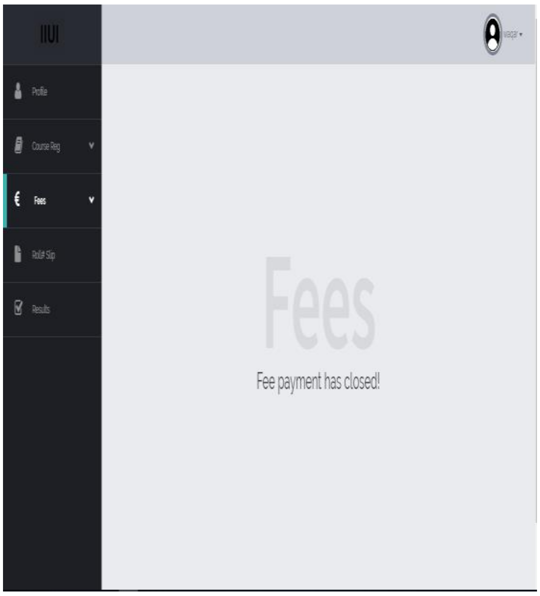
Signature of Parents/Guardian
(Copy of CNIC)

Signature of student
(Copy of CNIC)

Allowed two equal installments of tution fee only.

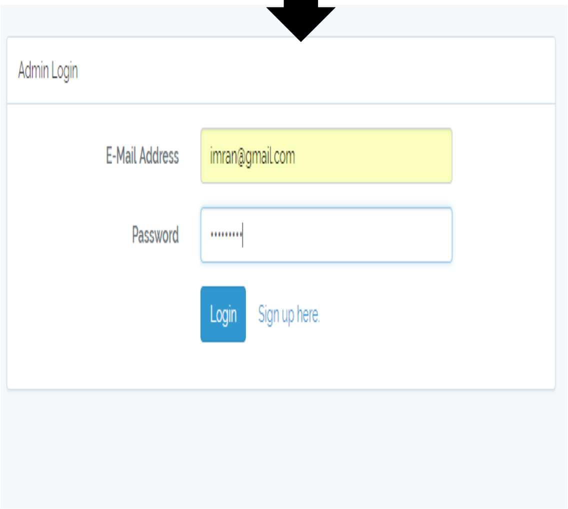
Note: In case of installment (Print this form and goto your program office for verification). and then goto fee section and take slip .

Pay your 1st installment before due date (2018-12-01)



Fees

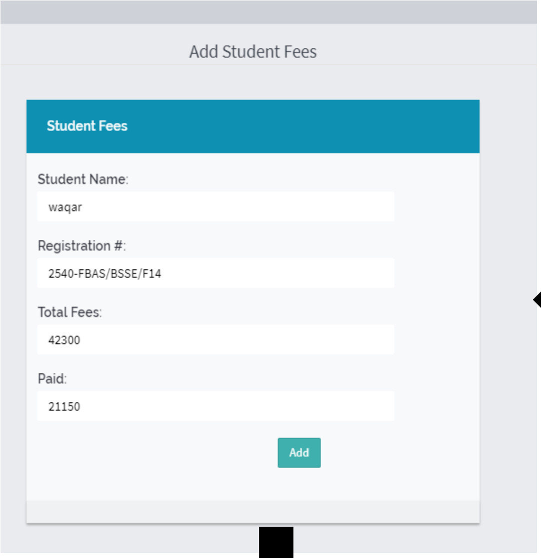
Fee payment has closed!



Admin Login

E-Mail Address: imran@gmail.com

Password: ••••••••



Add Student Fees

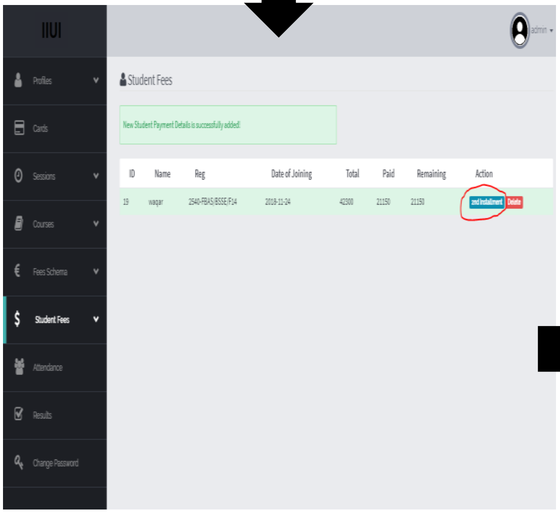
Student Fees

Student Name: waqar

Registration #: 2540-FBAS/BSSE/F14

Total Fees: 42300

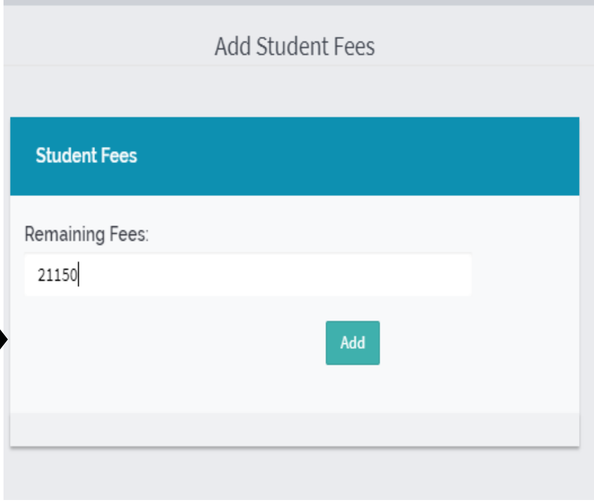
Paid: 21150



Student Fees

New Student Payment Details is successfully added!

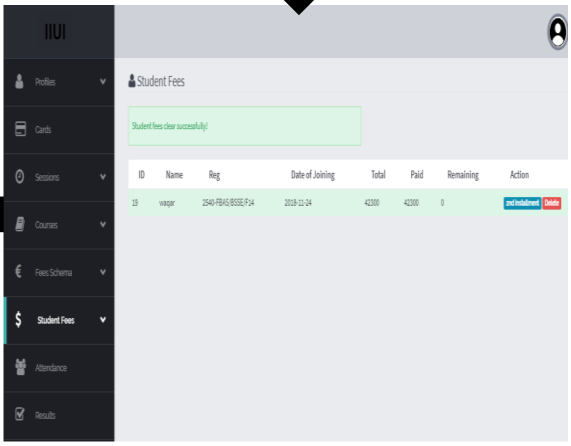
ID	Name	Reg	Date of Joining	Total	Paid	Remaining	Action
19	waqar	2540-FBAS/BSSE/F14	2018-11-24	42300	21150	21150	2nd Installment Delete



Add Student Fees

Student Fees

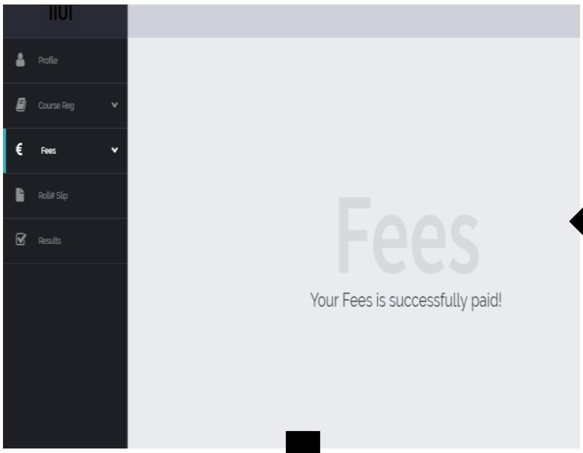
Remaining Fees: 21150



Student Fees

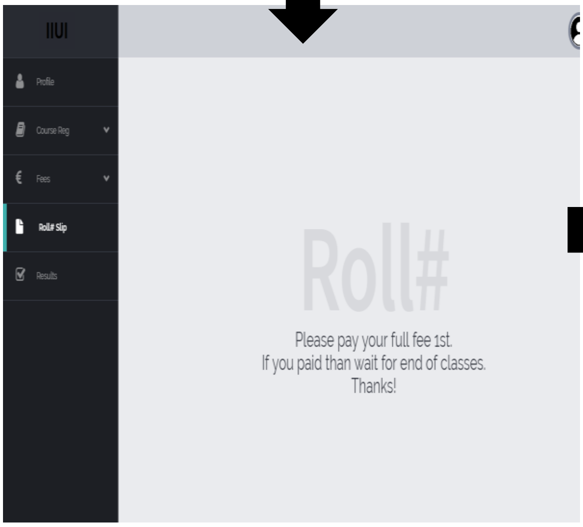
Student fees clear successfully!

ID	Name	Reg	Date of Joining	Total	Paid	Remaining	Action
19	waqar	2540-FBAS/BSSE/F14	2018-11-24	42300	42300	0	2nd Installment Delete



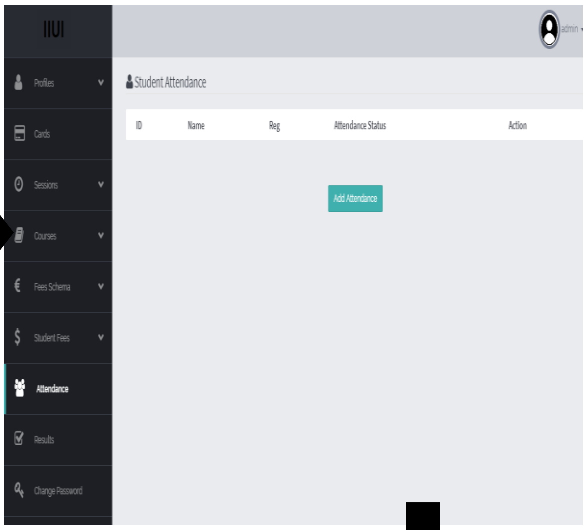
Fees

Your Fees is successfully paid!



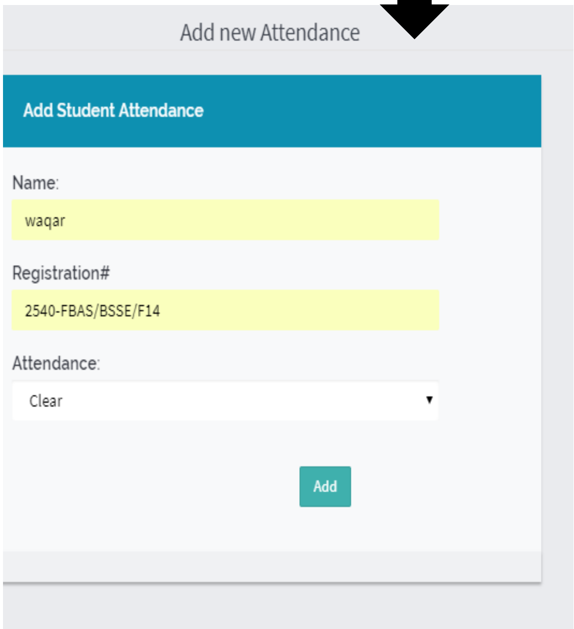
Roll#

Please pay your full fee 1st.
If you paid than wait for end of classes.
Thanks!



Student Attendance

ID	Name	Reg	Attendance Status	Action



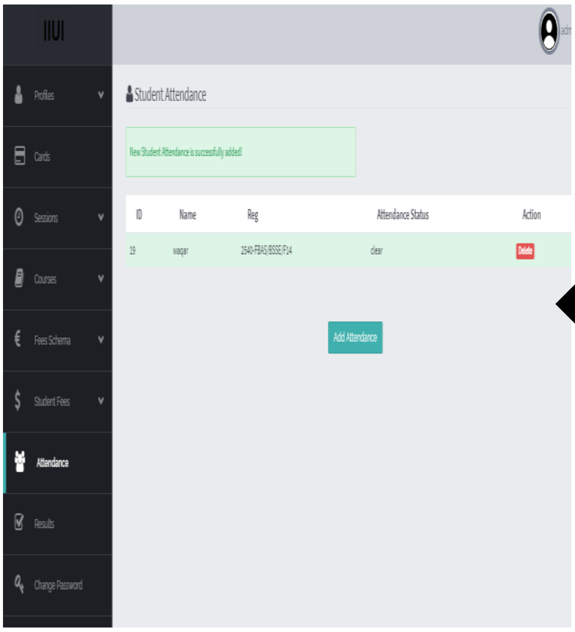
Add new Attendance

Add Student Attendance

Name: waqar

Registration#: 2540-FBAS/BSSE/F14

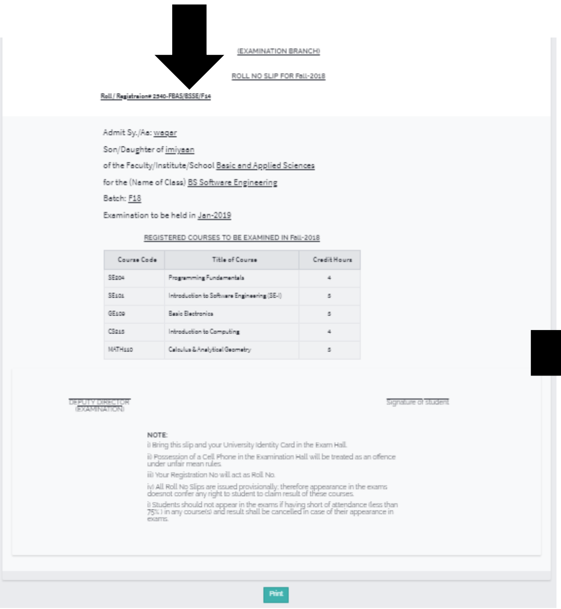
Attendance: Clear



Student Attendance

New Student Attendance is successfully added!

ID	Name	Reg	Attendance Status	Action
19	waqar	2540-FBAS/BSSE/F14	clear	Delete



(EXAMINATION BRANCH)

ROLL NO SLIP FOR Fall-2018

Roll / Registraion# 2940-FBAS/BSSE/F14

Admit Sy./Aa: waqar

Son/Daughter of imiyaan

of the Faculty/Institute/School Basic and Applied Sciences

for the (Name of Class) BS Software Engineering

Batch: F18

Examination to be held in Jan-2019

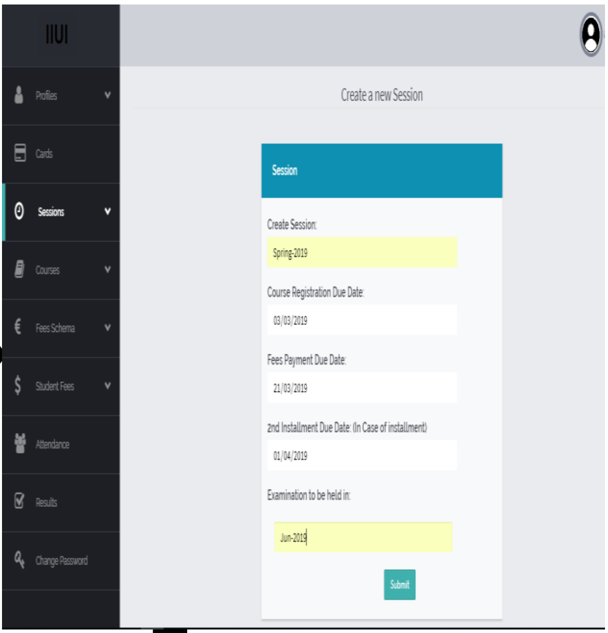
REGISTERED COURSES TO BE EXAMINED IN Fall-2018

Course Code	Title of Course	Credit Hours
SE204	Programming Fundamentals	4
SE101	Introduction to Software Engineering (SE-I)	5
GE109	Basic Electronics	5
CS215	Introduction to Computing	4
MATH110	Calculus & Analytical Geometry	5

DEPUTY DIRECTOR
(EXAMINATION)

Signature of student

NOTE:
i) Bring this slip and your University Identity Card in the Exam Hall.
ii) Possession of a Cell Phone in the Examination Hall will be treated as an offence under unfair mean rules.
iii) Your Registration No will act as Roll No.
iv) All Roll No Slips are issued provisionally; therefore appearance in the exams doesnot confer any right to student to claim result of these courses.
i) Students should not appear in the exams if having short of attendance (less than 75%) in any course(s) and result shall be cancelled in case of their appearance in exams.



Create a new Session

Session

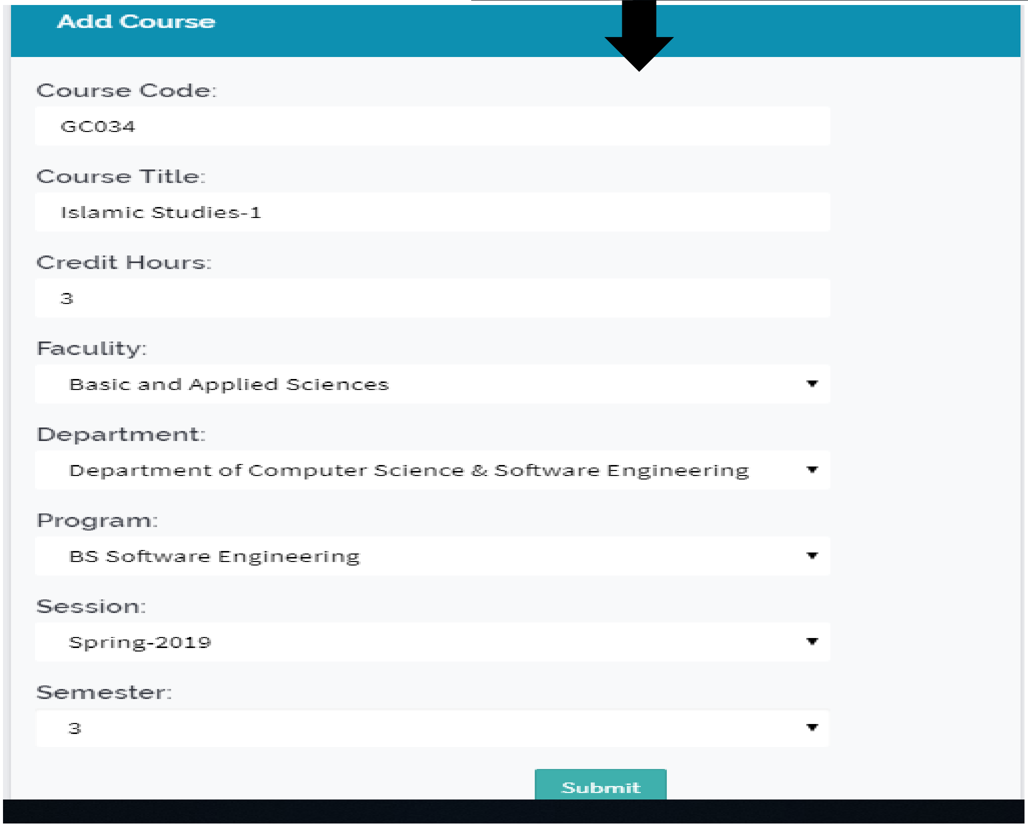
Create Session: Spring-2019

Course Registration Due Date: 03/03/2019

Fees Payment Due Date: 21/03/2019

2nd Installment Due Date: (In Case of installment) 01/04/2019

Examination to be held in: Jun-2019



Add Course

Course Code: GC034

Course Title: Islamic Studies-1

Credit Hours: 3

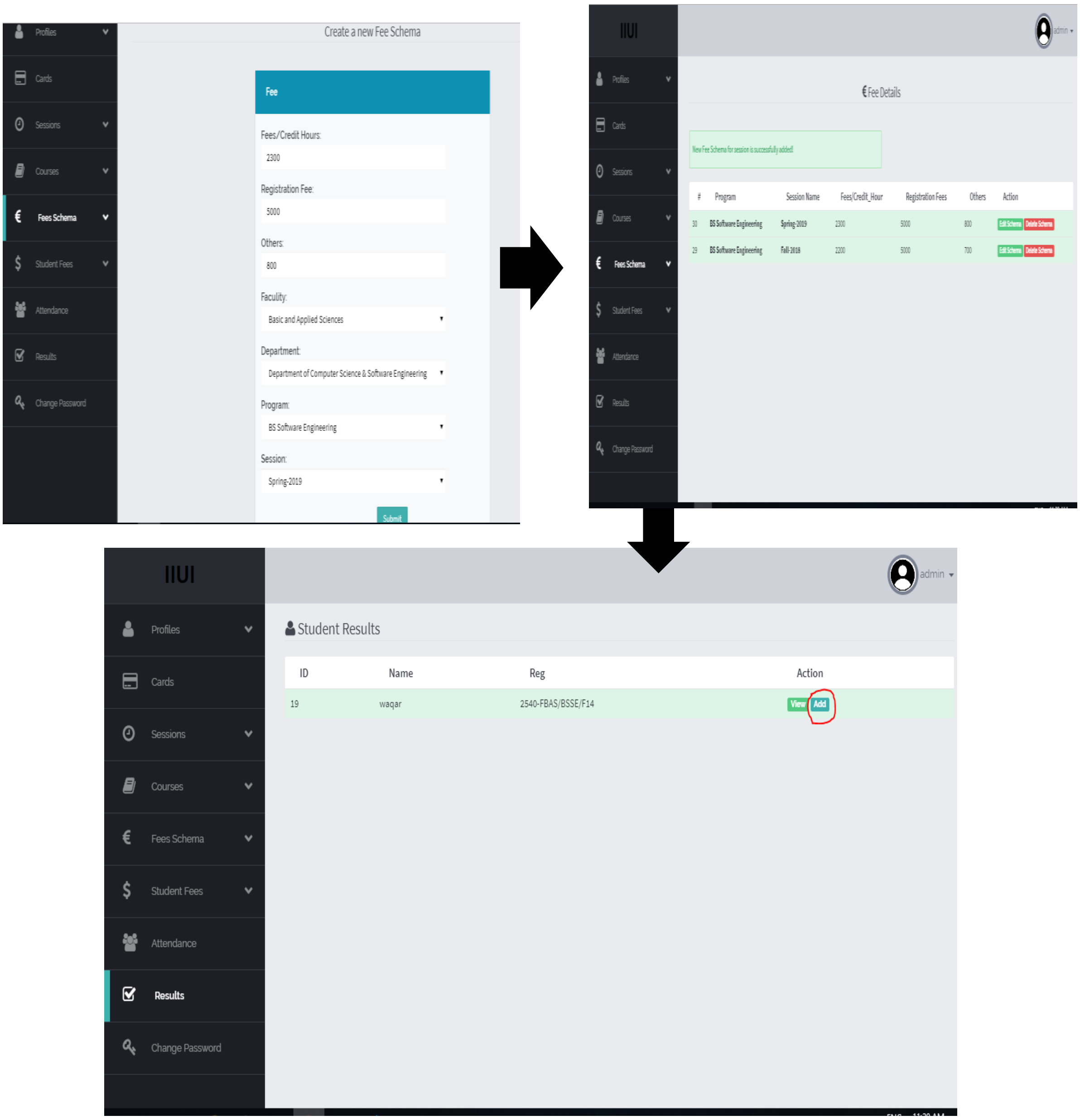
Faculty: Basic and Applied Sciences

Department: Department of Computer Science & Software Engineering

Program: BS Software Engineering

Session: Spring-2019

Semester: 3



Create a new Fee Schema

Fee

Fees/Credit Hours: 2300

Registration Fee: 5000

Others: 800

Faculty: Basic and Applied Sciences

Department: Department of Computer Science & Software Engineering

Program: BS Software Engineering

Session: Spring-2019

Submit

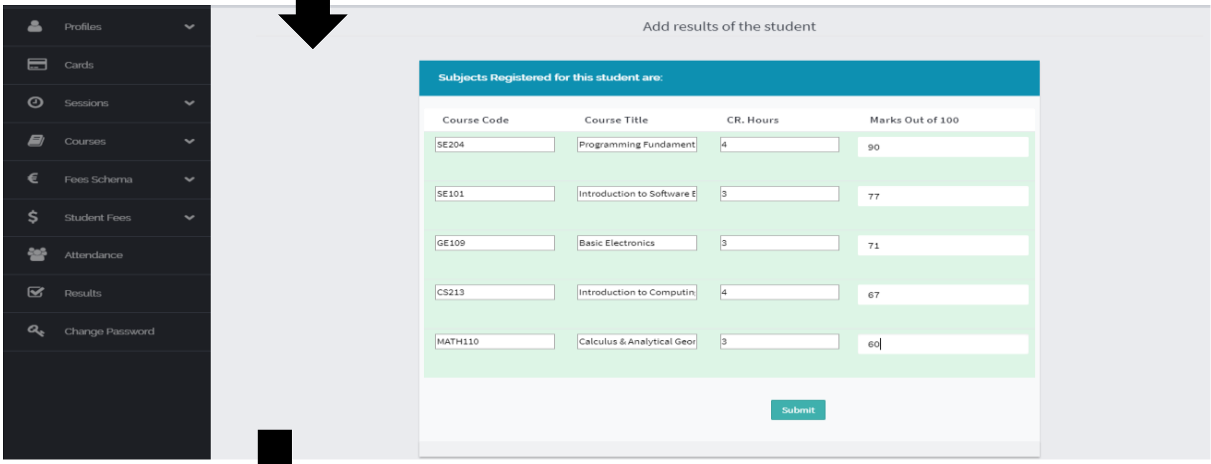
€ Fee Details

New Fee Schema for session is successfully added!

#	Program	Session Name	Fees/Credit_Hour	Registration Fees	Others	Action
30	BS Software Engineering	Spring-2019	2300	5000	800	Edit Schema Delete Schema
29	BS Software Engineering	Fall-2018	2200	5000	700	Edit Schema Delete Schema

Student Results

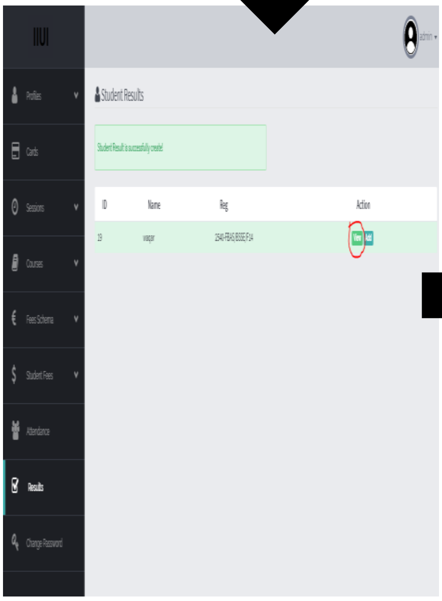
ID	Name	Reg	Action
19	waqar	2540-FBAS/BSSE/F14	View Add



Add results of the student

Subjects Registered for this student are:

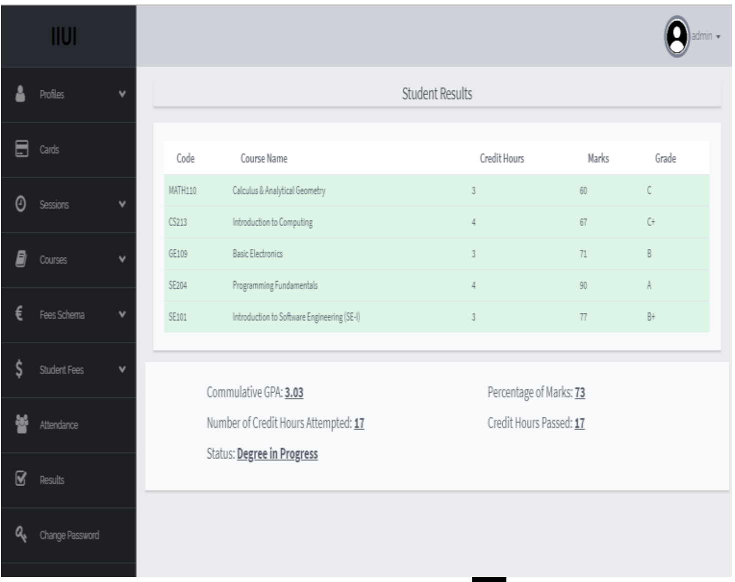
Course Code	Course Title	CR. Hours	Marks Out of 100
SE204	Programming Fundament	4	90
SE101	Introduction to Software E	3	77
GE109	Basic Electronics	3	71
CS213	Introduction to Computin	4	67
MATH110	Calculus & Analytical Geor	3	60



Student Results

Student Result is successfully create!

ID	Name	Reg	Action
19	waqar	2540-FBAS/BSSE/F14	View Add



Student Results

Code	Course Name	Credit Hours	Marks	Grade
MATH110	Calculus & Analytical Geometry	3	60	C
CS213	Introduction to Computing	4	67	C+
GE109	Basic Electronics	3	71	B
SE204	Programming Fundamentals	4	90	A
SE101	Introduction to Software Engineering (SE-I)	3	77	B+

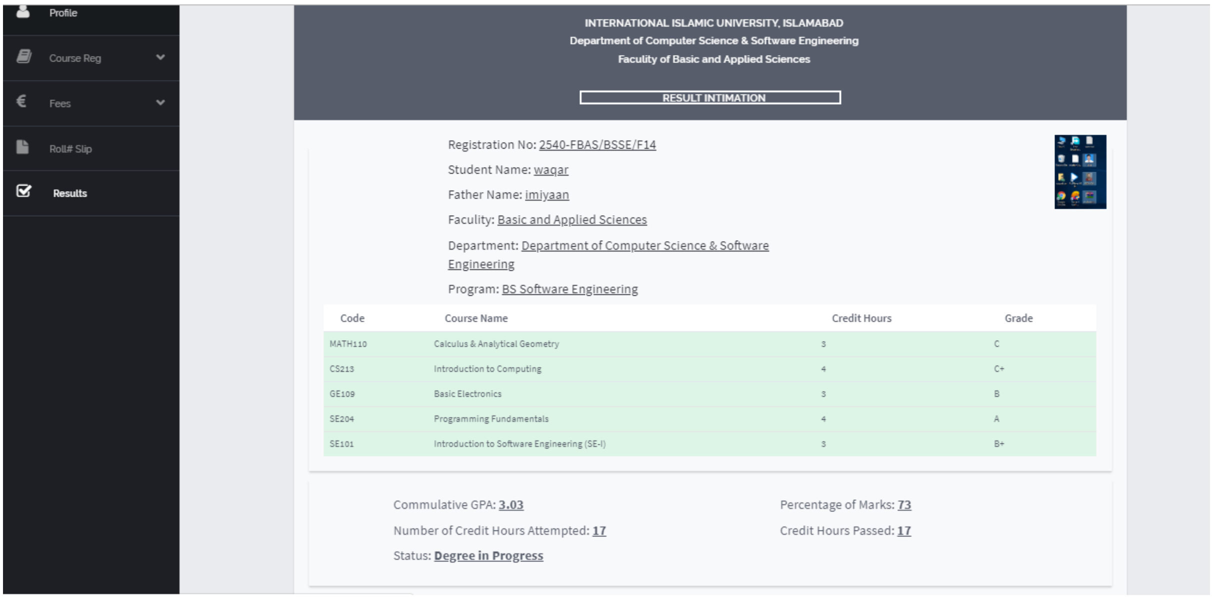
Commulative GPA: 3.03

Percentage of Marks: 73

Number of Credit Hours Attempted: 17

Credit Hours Passed: 17

Status: Degree in Progress



INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD
Department of Computer Science & Software Engineering
Faculty of Basic and Applied Sciences

RESULT INTIMATION

Registration No: 2540-FBAS/BSSE/F14

Student Name: waqar

Father Name: imiyaan

Faculty: Basic and Applied Sciences

Department: Department of Computer Science & Software Engineering

Program: BS Software Engineering

Code	Course Name	Credit Hours	Grade
MATH110	Calculus & Analytical Geometry	3	C
CS213	Introduction to Computing	4	C+
GE109	Basic Electronics	3	B
SE204	Programming Fundamentals	4	A
SE101	Introduction to Software Engineering (SE-I)	3	B+

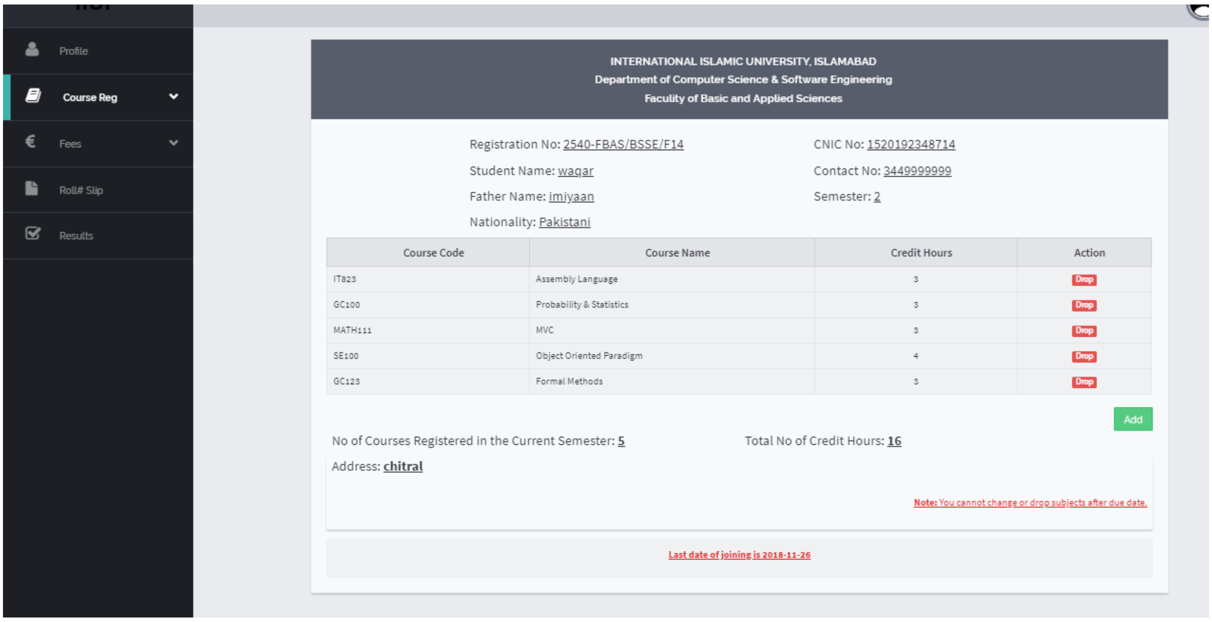
Commulative GPA: 3.03

Percentage of Marks: 73

Number of Credit Hours Attempted: 17

Credit Hours Passed: 17

Status: Degree in Progress



INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD
Department of Computer Science & Software Engineering
Faculty of Basic and Applied Sciences

Registration No: 2540-FBAS/BSSE/F14 CNIC No: 1520192348714

Student Name: waqar Contact No: 3449999999

Father Name: imiyaan Semester: 2

Nationality: Pakistani

Course Code	Course Name	Credit Hours	Action
IT823	Assembly Language	3	Drop
GC100	Probability & Statistics	3	Drop
MATH111	MVC	3	Drop
SE100	Object Oriented Paradigm	4	Drop
GC123	Formal Methods	3	Drop

No of Courses Registered in the Current Semester: 5 Total No of Credit Hours: 16

Address: chitral

Note: You cannot change or drop subjects after due date.

Last date of joining is 2018-11-26

APPENDIX B

REFERENCES

References:

Websites

creately (2018). creatly Retrieved 03 OCT, 2018, from

https://creately.com/app/?tempID=gc7qvpsj1&login_type=demo#

yuml (2018). yuml Retrieved 04 OCT, 2018, from

https://yuml.me/diagram/nofunky/activity/samples

lucidchart (2018). lucidchart Retrieved 11OCT, 2018, from

https://www.lucidchart.com/users/trialIntro/3

gliffy (2018). gliffy Retrieved 01 NOV, 2018, from

https://www.gliffy.com/go/html5/12122432

Books

[1] Larman, Craig, 1990. Appling UML and Patterns. 4th ed. Valtech Pearson.