CSE-190009

Student Information System

A project report submitted to the department of Computer Science & Engineering of Sonargaon University (SU) in fulfillment of the requirements for the degree of Bachelor of Science in Computer Science & Engineering

Sonargaon University (SU)

Submitted By

Md Uzzal Hossain

ID: CSE1503006012

with the cooperation

Md Saiful Islam Arif

ID: CSE1503006002

Tareque Islam Choudhury

ID: CSE1503006011

Tinni Rahman

ID: CSE1503006007

Farha Akter

ID: CSE1503005002

Supervised By

Sadia Tasnim Barsha

Lecturer

B.Sc. in Computer Science & Engineering

Sonargaon University

Date of Submission: 17 May, 2019

Sonargaon University (SU)
Department of Computer Science & Engineering

LETTER OF TRANSMITTAL

May 17, 2019

To

Sadia Tasnim Barsha

Lecturer

Department of Computer Science & Engineering

Sonargaon University (SU)

147/I, Green Road, Tejgaon, Dhaka-1215, Bangladesh.

Subject: Submission of Project Report.

Dear Madam,

This is the great pleasure that we are submitting here with the project of "student information system" It is an important topic. The project report has been done according to the requirement and guidelines of Sonargaon University (SU).

We hope that this report will certainly help you in evaluation our project report on "student information system" We Would be very glad to provide any assistance in interpreting any part of the paper, whenever necessary.

Thanking you

Md Uzzal Hossain

CSE1503006012

Group Leader

Sonargaon University (SU)
Department of Computer Science & Engineering

CERTIFICATION

This is to certify that the project paper on "Student Information System" is the confide record of project work done by Md Uzzal Hossain (Student ID: CSE1503006012) as a group Leader and others for partial fulfillment of the requirement of the degree of B.Sc. in Computer Science & Engineering from the Sonargaon University (SU).

This project work has been carried out under my guidance and is a record of successful work.

Supervisor

.....

Sadia Tasnim Barsha

Lecturer

Department of Computer Science & Engineering

Sonargaon University (SU)

147/I, Green Road, Tejgaon, Dhaka-1215, Bangladesh.

Sonargaon University (SU)
Department of Computer Science & Engineering

DECLARATION

This is to declare that the work and material presented in the report has been carried out by us and has not previously been submitted to any University/College/Organization for any academic qualification.

We thereby ensure that the work that has been presented does not breach existing copyright. We undertake to identify the university against any loss or damage arising from breach of the foregoing obligation.

Thanking You
Session 6B

1.....
Md Uzzal Hossain

CSE1503006012

2.....
Md Saiful Islam Arif

CSE1503006002

3.....
Tareque Islam Choudhury

CSE1503006011

4.....
Tinni Rahman

CSE1503006007

5.....
Farha Akter

CSE1503005002

Sonargaon University (SU)
Department of Computer Science & Engineering

LETTER OF ACCEPTANCE

This Project entitled "Student Information System" submitted By Md. Uzzal Hossain ID: CSE1503006012, Md. Saiful Islam Arif ID: CSE1503006002, Tareque Islam chowdhury ID: CSE1503006011, Farha Akter ID: CSE1503005002, Tinni Rahman ID: CSE1503006007 to the Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh has been accepted as satisfactory for the partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering and approved as to its style and contents.

Board of Examiners

.....	.....
Examiner 1	Examiner 2
.....	.....
Examiner 3	Examiner 4
.....	.....
Examiner 5	Examiner 6

Sonargaon University (SU)

Department of Computer Science & Engineering

ACKNOWLEDGMENT

All praises and profound gratitude to the Almighty Allah who is the most beneficent and the most merciful for allowing great opportunity and ability to bring this effort to fruition safety and peacefully. We sincerely acknowledge and express my deep sense of gratitude to **Sadia Tasnim Barsha** (Lecturer) the guide of this project. As a guide, he gave maximum help and coordination in finishing the project work. With his past years of experience and teaching steered us to come out with success through the most difficult problems faced by us. His active interest in this topic and valuable advice was the source of the author's inspiration. We would like to place on record our deep sense of gratitude to our guides for their cooperation and unfailing courtesy to us at every stage.

We sincerely would like to thank all instructors and staffs of the Computer Science & Engineering Department of Sonargaon University (SU), Dhaka, which contributed in various ways to the completion of this project. Finally, we would like to express our deepest gratitude to our entire group member whose support and manual labor contributed in various ways for the completion of this thesis work.

Authors

Abstract

An organized and systematic solution is essential for all universities. There are many departments of administration for the maintenance of university information and student databases in any institution. All these departments provide various records regarding students and necessary information for a student. Most of these track records need to maintain information about the students and faculty. This information could be the general details like student, teacher, notice, result, feedback and specific information related to departments like collection of data. All the modules in university administration are interdependent. They are maintained manually. So, they need to be automated and centralized. For example, when a student needs any important lectures and education related document, this system able to provide. So, it needs to contact all the modules that are office, department and examination and result of students. Administrator and faculty using the system will find that the process of recording and retrieving student's information and managing their classes and provide educational resources for students. In general, this project aims to enhance efficiency and at the same time maintain information accurateness. Later in this report, features and improvement that allow achievement to this goal will be demonstrated and highlighted. Our work is useful for easy user interface. We are planning to utilize the powerful database management, data retrieval and data manipulation. We will provide more ease for managing the data than manually maintaining in the documents. Our work is useful for saving valuable time and reduces the huge paper work.

Keywords: Education, Database managements, student's database, students' academic performance, Results. Notice, Document management, Feedback, Session wise and course wise lecture.

Contents

Table of Contents

List of figures

Chapter 1

Introduction

1.1 Introduction

Student Information System (SIS) deals with all kind of student details, academic related reports, course details, curriculum, batch details, session details, session and course wise lecture documents, teacher information and other resource related details too. It tracks all the details of a student from the day one to the end of his course which can be used for all reporting purpose, tracking of attendance, progress in the course, completed semesters years, coming semester year curriculum details, exam details, project or any other assignment details, important notices, feedback, final exam result etc.

The student Information system is an automated version of manual Student Information System.

Our design can facilitate us to explore all the activities happening in the college, even we can get to know which faculty is assigned to which course, the current status of a student, attendance percentage of a student and upcoming requirements of a student. The Student Information System is an automated version of manual Student Management System. It can handle all details about a student. The details include college details, subject details, student personnel details, academic details, exam details etc.

In case of manual system, they need a lot of time, manpower etc. Here almost all work is computerized. So, the accuracy is maintained. Maintaining backup is very easy. It can do with in a few minutes. Our system has three type of accessing modes, student, teacher and administrator. Student Information System is managed by an administrator. It is the job of the administrator to insert update and monitor the whole process. When a user logs in to the system. He/she would only view details of the student. He/she can't perform any changes. Our system has seven modules, they are administrator, student, course, department, exam, attendance, and section. These modules and its attributes with entity relationship module presented in the ER diagram section.

1.2 Project Top View

Project Id	SUSIS-01
Project Name	SU Student Information System
Project Manager	Sadia Tasnim Barsha
Project Quality Facilitator	Md Uzzal Hossain
Client Name	SU Student Information System
Project Category	Medium
Platform/ Technology Description (Operating system, database, language, front-end, etc.)	Platform-windows Technology-ASP .NET Database-SQL Server, Language – C#
Project Start Date	01/01/2019
Estimated Project End Date	16/05/2019
Total Estimated Calendar Days	82days
Total Estimated Size (say, QSU)	
Total Estimated Effort	

1.3 Project Scope, Objectives

- Student Information system is a management information system for education establishments to manage student data.
- Provide capabilities for registering students in courses, admitting students, tracking student attendance and submitting students' documents and Result.
- Ensure data integrity, privacy, and security in an open-access environment.
- To collect easily session wise and course wise lecture any time from online.

1.4 Motivation

- Student Information System is an online base software. So that, student collect their every course lecture and important document anytime from anywhere.
- Sometimes student collect their important file or document by pen drive. It is long time process and

file would be missing. The system is very secure. User can easily access and collect.

- Manage session and course wise document.
- Get student and teacher details.
- Manage and view feedback and notice.
- Show result, notice and another important document.

Chapter 2

Project Review

2.1 Project management

Project management skills are put to good use for this project. Having gone through project management modules in Time Series Analysis, Optimization and with two interns Project Management for Business and IT respectively, they enhanced my knowledge on managing a project. Project management focuses on achieving the objectives by applying five processes presented in Figure below.

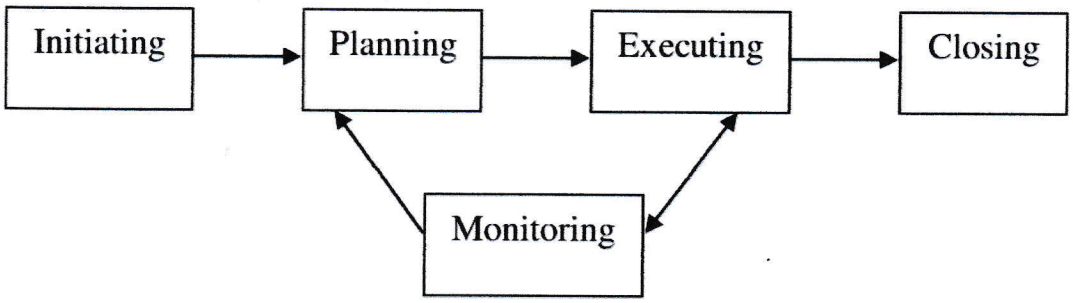


Figure 2.1: Project Development Phases

2.2 Student Information system

There are many software development companies that offer student management system for schools in the market. There are records on the past years projects on student management system is done by students. Through the researches, it is observed that there are features where this project can adopt and implement. One of it will be with addition of new course or class, or even upgrading of students to the next level, the school administrator can easily register all of them within a particular class into the new one using just a page and not having to register one at a time. With this feature, it helps administrator to save time as well as increase their efficiency.

Components

- Student features.
- Teacher features.
- Administrator features.

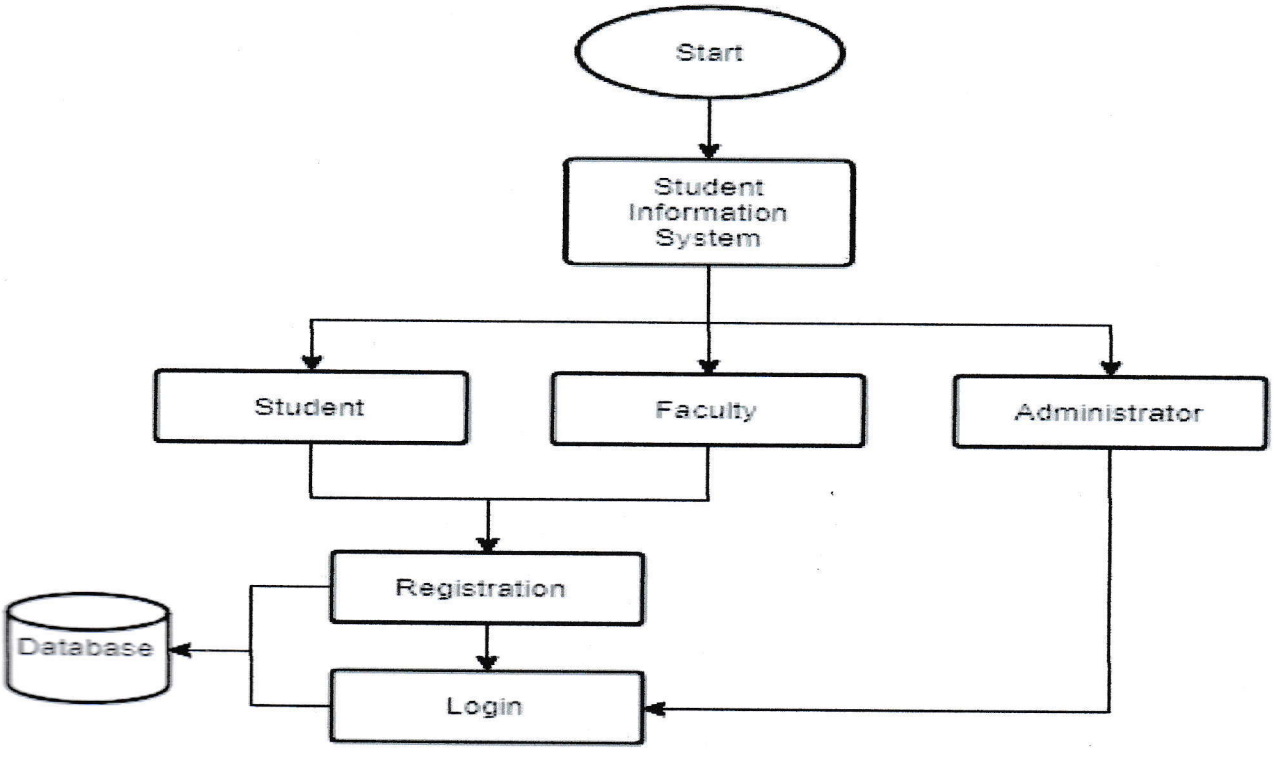


Figure 2.2.1: Data Flow Diagram for SIS Components

Roles or Features

- Document Management.
- Upload single file.
- Upload multiple file.
- Session or Cabinet.
- Course or Drawer.
- Student Teacher details.
- Result.
- Feedback.
- Attendance.
- Notice.
- Student Management.
- Teacher Management.
- Batch.

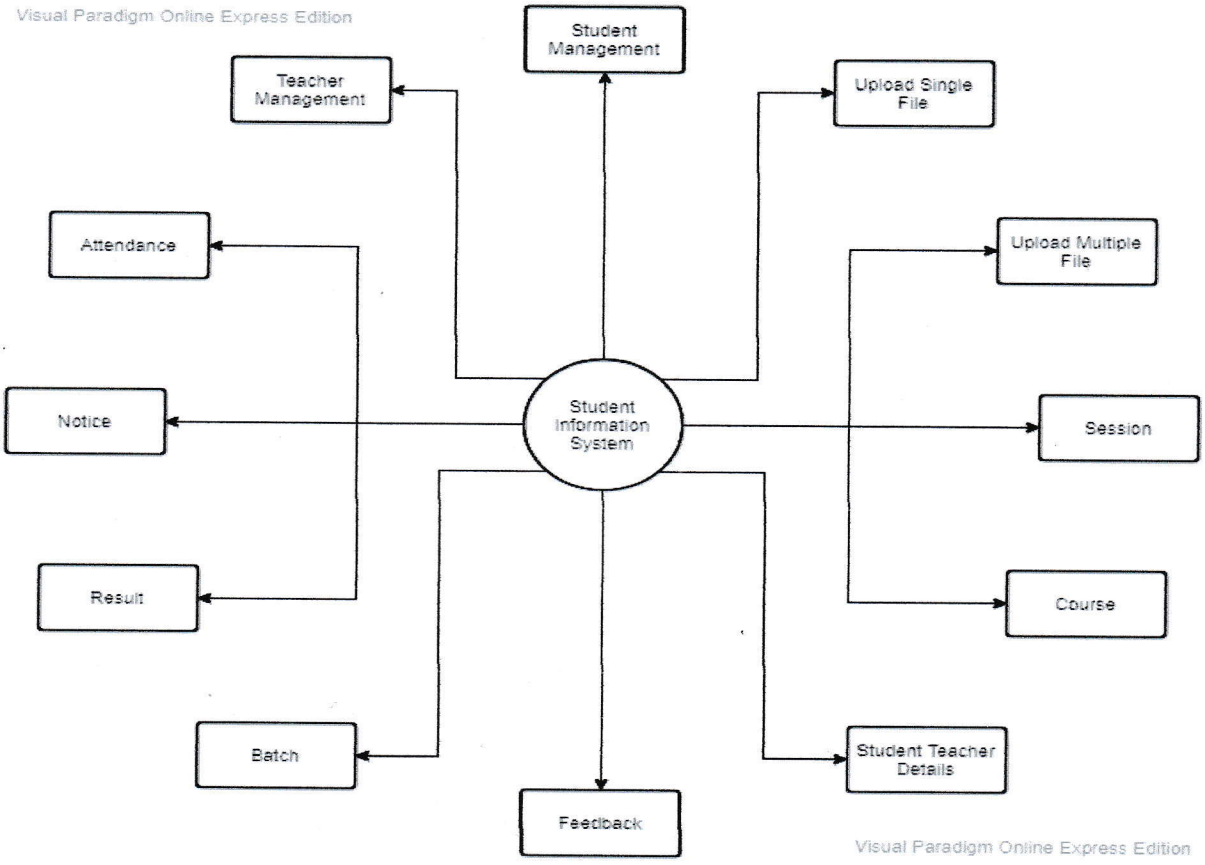


Figure 2.2.2: Data Flow Diagram for SIS Roles or Features

Student Features

- Document Management.
- Result.
- Feedback.
- Attendance.
- Notice.
- Student Teacher details.

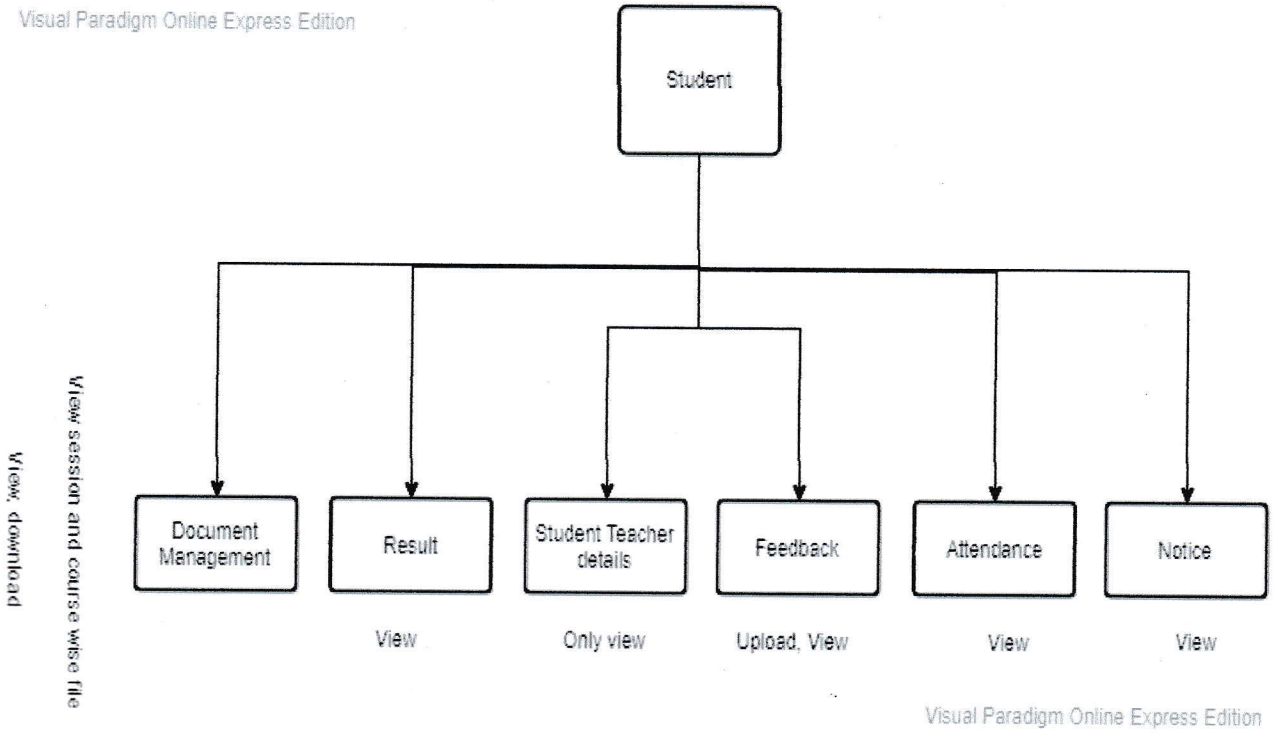


Figure 2.2.3: Data Flow Diagram for Student Features

Work Process:

- Firstly, Student registered SU Student Information System with their required all information.
- After registered SUSIS System then Login this system and get Home page (Document management).
- Document management module arrange session and course wise file. Suppose, Session-CSE0318-Neon(6B),
- Course- Software Engineering.

- Student can download and view their session and course wise file.
- They search easily with all narrations. Suppose Section name, File Name, Subject Name etc.
- They send and receive feedback.
- Get semester result with details and generate mark sheet.
- Show notice.
- Get Student Teacher details information.

Teacher Features

- Document Management.
- Upload single file.
- Upload multiple file.
- Course or Drawer.
- Student Teacher details.
- Result.
- Feedback.
- Attendance.
- Notice.

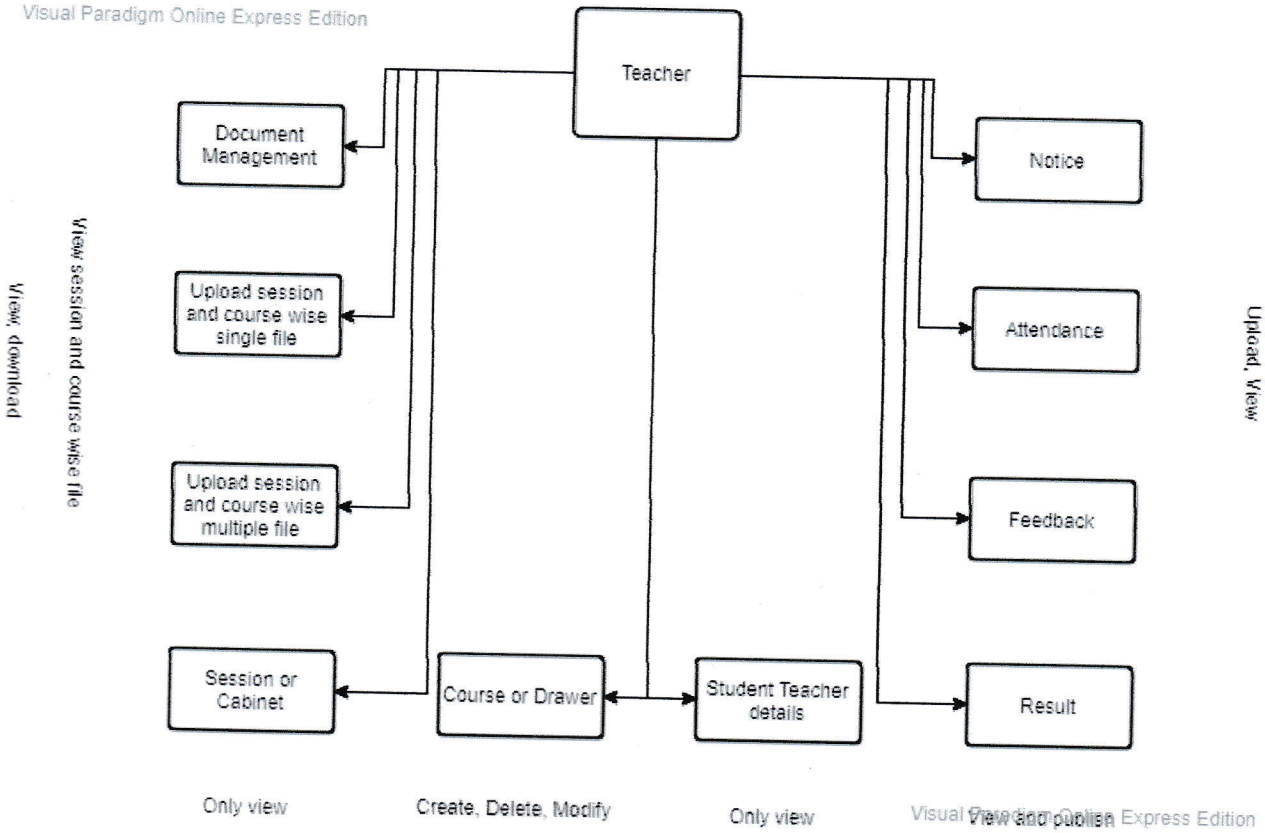


Figure 2.2.4: DFD for teacher

Work Process:

- Firstly, registered SU Student Information System with their required all information.
- After registered SUSIS System then Login this system and get Home page (Document management).
- Document management module arrange session and course wise file. Suppose, Session-CSE0318-Neon(6B),
- Course- Software Engineering.
- Teacher can upload their session and course wise file.
- They search easily with all narrations. Suppose Section name, File Name, Subject Name etc.
- They send and receive feedback.
- Published semester result session wise.
- Show notice.
- Get Student Teacher details information.

Administrator Features

- Administrator Features
- Document Management.
- Upload single file.
- Upload multiple file.
- Session or Cabinet.
- Course or Drawer.
- Student Teacher details.
- Result.
- Feedback.
- Attendance.
- Notice.
- Student Management.
- Teacher Management.
- Batch.

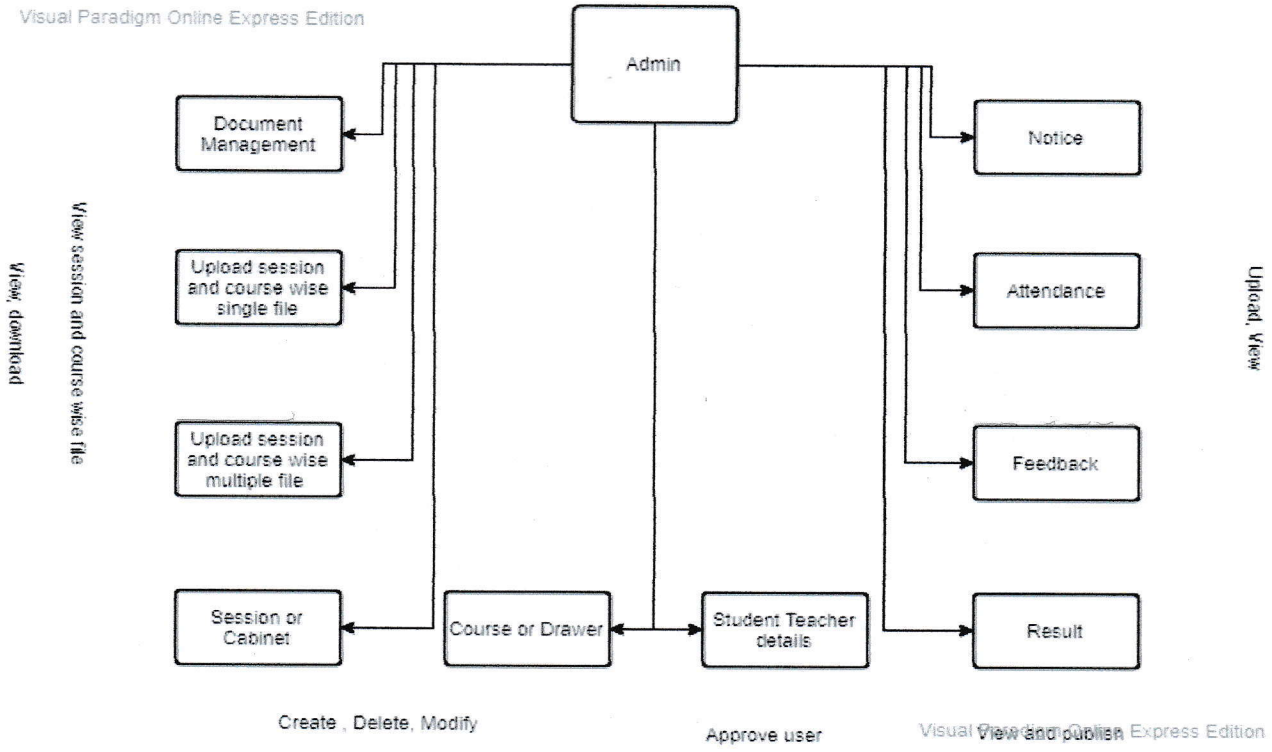


Figure 2.2.5: DFD for administrator

Work Process:

- Firstly, registered SU Student Information System with their required all information.
- After registered SUSIS System then Login this system and get Home page (Document management).
- Document management module arrange session and course wise file. Suppose, Session-CSE0318-Neon(6B),
- Course- Software Engineering.
- Administrator get upload file system.
- Create session and course for upload file.
- Approved or denied user for login this system.
- Create batch.
- They search easily with all narrations. Suppose Section name, File Name, Subject Name etc.
- They send and receive feedback.
- Published semester result session wise.
- Show notice and Student Teacher details information.

Chapter 3

Resources

3.1 System Development life cycle

Systems Development Life Cycle (SDLC) is the most common process adopted to develop a project and not surprisingly, this project is following this model too. To be precise, waterfall model is being applied. Waterfall model is a sequential model process where the input of a phase actually results from the previous phase.

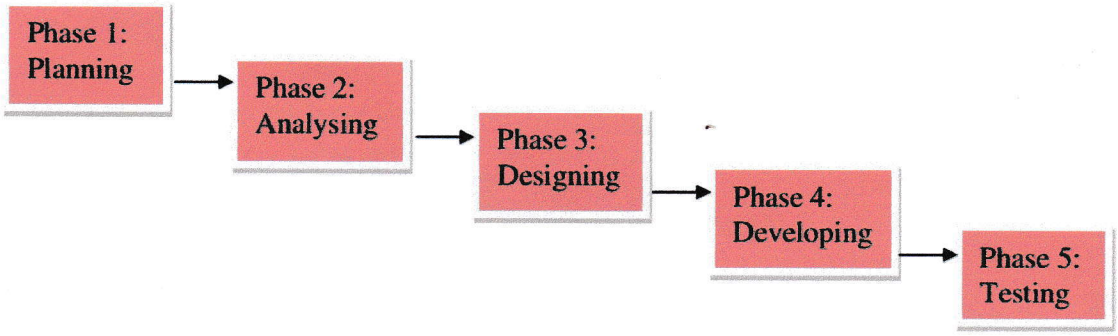


Figure 3.1: SDLC Phases

There are five phases in this model and the first phase is the planning stage. The planning stage determines the objectives of the project and whether the project should be given the green light to proceed. This is where the proposal submission comes into picture. After obtaining the approval, the next phase is analysis. Gathering and analyzing the system and user requirements is essential for entry to the design step.

With the user requirements gathering completed, there is a need to prepare the resources for the project. Be it software or hardware components, careful consideration and selection is to be taken care at this stage. The decision on the appropriate resources to be used is further elaborated under the subsections below. The next step is to design the system and database structure.

Results from the analysis and preparation that were concluded from the previous stage are put into action. With the user requirements in mind, the flow of the system is planned and

the user interface is designed to suit their easy navigation needs. In addition, the number of tables, attributes, primary and unique keys of the database is listed.

After completing the design, actual coding begins. Database is created and codes are written. Some of the codes required amendments and improvement to it so these are being developed at this fourth stage of the waterfall model. With the development completed, testing will begin. The codes and database are tested to ensure the results obtained are as intended. More time is spent on both development and testing stages because it is inevitable to have errors and issues and buffer time is allocated for troubleshooting.

Chapter 4

System Analysis and Design

4.1 Use case analysis

In order to provide a clearer picture of the functionality provided by the student management module, we have done a use case analysis. Figure below is a use case diagram to present the functionality in the student management module.

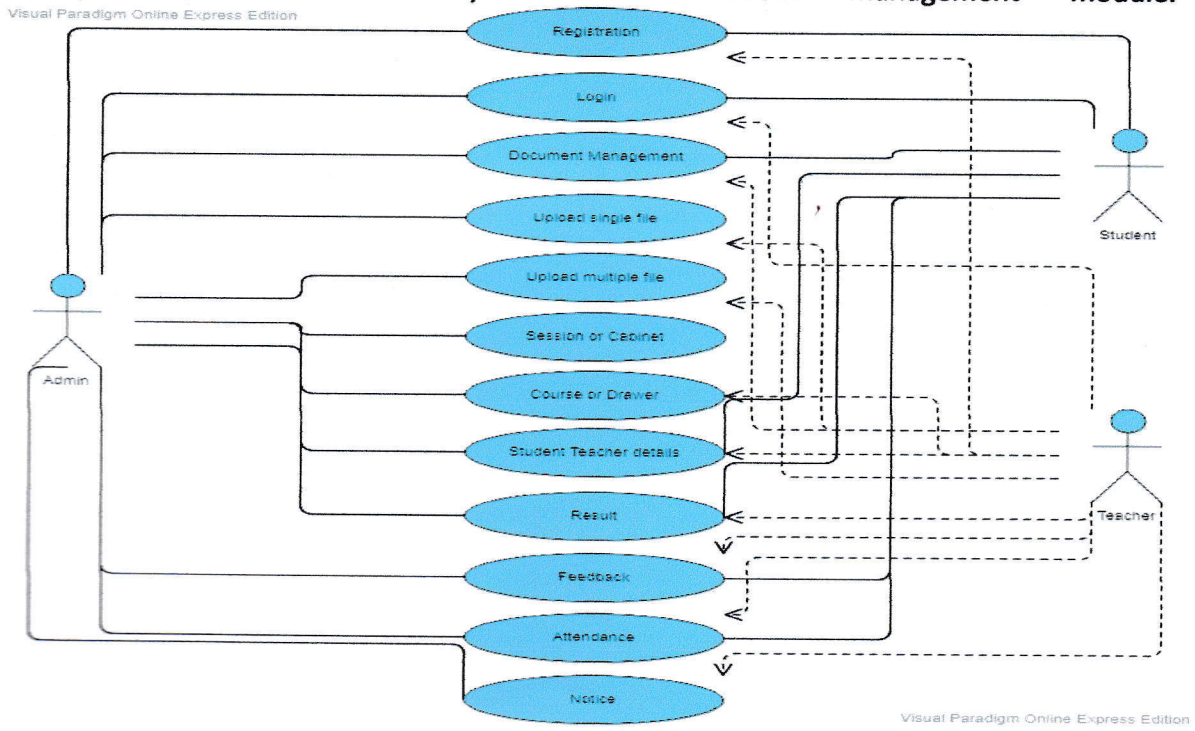
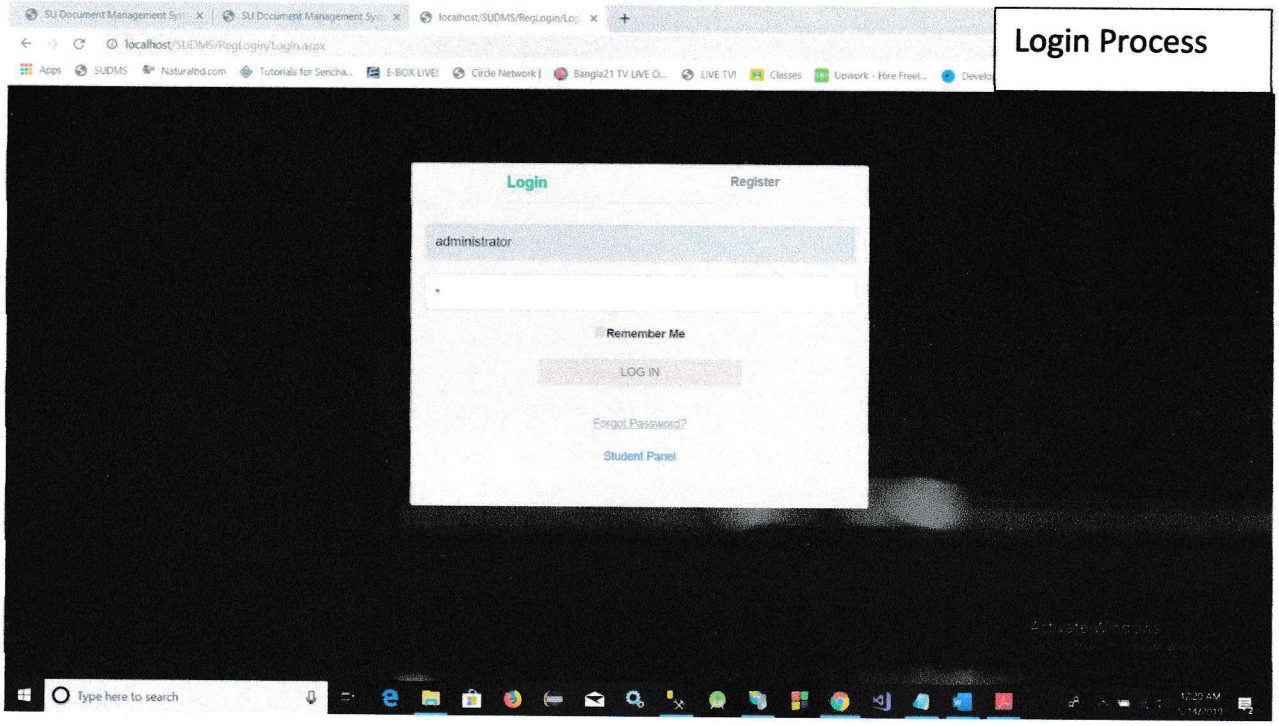
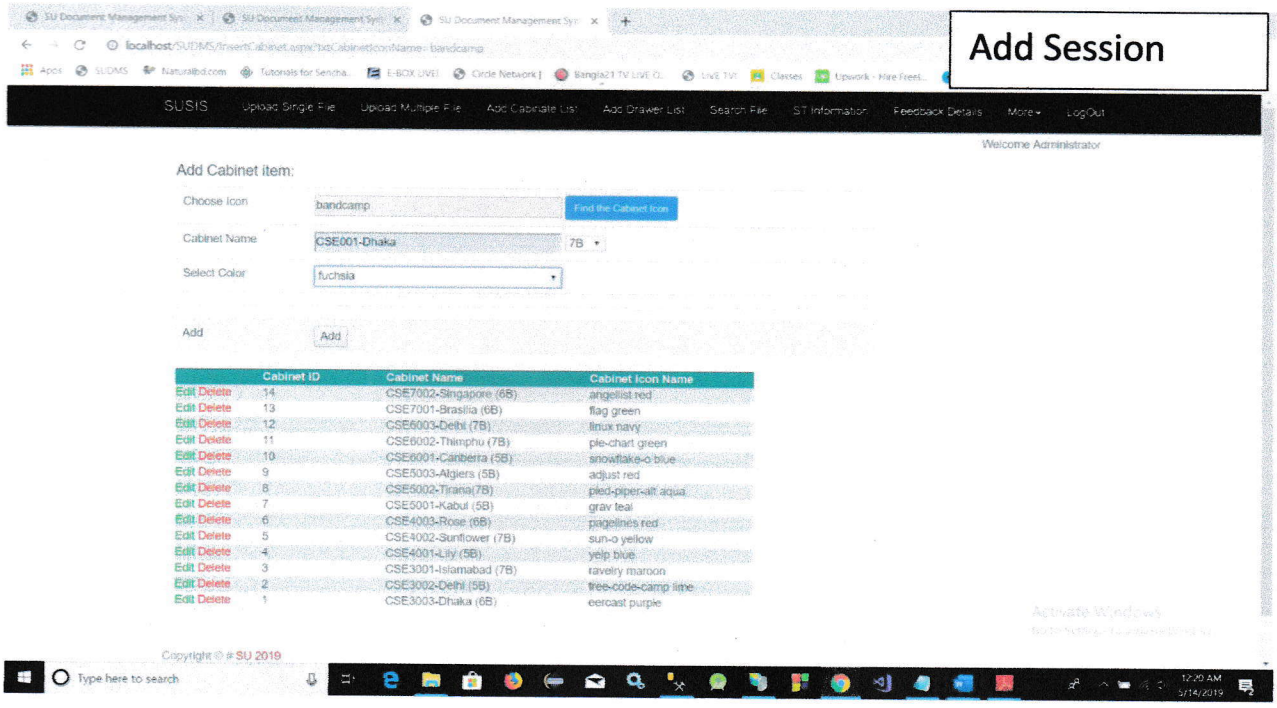


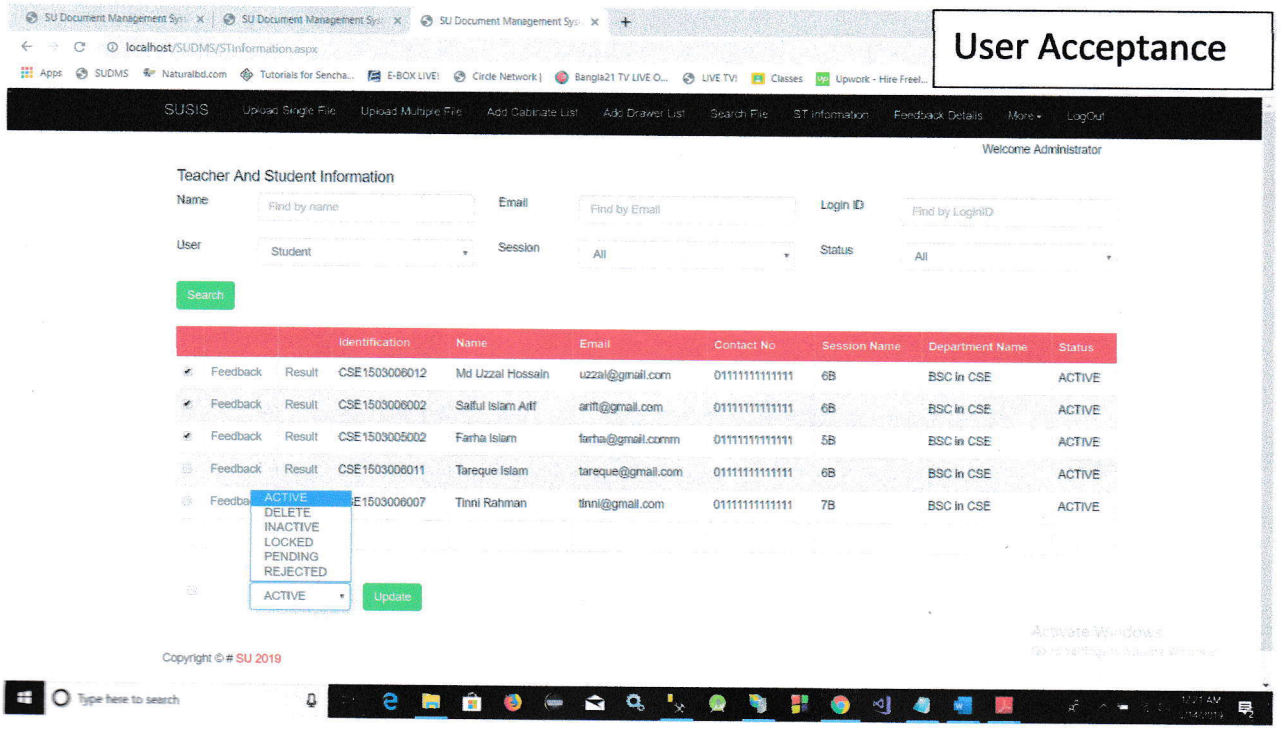
Figure 4.1: use case analysis

4.2 System Design

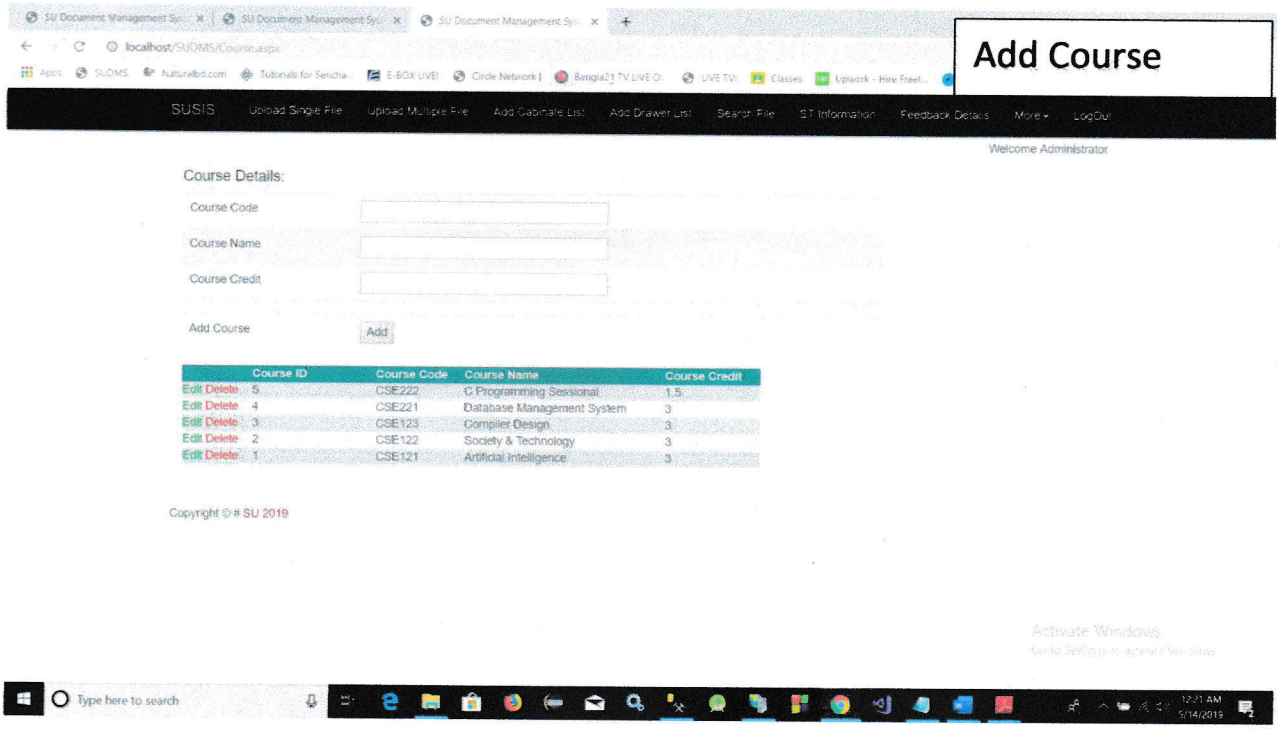
Administrator feature



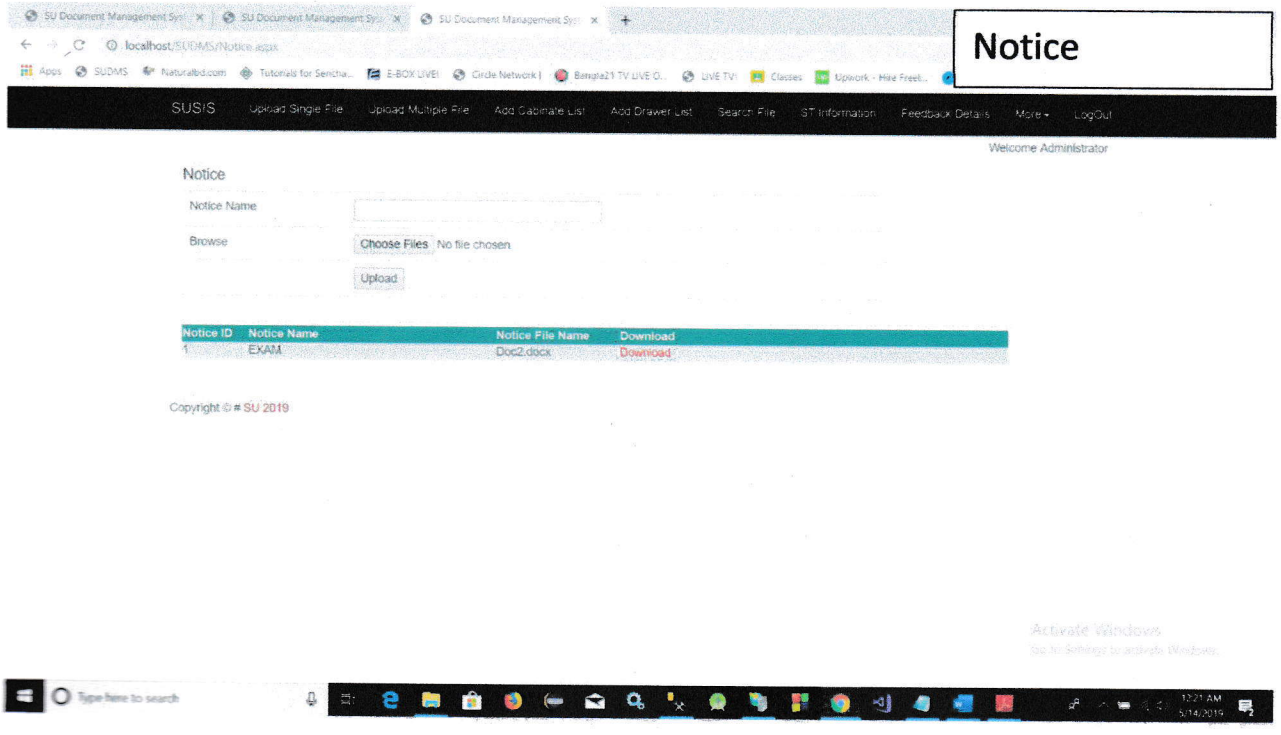

Add Session
SUSIS
Upload Single File
Upload Multiple File
Add Cabinate List
Add Drawer List
Search File
ST Information
Feedback Details
More
LogOut
Welcome Administrator
Add Cabinet item:
Choose Icon
bandcamp
Find the Cabinet Icon
Cabinet Name
CSE001-Dhaka
7B
Select Color
fuchsia
Add
Add
Cabinet ID
Cabinet Name
Cabinet Icon Name
Edit Delete 14 CSE7002-Singapore (6B) angellist red
Edit Delete 13 CSE7001-Brasilia (6B) flag green
Edit Delete 12 CSE6003-Delhi (7B) linux navy
Edit Delete 11 CSE6002-Thimphu (7B) pie-chart green
Edit Delete 10 CSE6001-Canberra (5B) snowflake-o blue
Edit Delete 9 CSE5003-Algiers (5B) adjust red
Edit Delete 8 CSE5002-Tirana(7B) pied-piper-alt aqua
Edit Delete 7 CSE5001-Kabul (5B) grav teal
Edit Delete 6 CSE4003-Rose (6B) pagelines red
Edit Delete 5 CSE4002-Sunflower (7B) sun-o yellow
Edit Delete 4 CSE4001-Lily (5B) yelp blue
Edit Delete 3 CSE3001-Islamabad (7B) ravelry maroon
Edit Delete 2 CSE3002-Delhi (5B) free-code-camp lime
Edit Delete 1 CSE3003-Dhaka (6B) eercast purple
Copyright © # SU 2019
Type here to search


User Acceptance
localhost/SUDMS/STInformation.aspx
SUSIS
Upload Single File
Upload Multiple File
Add Cabinate List
Add Drawer List
Search File
ST Information
Feedback Details
More
LogOut
Welcome Administrator
Teacher And Student Information
Name
Find by name
Email
Find by Email
Login ID
Find by LoginID
User
Student
Session
All
Status
All
Search
Identification
Name
Email
Contact No
Session Name
Department Name
Status
Feedback Result CSE1503006012 Md Uzzal Hossain uzzal@gmail.com 01111111111111 6B BSC in CSE ACTIVE
Feedback Result CSE1503006002 Saiful Islam Arif arift@gmail.com 01111111111111 6B BSC in CSE ACTIVE
Feedback Result CSE1503005002 Farha Islam farha@gmail.comm 01111111111111 5B BSC in CSE ACTIVE
Feedback Result CSE1503006011 Tareque Islam tareque@gmail.com 01111111111111 6B BSC in CSE ACTIVE
Feedba E1503006007 Tinni Rahman tinni@gmail.com 01111111111111 7B BSC in CSE ACTIVE
ACTIVE
DELETE
INACTIVE
LOCKED
PENDING
REJECTED
ACTIVE
Update
Copyright © # SU 2019
Type here to search

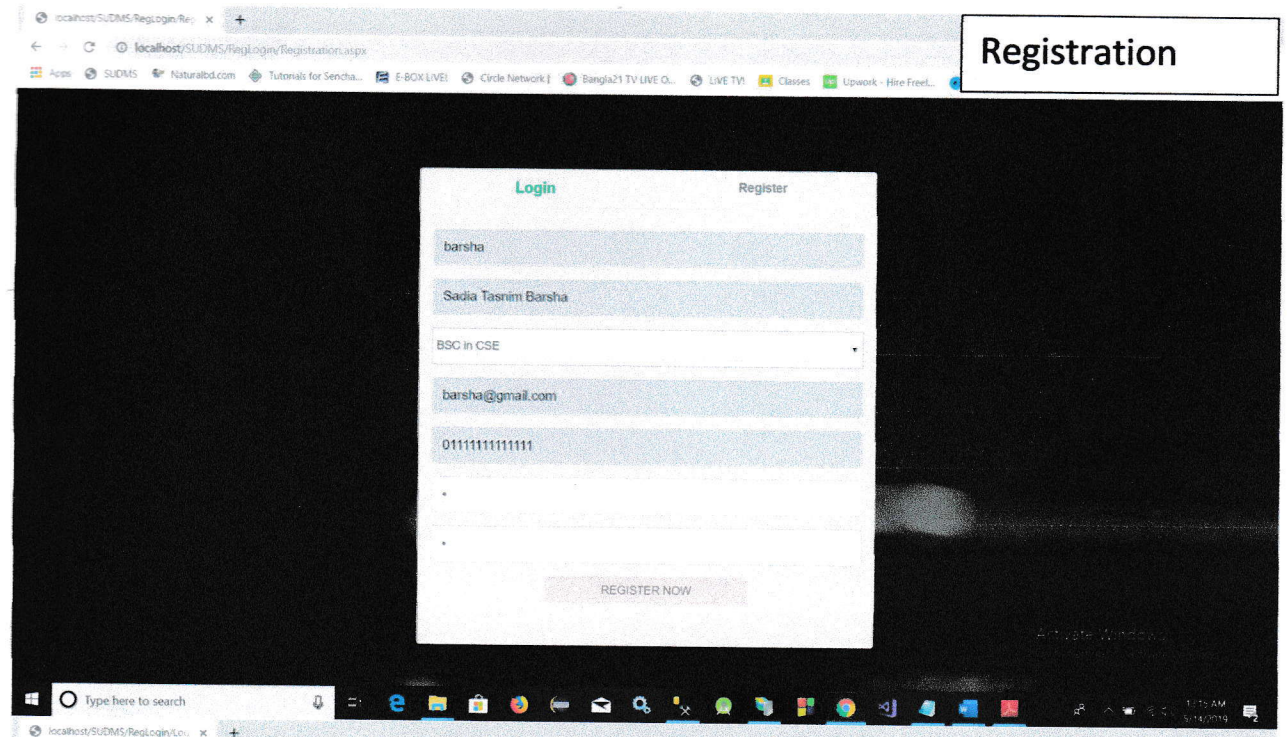
Add Course

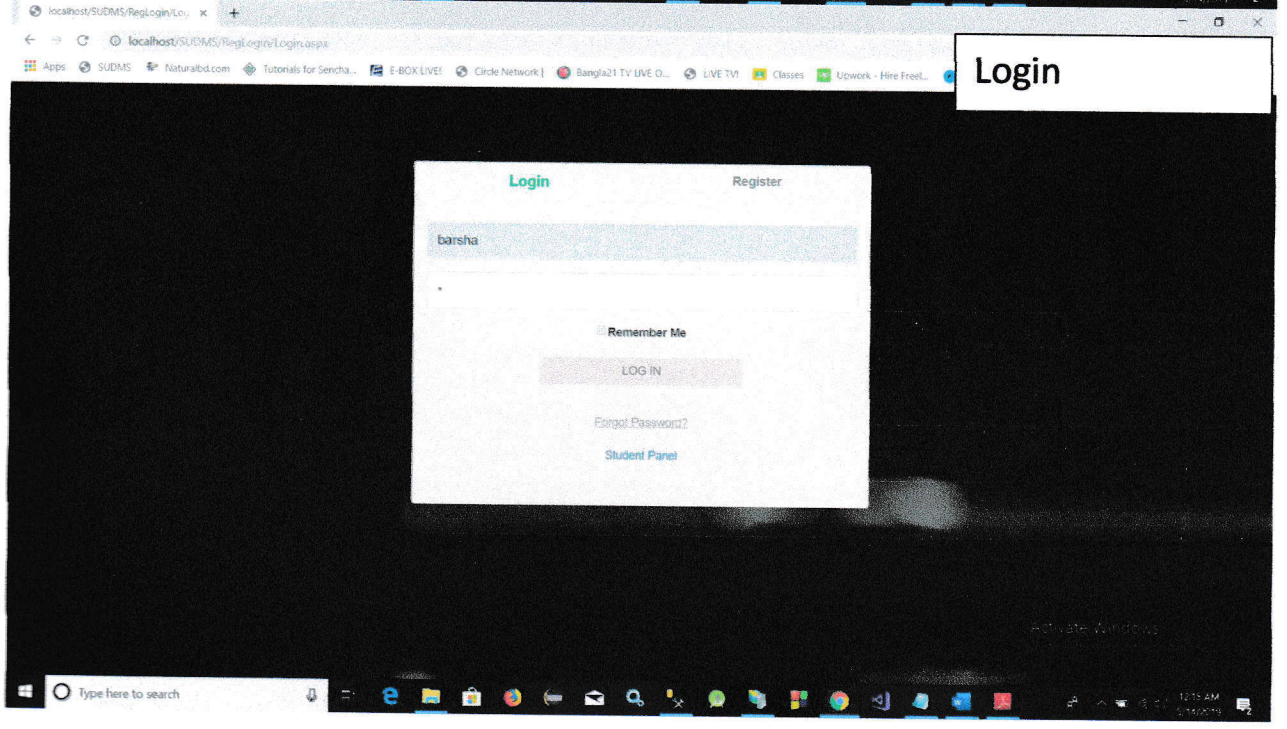


Notice

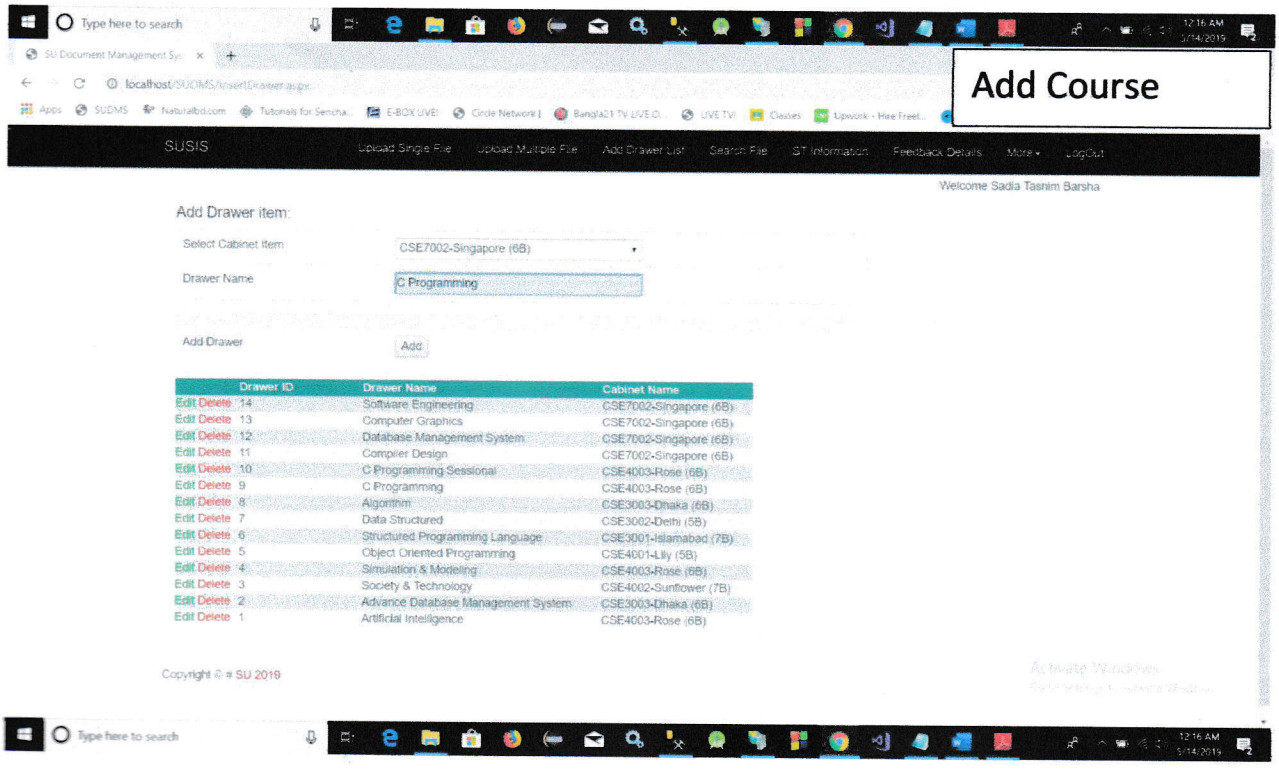


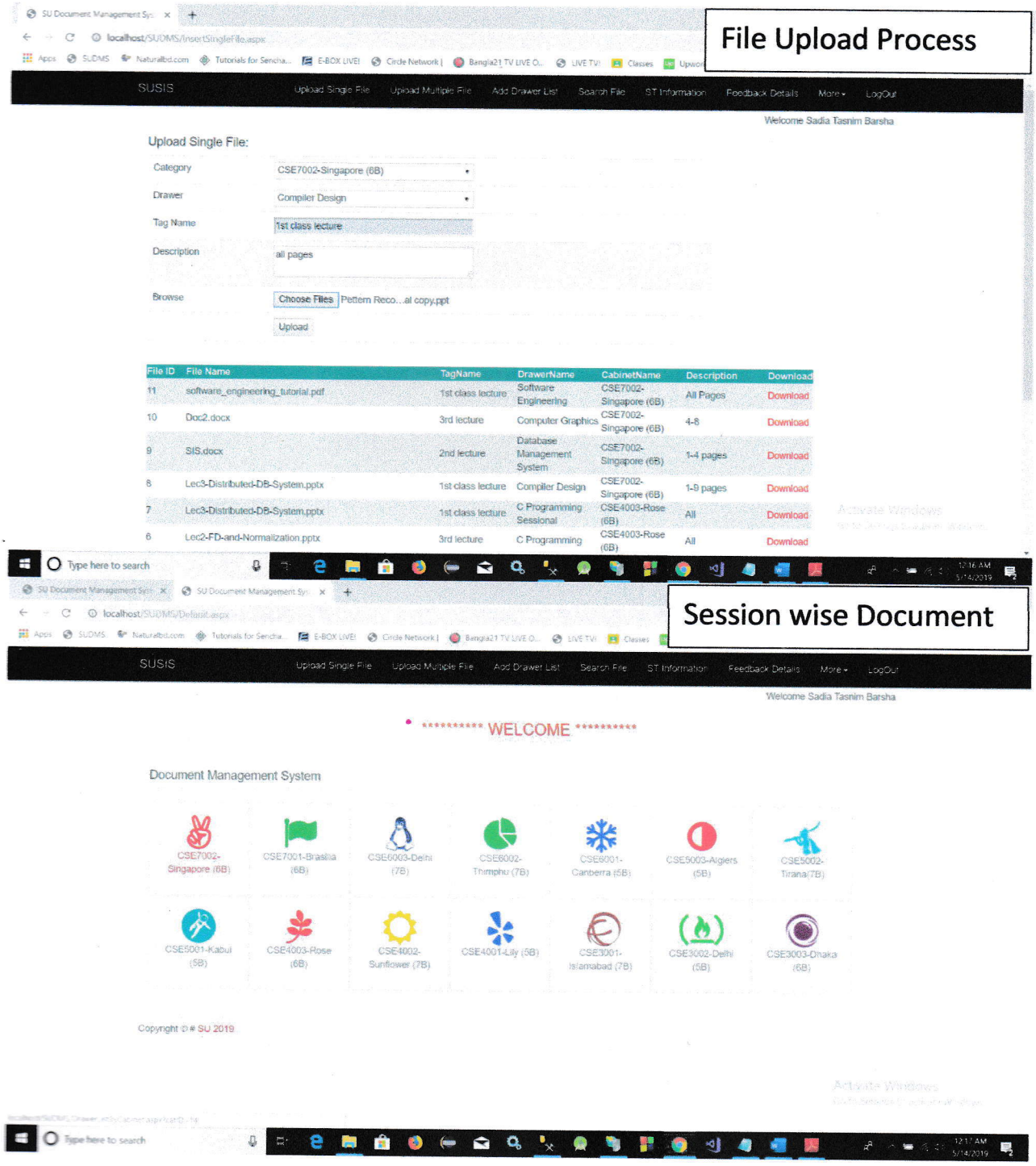
Teacher features



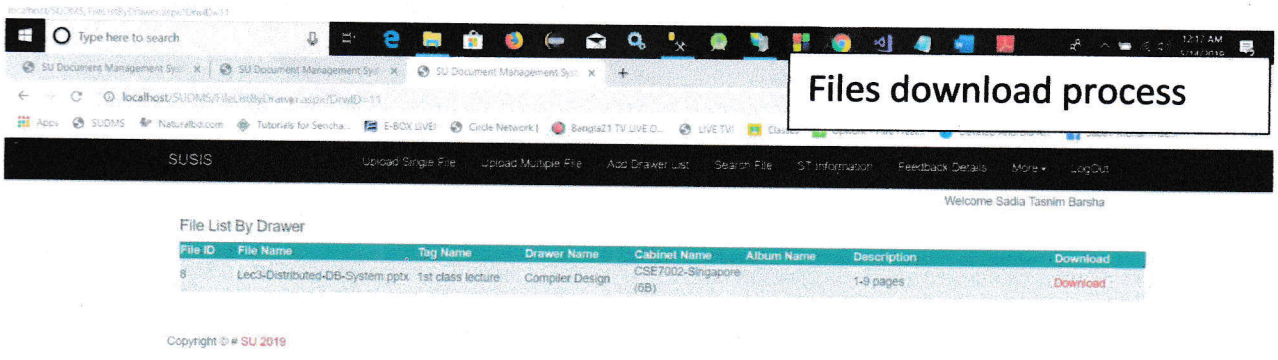



Document management
SUSIS
Upload Single File
Upload Multiple File
Add Drawer List
Search File
ST Information
Feedback Details
More
LogOut
Welcome Sadia Tasnim Barsha
********** WELCOME **********
Document Management System
CSE7002-Singapore (6B)
CSE7001-Brasilia (6B)
CSE6003-Delhi (7B)
CSE6002-Thimphu (7B)
CSE6001-Canberra (5B)
CSE5003-Algiers (5B)
CSE5002-Tirana(7B)
CSE5001-Kabul (5B)
CSE4003-Rose (6B)
CSE4002-Sunflower (7B)
CSE4001-Lily (5B)
CSE3001-Islamabad (7B)
CSE3002-Delhi (5B)
CSE3003-Dhaka (6B)
Copyright © SU 2019

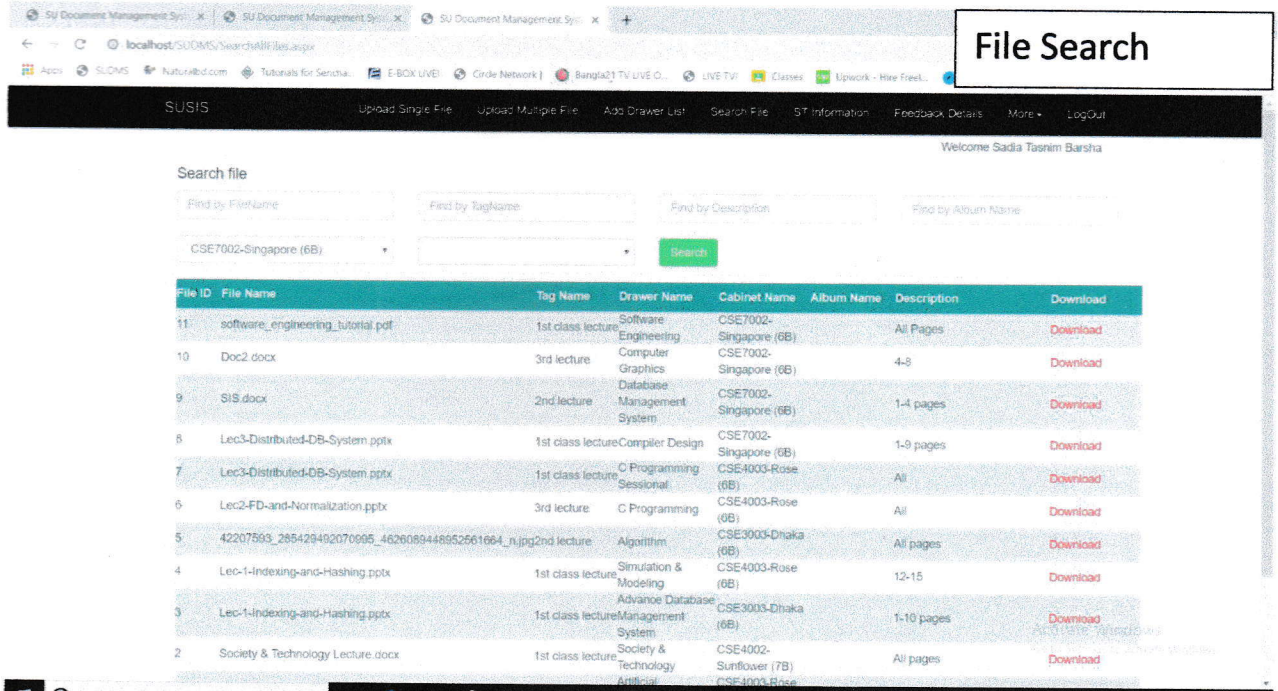

Add Course
SUSIS
Upload Single File
Upload Multiple File
Add Drawer List
Search File
ST Information
Feedback Details
More
LogOut
Welcome Sadia Tasnim Barsha
Add Drawer item:
Select Cabinet Item
CSE7002-Singapore (6B)
Drawer Name
C Programming
Add Drawer
Add
Drawer ID | Drawer Name | Cabinet Name
Edit Delete 14 | Software Engineering | CSE7002-Singapore (6B)
Edit Delete 13 | Computer Graphics | CSE7002-Singapore (6B)
Edit Delete 12 | Database Management System | CSE7002-Singapore (6B)
Edit Delete 11 | Compiler Design | CSE7002-Singapore (6B)
Edit Delete 10 | C Programming Sessional | CSE4003-Rose (6B)
Edit Delete 9 | C Programming | CSE4003-Rose (6B)
Edit Delete 8 | Algorithm | CSE3003-Dhaka (6B)
Edit Delete 7 | Data Structured | CSE3002-Delhi (5B)
Edit Delete 6 | Structured Programming Language | CSE3001-Islamabad (7B)
Edit Delete 5 | Object Oriented Programming | CSE4001-Lily (5B)
Edit Delete 4 | Simulation & Modeling | CSE4003-Rose (6B)
Edit Delete 3 | Society & Technology | CSE4002-Sunflower (7B)
Edit Delete 2 | Advance Database Management System | CSE3003-Dhaka (6B)
Edit Delete 1 | Artificial Intelligence | CSE4003-Rose (6B)
Copyright © SU 2019


File Upload Process
SUSIS
Upload Single File
Upload Multiple File
Add Drawer List
Search File
ST Information
Feedback Details
More
LogOut
Welcome Sadia Tasnim Barsha
Upload Single File:
Category
CSE7002-Singapore (6B)
Drawer
Compiler Design
Tag Name
1st class lecture
Description
all pages
Browse
Choose Files
Pettern Reco...al copy.ppt
Upload
File ID
File Name
TagName
DrawerName
CabinetName
Description
Download
11
software_engineering_tutorial.pdf
1st class lecture
Software Engineering
CSE7002-Singapore (6B)
All Pages
Download
10
Doc2.docx
3rd lecture
Computer Graphics
CSE7002-Singapore (6B)
4-8
Download
9
SIS.docx
2nd lecture
Database Management System
CSE7002-Singapore (6B)
1-4 pages
Download
8
Lec3-Distributed-DB-System.pptx
1st class lecture
Compiler Design
CSE7002-Singapore (6B)
1-9 pages
Download
7
Lec3-Distributed-DB-System.pptx
1st class lecture
C Programming Sessional
CSE4003-Rose (6B)
All
Download
6
Lec2-FD-and-Normalization.pptx
3rd lecture
C Programming
CSE4003-Rose (6B)
All
Download
Session wise Document
SUSIS
Upload Single File
Upload Multiple File
Add Drawer List
Search File
ST Information
Feedback Details
More
LogOut
Welcome Sadia Tasnim Barsha
********** WELCOME **********
Document Management System
CSE7002-Singapore (6B)
CSE7001-Brasilia (6B)
CSE6003-Delhi (7B)
CSE6002-Thimphu (7B)
CSE6001-Canberra (5B)
CSE5003-Algiers (5B)
CSE5002-Tirana(7B)
CSE5001-Kabul (5B)
CSE4003-Rose (6B)
CSE4002-Sunflower (7B)
CSE4001-Lily (5B)
CSE3001-Islamabad (7B)
CSE3002-Delhi (5B)
CSE3003-Dhaka (6B)
Copyright © # SU 2019

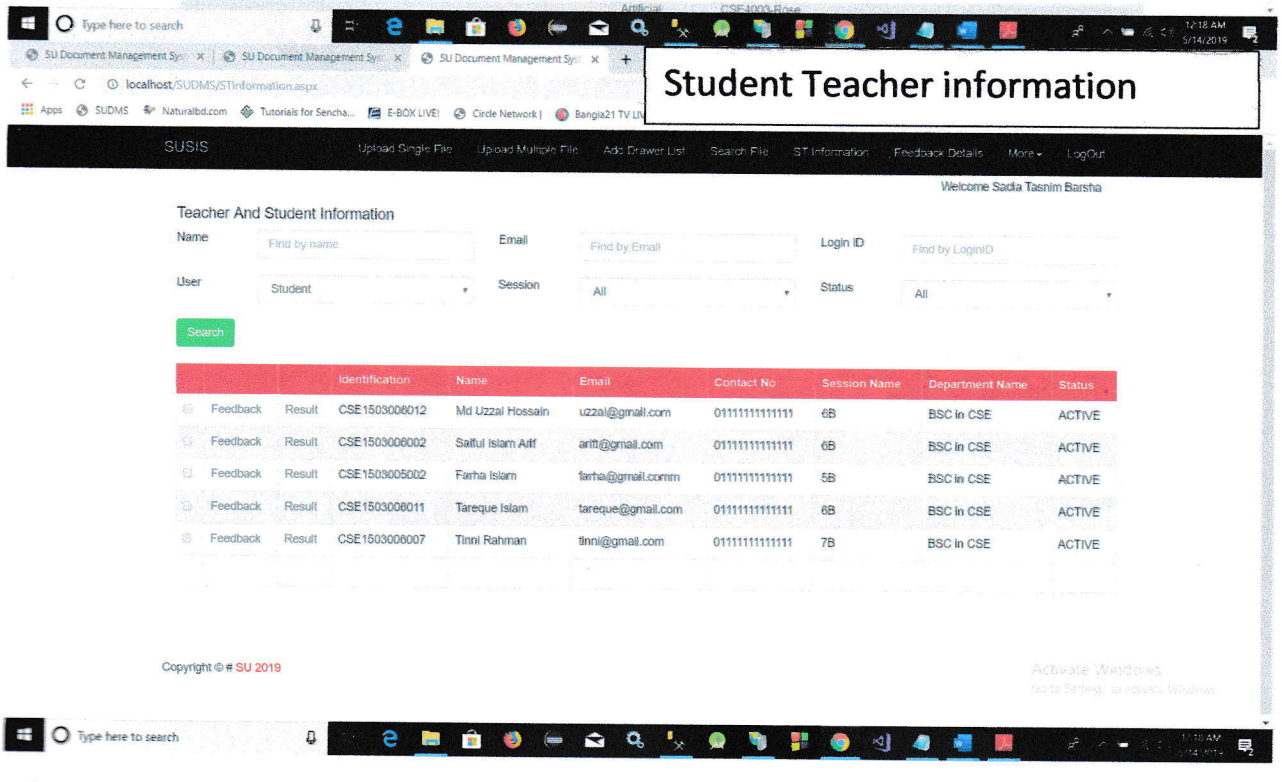

Course wise document
SUSIS
Upload Single File
Upload Multiple File
Add Drawer List
Search File
ST Information
Feedback Details
More
LogOut
Welcome Sadia Tasnim Barsha
Drawer List By Cabinet
Compiler Design
Copyright © # SU 2019


Files download process
SUSIS
Upload Single File
Upload Multiple File
Add Drawer List
Search File
ST Information
Feedback Details
More
LogOut
Welcome Sadia Tasnim Barsha
File List By Drawer
File ID
File Name
Tag Name
Drawer Name
Cabinet Name
Album Name
Description
Download
6
Lec3-Distributed-DB-System.pptx
1st class lecture
Compiler Design
CSE7002-Singapore (6B)
1-9 pages
Download
Copyright © # SU 2019

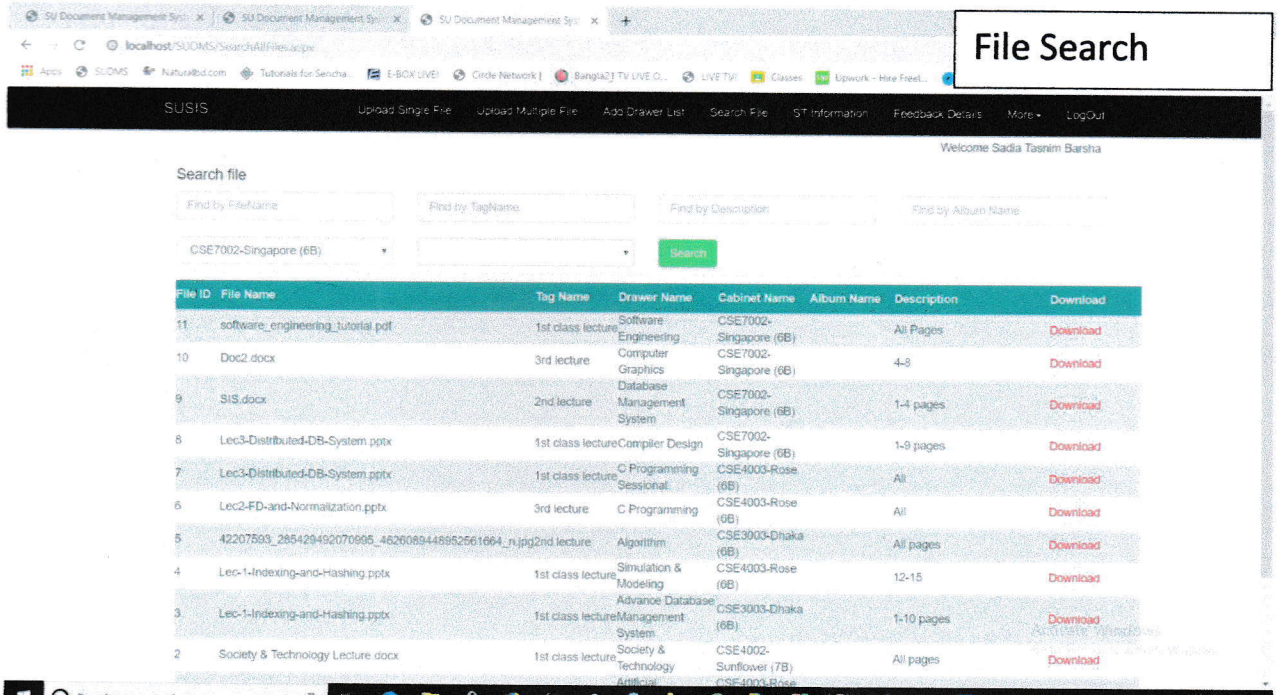
File Search



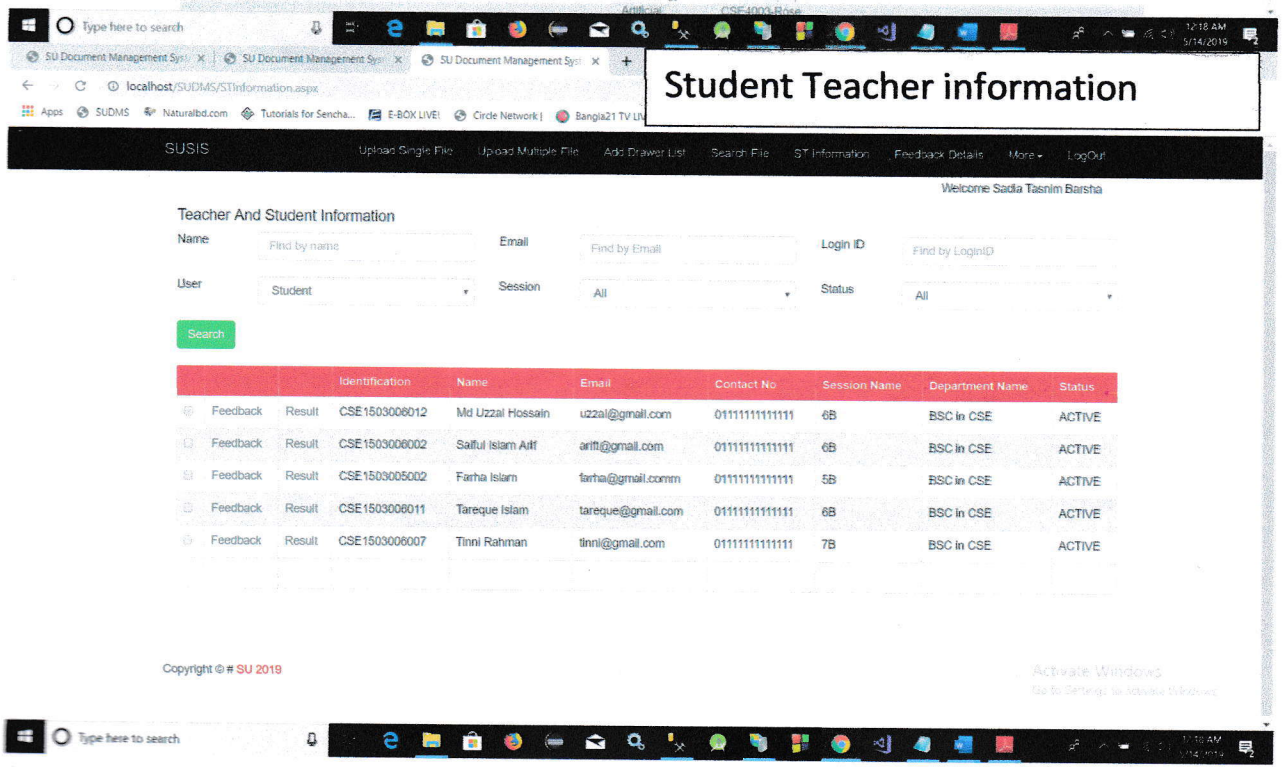
Student Teacher information



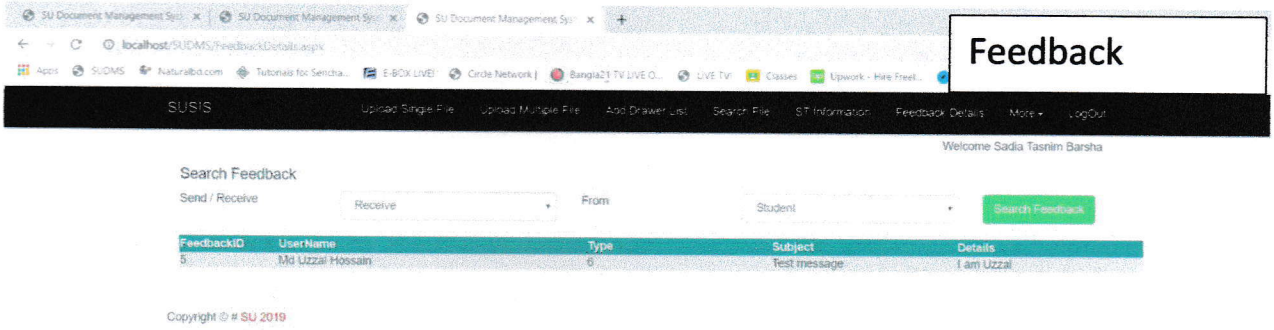
File Search



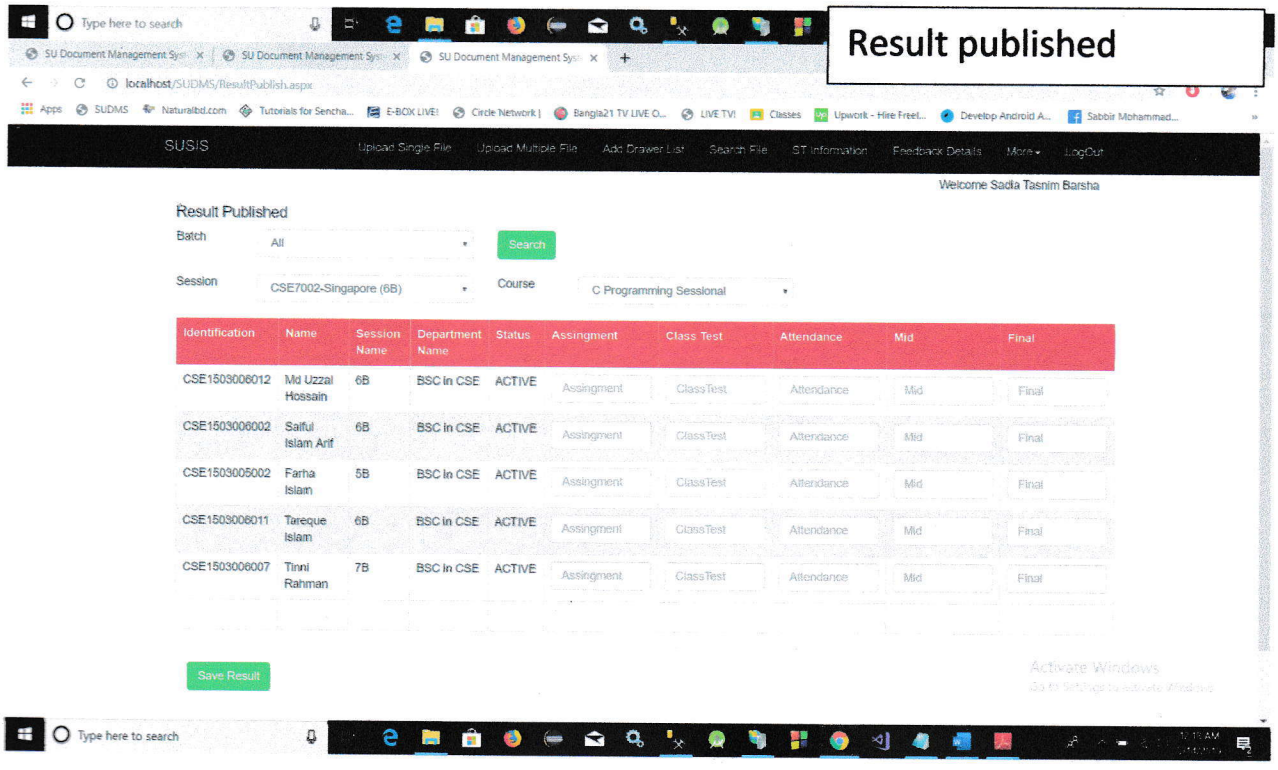
Student Teacher information



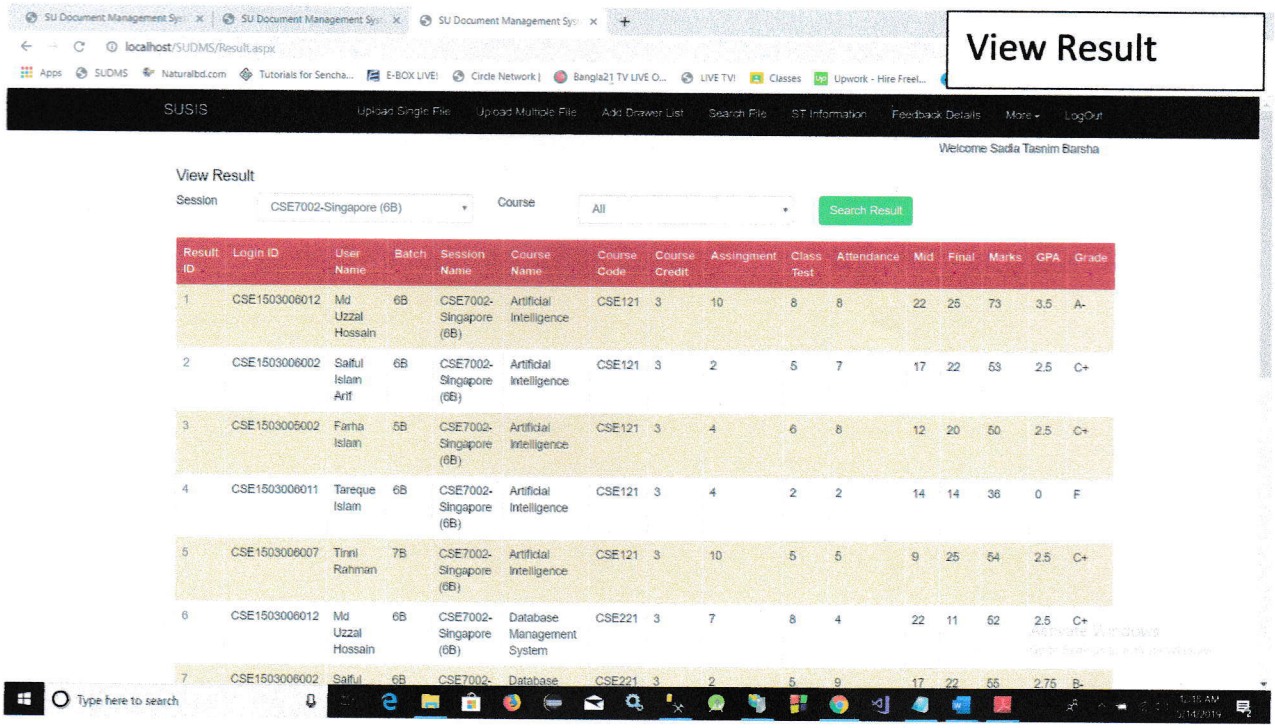
Feedback



Result published



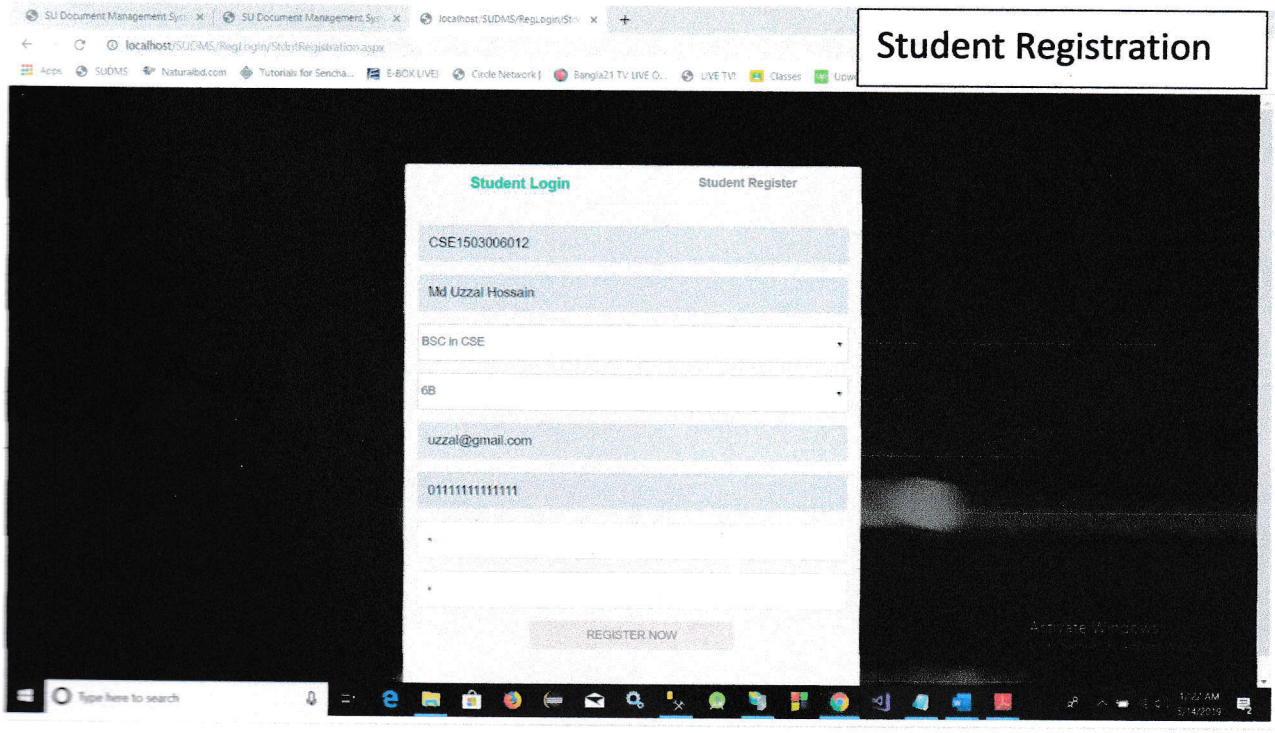
View Result



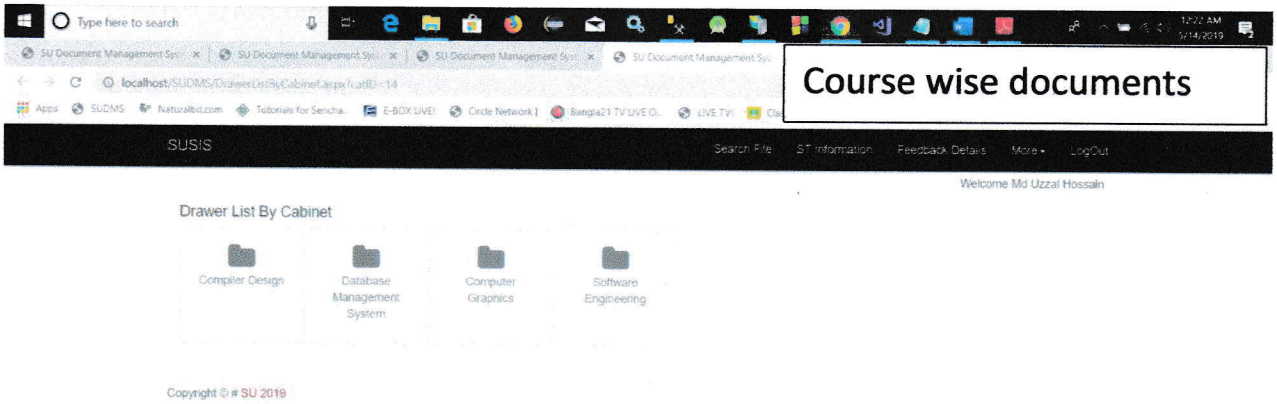
Result ID	Login ID	User Name	Batch	Session Name	Course Name	Course Code	Course Credit	Assingment	Class Test	Attendance	Mid	Final	Marks	GPA	Grade
1	CSE1503006012	Md Uzzal Hossain	6B	CSE7002-Singapore (6B)	Artificial Intelligence	CSE121	3	10	8	8	22	25	73	3.5	A-
2	CSE1503006002	Saiful Islam Arif	6B	CSE7002-Singapore (6B)	Artificial Intelligence	CSE121	3	2	5	7	17	22	53	2.5	C+
3	CSE1503005002	Farha Islam	5B	CSE7002-Singapore (6B)	Artificial Intelligence	CSE121	3	4	6	8	12	20	50	2.5	C+
4	CSE1503006011	Tareque Islam	6B	CSE7002-Singapore (6B)	Artificial Intelligence	CSE121	3	4	2	2	14	14	36	0	F
5	CSE1503006007	Tinni Rahman	7B	CSE7002-Singapore (6B)	Artificial Intelligence	CSE121	3	10	5	5	9	25	54	2.5	C+
6	CSE1503006012	Md Uzzal Hossain	6B	CSE7002-Singapore (6B)	Database Management System	CSE221	3	7	8	4	22	11	52	2.5	C+
7	CSE1503006002	Saiful	6B	CSE7002-	Database	CSE221	3	2	5	9	17	22	55	2.75	B-

Student features

Student Registration

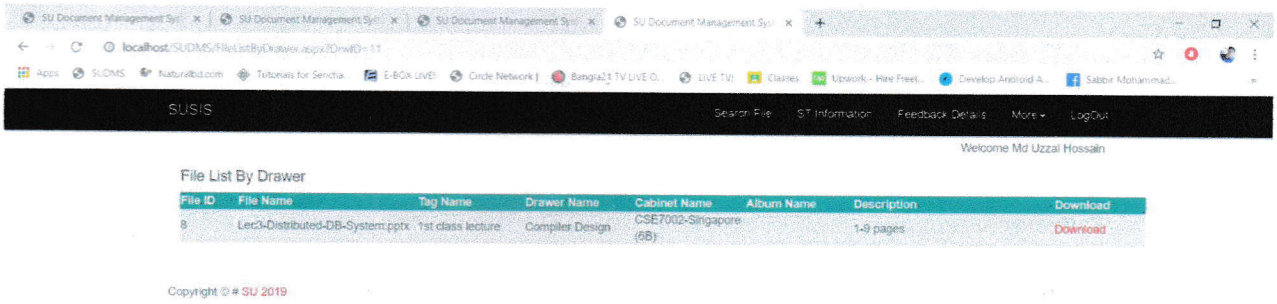


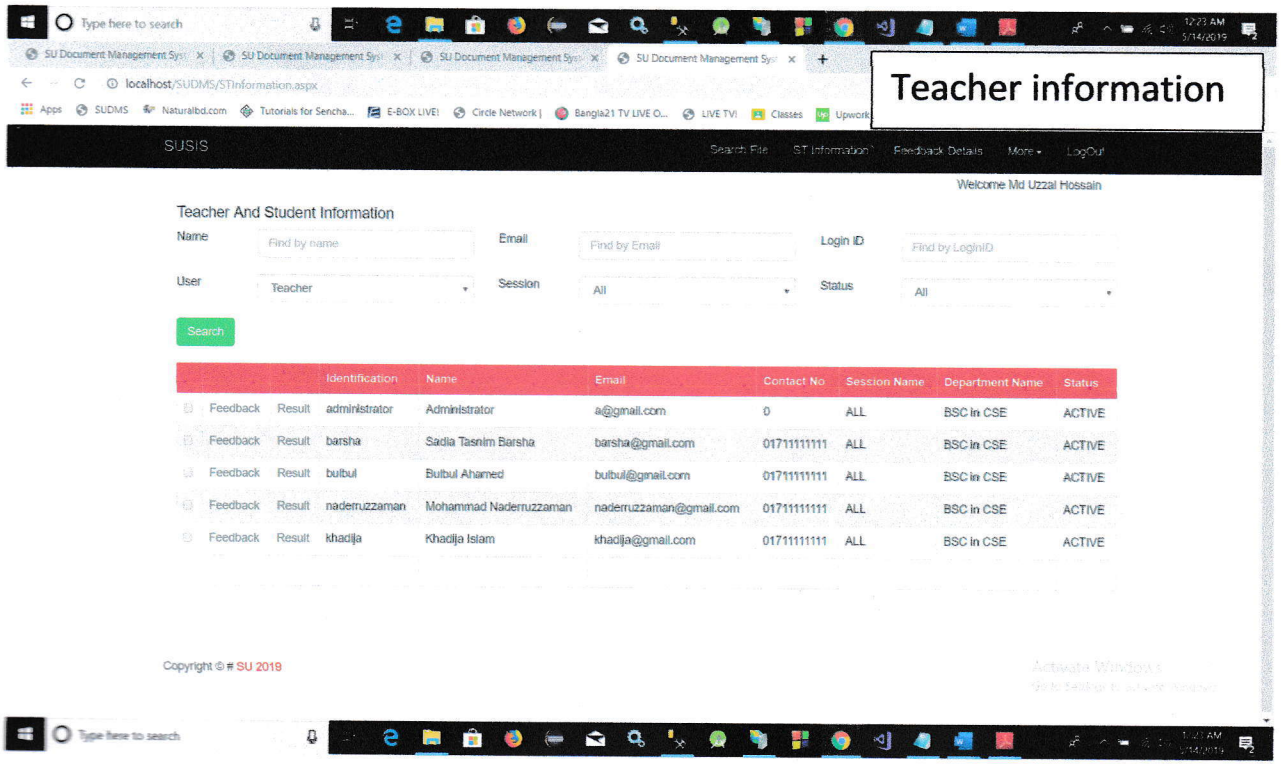
Own Session wise documents



Course wise documents

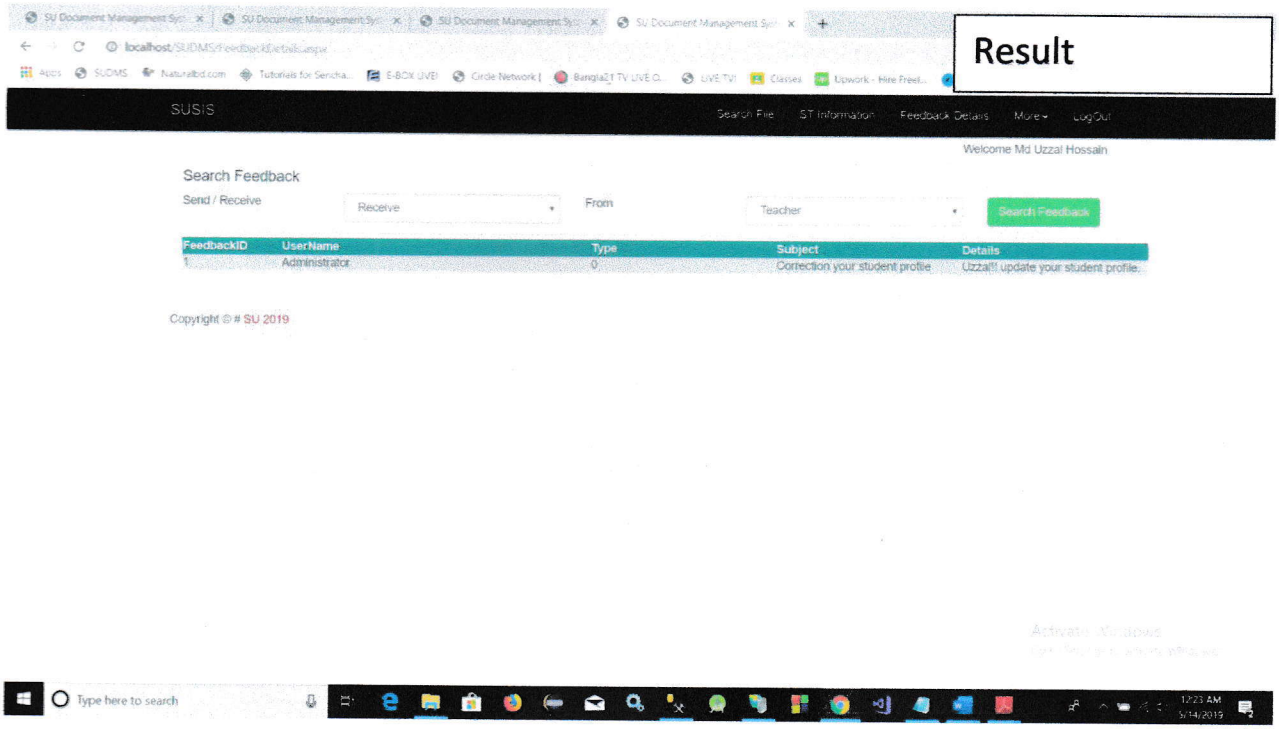
Download files


SUSIS
Search File
ST Information
Feedback Details
More
LogOut
Welcome Md Uzzal Hossain
File List By Drawer
File ID
File Name
Tag Name
Drawer Name
Cabinet Name
Album Name
Description
Download
8
Lec3-Distributed-DB-System.pptx
1st class lecture
Compiler Design
CSE7002-Singapore (6B)
1-9 pages
Download
Copyright © # SU 2019

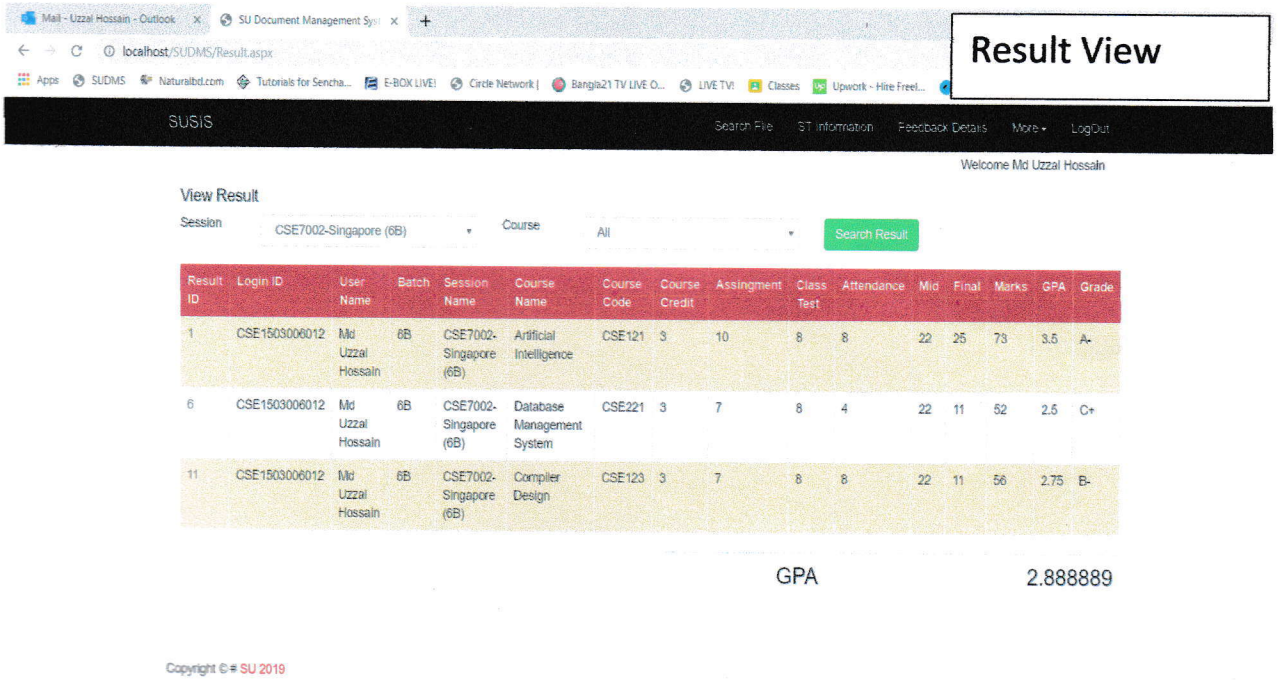

Teacher information
SUSIS
Search File
ST Information
Feedback Details
More
LogOut
Welcome Md Uzzal Hossain
Teacher And Student Information
Name
Find by name
Email
Find by Email
Login ID
Find by LoginID
User
Teacher
Session
All
Status
All
Search
Identification
Name
Email
Contact No
Session Name
Department Name
Status
Feedback Result administrator Administrator a@gmail.com 0 ALL BSC in CSE ACTIVE
Feedback Result barsha Sadia Tasnim Barsha barsha@gmail.com 01711111111 ALL BSC in CSE ACTIVE
Feedback Result bulbul Bulbul Ahamed bulbul@gmail.com 01711111111 ALL BSC in CSE ACTIVE
Feedback Result naderruzzaman Mohammad Naderruzzaman naderruzzaman@gmail.com 01711111111 ALL BSC in CSE ACTIVE
Feedback Result khadija Khadija Islam khadija@gmail.com 01711111111 ALL BSC in CSE ACTIVE
Copyright © # SU 2019

Feedback

Result



Result View

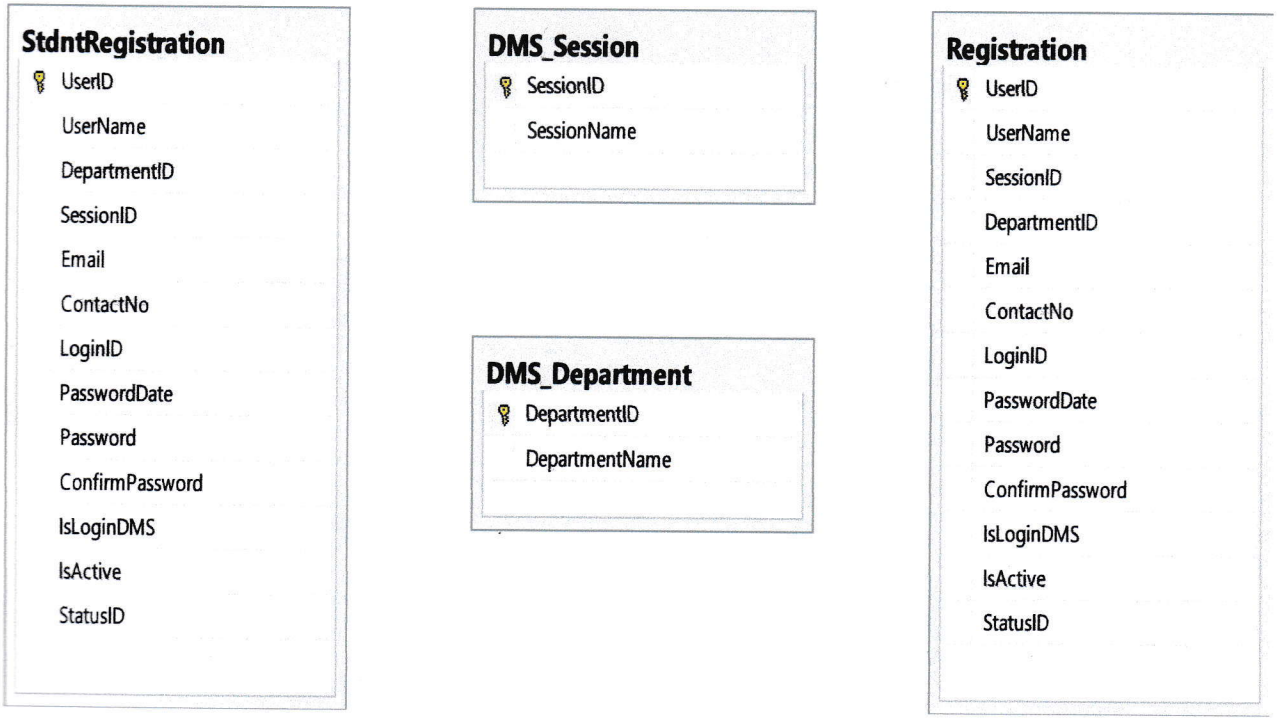


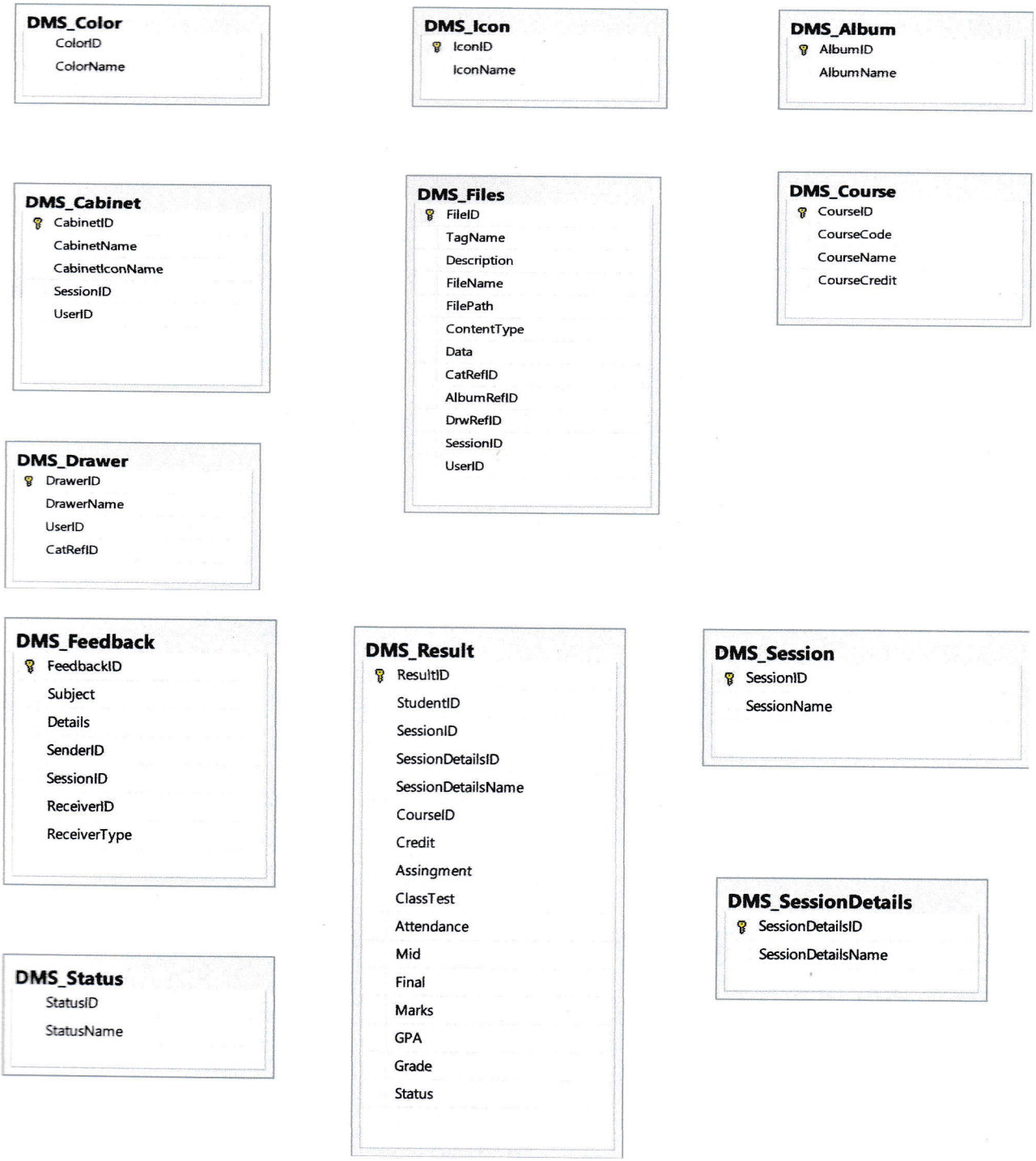
Chapter 5

Database Design and Development

5.1 Database design

ER Diagram




DMS_Color
ColorID
ColorName
DMS_Icon
IconID
IconName
DMS_Album
AlbumID
AlbumName
DMS_Cabinet
CabinetID
CabinetName
CabinetIconName
SessionID
UserID
DMS_Files
FileID
TagName
Description
FileName
FilePath
ContentType
Data
CatRefID
AlbumRefID
DrwRefID
SessionID
UserID
DMS_Course
CourseID
CourseCode
CourseName
CourseCredit
DMS_Drawer
DrawerID
DrawerName
UserID
CatRefID
DMS_Feedback
FeedbackID
Subject
Details
SenderID
SessionID
ReceiverID
ReceiverType
DMS_Result
ResultID
StudentID
SessionID
SessionDetailsID
SessionDetailsName
CourseID
Credit
Assingment
ClassTest
Attendance
Mid
Final
Marks
GPA
Grade
Status
DMS_Session
SessionID
SessionName
DMS_SessionDetails
SessionDetailsID
SessionDetailsName
DMS_Status
StatusID
StatusName

5.2 Database development

Database creation script

```
CREATE DATABASE DMS
```

Database table creation scripts

```
/****** Object:  Table [dbo].[Registration]    Script Date: 5/13/2019 8:36:51 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Registration](
	[UserID] [int] IDENTITY(1,1) NOT FOR REPLICATION NOT NULL,
	[UserName] [varchar](50) NULL,
	[SessionID] [int] NULL,
	[DepartmentID] [int] NULL,
	[Email] [varchar](50) NULL,
	[ContactNo] [varchar](50) NULL,
	[LoginID] [varchar](50) NULL,
	[PasswordDate] [date] NULL,
	[Password] [nvarchar](50) NULL,
	[ConfirmPassword] [nvarchar](50) NULL,
	[IsLoginDMS] [bit] NULL,
	[IsActive] [bit] NULL,
	[StatusID] [int] NULL,
 CONSTRAINT [PK_Registration] PRIMARY KEY CLUSTERED
(
	[UserID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[StdntRegistration]    Script Date: 5/13/2019 8:36:51 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[StdntRegistration](
	[UserID] [int] IDENTITY(1,1) NOT NULL,
	[UserName] [varchar](50) NULL,
```

```
	[DepartmentID] [int] NULL,
	[SessionID] [int] NULL,
	[Email] [varchar](50) NULL,
	[ContactNo] [varchar](50) NULL,
	[LoginID] [varchar](50) NULL,
	[PasswordDate] [date] NULL,
	[Password] [nvarchar](50) NULL,
	[ConfirmPassword] [nvarchar](50) NULL,
	[IsLoginDMS] [bit] NULL,
	[IsActive] [bit] NULL,
	[StatusID] [int] NULL,
 CONSTRAINT [PK_StdntRegistration] PRIMARY KEY CLUSTERED
(
	[UserID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO



/****** Object:  Table [dbo].[DMS_Feedback]    Script Date: 5/13/2019 8:36:51 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[DMS_Feedback](
	[FeedbackID] [int] IDENTITY(1,1) NOT NULL,
	[Subject] [varchar](100) NULL,
	[Details] [varchar](max) NULL,
	[SenderID] [int] NULL,
	[SessionID] [int] NULL,
	[ReceiverID] [int] NULL,
	[ReceiverType] [varchar](50) NULL,
 CONSTRAINT [PK_DMS_Feedback] PRIMARY KEY CLUSTERED
(
	[FeedbackID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[DMS_Files]    Script Date: 5/13/2019 8:36:51 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
```

```
GO
CREATE TABLE [dbo].[DMS_Files](
	[FileID] [int] IDENTITY(1,1) NOT NULL,
	[TagName] [varchar](50) NULL,
	[Description] [varchar](100) NULL,
	[FileName] [varchar](50) NULL,
	[FilePath] [varchar](500) NULL,
	[ContentType] [nvarchar](50) NULL,
	[Data] [varbinary](max) NULL,
	[CatRefID] [int] NULL,
	[AlbumRefID] [int] NULL,
	[DrwRefID] [int] NULL,
	[SessionID] [int] NULL,
	[UserID] [int] NULL,
 CONSTRAINT [PK_DMS_Files] PRIMARY KEY CLUSTERED
(
	[FileID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[DMS_Result]    Script Date: 5/13/2019 8:36:51 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[DMS_Result](
	[ResultID] [int] IDENTITY(1,1) NOT NULL,
	[StudentID] [int] NULL,
	[SessionID] [int] NULL,
	[SessionDetailsID] [int] NULL,
	[SessionDetailsName] [varchar](50) NULL,
	[CourseID] [int] NULL,
	[Credit] [float] NULL,
	[Assingment] [int] NULL,
	[ClassTest] [int] NULL,
	[Attendance] [int] NULL,
	[Mid] [int] NULL,
	[Final] [int] NULL,
	[Marks]  AS ((((([Assingment]+[ClassTest])+[Attendance])+[Mid])+[Final]),
	[GPA] [float] NULL,
	[Grade] [varchar](10) NULL,
	[Status] [bit] NULL,
 CONSTRAINT [PK_DMS_Result] PRIMARY KEY CLUSTERED
(
	[ResultID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

```
SET ANSI_PADDING OFF
GO
```

Store procedure creation scripts

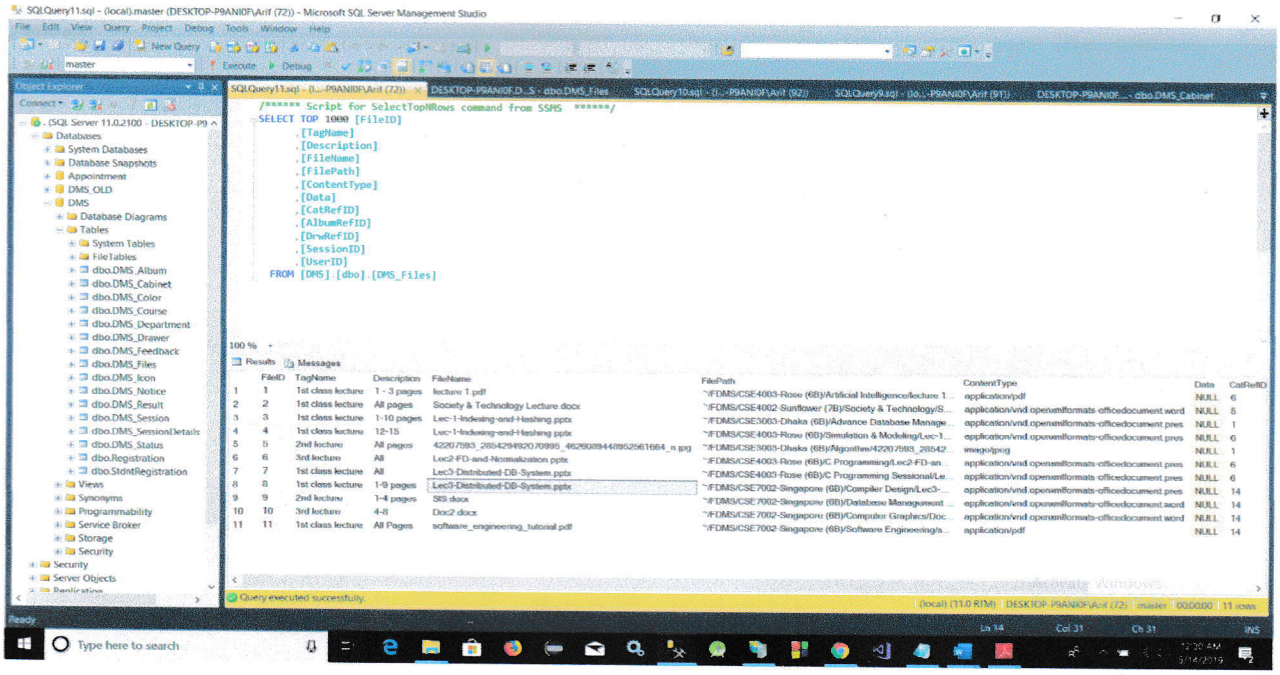
```
USE [DMS]
GO
/****** Object:  StoredProcedure [dbo].[DMS_GetAllFile]    Script Date: 5/13/2019 8:36:51 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DMS_GetAllFile]
AS
BEGIN

	SELECT
			FileID,
			TagName,
			[Description],
			[FileName],
			ContentType,
			Data,
			DR.DrawerName,
			C.CabinetName,
			A.AlbumName

	 FROM [dbo].[DMS_Files] D
	 LEFT JOIN [dbo].[DMS_Cabinet] C on C.CabinetID = D.CatRefID
	 LEFT JOIN [dbo].[DMS_Album] A on A.AlbumID = D.AlbumRefID
	 LEFT JOIN [dbo].[DMS_Drawer] DR on DR.DrawerID = D.DrwRefID

	 ORDER BY FileID DESC

	SET NOCOUNT ON;

END

GO
/****** Object:  StoredProcedure [dbo].[DMS_GetResultMarkSheetByResultID]    Script Date: 5/13/2019 8:36:51 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DMS_GetResultMarkSheetByResultID]
	@ResultID int
AS
BEGIN
```

```
        SELECT  [ResultID]
              ,S.LoginID
              ,S.UserName
              ,SE.SessionName Batch
              ,SD.SessionDetailsName
      ,C.CourseName
              ,C.CourseCode
              ,C.CourseCredit
      ,[Assingment]
      ,[ClassTest]
      ,[Attendance]
      ,[Mid]
      ,[Final]
      ,[Marks]
      ,[GPA]
      ,[Grade]
  FROM [DMS].[dbo].[DMS_Result] R
  LEFT JOIN DMS.dbo.StdntRegistration S ON S.UserID=R.StudentID
  LEFT JOIN DMS.dbo.DMS_Course C ON C.CourseID=R.CourseID
  LEFT JOIN DMS.dbo.DMS_SessionDetails SD ON SD.SessionDetailsID=R.SessionDetailsID
  LEFT JOIN DMS.dbo.DMS_Session SE ON SE.SessionID=S.SessionID

  WHERE R.ResultID=@ResultID



          --ORDER BY FileID DESC

        SET NOCOUNT ON;


END

GO
/****** Object:  StoredProcedure [dbo].[DMS_SaveResult]    Script Date: 5/13/2019 8:36:51 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO


CREATE PROCEDURE [dbo].[DMS_SaveResult]
                (

                        @xmlResult                      xml,
                        @SessionDetailsIDint,
                        @CourseID                       int,
                        @ResultID                       int     OUTPUT

                )

AS
BEGIN
```

```
BEGIN TRANSACTION
        BEGIN
INSERT INTO [DMS].[dbo].[DMS_Result]
(

                StudentID,
                SessionDetailsID,
                CourseID,
                Assingment,
                ClassTest,
                Attendance,
                Mid,
                Final,
                [Status]


)
        SELECT   doc.c.value('@StdID', 'int'),
        @SessionDetailsID,
  @CourseID,
        doc.c.value('@Assingment','int'),
        doc.c.value('@ClassTest','int'),
        doc.c.value('@Attendance','int'),
        doc.c.value('@Mid','int'),
        doc.c.value('@Final','int'),
        0

        FROM @xmlResult.nodes('/publishResult/results')doc(c)

        SET @ResultID = @@IDENTITY
        END
COMMIT TRANSACTION

BEGIN TRANSACTION
        UPDATE DMS_Result
        SET Grade = CASE
        WHEN Marks >= 80 THEN 'A+'
        WHEN Marks>=75 AND Marks<80 THEN 'A'
        WHEN Marks>=70 AND Marks<75 THEN 'A-'
        WHEN Marks>=65 AND Marks<70 THEN 'B+'
        WHEN Marks>=60 AND Marks<65 THEN 'B'
        WHEN Marks>=55 AND Marks<60 THEN 'B-'
        WHEN Marks>=50 AND Marks<55 THEN 'C+'
        WHEN Marks>=45 AND Marks<50 THEN 'C'
        WHEN Marks>=40 AND Marks<45 THEN 'D'
        ELSE 'F' END,
        GPA = CASE
        WHEN Marks >= 80 THEN 4.00
        WHEN Marks>=75 AND Marks<80 THEN 3.75
        WHEN Marks>=70 AND Marks<75 THEN 3.50
        WHEN Marks>=65 AND Marks<70 THEN 3.25
        WHEN Marks>=60 AND Marks<65 THEN 3.00
        WHEN Marks>=55 AND Marks<60 THEN 2.75
        WHEN Marks>=50 AND Marks<55 THEN 2.50
        WHEN Marks>=45 AND Marks<50 THEN 2.25
        WHEN Marks>=40 AND Marks<45 THEN 2.00
```

```
        ELSE 0.00 END
        WHERE GRADE IS NULL

    COMMIT TRANSACTION
END
```

Database table with value



Chapter 6

Project Management

6.1 Work breakdown structure

The Work Breakdown Structure provides a clear picture of the items that are required to be completed for the project work scope. It is arranged in a tree structure manner to show all sub-sections required to be accomplished in each phase throughout the project.

6.2 Risk management

Every project has uncertainties and they may affect the project success. These uncertainties are known as risk. There is a need to determine the risks involved in the project to reduce the chance of its occurrence as well as develop a plan to either mitigate or avoid the risks that have been identified.

After identifying and determining the risks, they are recorded in a table called the risk assessment matrix. This matrix lists the severity which is also known as the impact level in terms of High, Medium and Low. In addition to those details, the action plan that can be taken for each of the risks is clearly stated too.

Risk Event	Consequence	Probability	Impact	Risk Response Plan
Incompatible software application with the operating system	Delay in system implementation	Medium	Medium	a) To source for other software applications that is compatible with the operating system b) To find another workstation
User requirements keep changing	More time is required as modification is to be done throughout the whole project period thus unable to meet expected deadline of each phase	Medium	High	a) To check with user on the requirements frequently b) To allocate more time for the development/implementation and testing stage
Difficulties with the coding	Delay in system implementation	Medium	Medium	a) To research more on PHP language and MySQL via sources such as books and tutorial websites
User do not accept the system	Unable to satisfy objectives and complete the project	Low	High	a) Make a check with the user after implementation to ensure the system functioned as per request

Figure 6.2: Risk Assessment Matrix

Chapter 7

Conclusions

7.1 Conclusions

It is concluded that automation of existing student information system, such as the delivery of enrolment procedures and keeping the records of students information such as: keeping of admission requirements during enrolment, personal information, student subjects enrolled and class schedules, and knowing the overall performance of students will maximize the utilization of the full range of benefits of Information and Communications Technology.

In a nutshell, all the stakeholders, such as the faculty members, students and the administrators found that the developed Student Information System had met their expectations with regards to the student information system.

7.2 Limitation

- Without internet the system does not work.

7.3 Future work

- Student will be able to want their credit completion status.
- Student will be able to request get extra credit.
- Manage student fee payment and discounts. Information on if the student needs financial aid and other debt management.
- Manage all information related to rooms, mess, goods, etc. Track details of menu, reduce food expenses and optimize room allocation.

- Maintain student's every day attendance period wise. Generate attendance related reports on-demand for the stakeholders.

- It will be including questionaries' management system.

References

1. https://en.wikipedia.org/wiki/Student_information_system ,01.02.19
2. http://www.su.edu.bd/ 12.02.19
3. https://online.visual-paradigm.com/ ,01.03.19
4. https://www.w3schools.com/ ,11.02.19
5. http://stackoverflow.com/documentation/sql/topics ,17.02.19
6. http://sue786.su.edu.bd/sonargaon_erp/ ,03.04.19
7. https://www.tutorialspoint.com/ ,01.02.19
8. https://www.codecademy.com/ ,01.05.19
9. https://www.academia.edu/20198034/STUDENT_INFORMATION_SYSTEM_ITS_EFFICACY_TOWARDS_STUDENT_TRACKING_PROGRAM ,01.02.19
10. https://opensis.com/ ,09.02.19
11. https://www.tutorialspoint.com/asp.net/ ,25.04.19
12. https://www.c-sharpcorner.com/technologies/asp-dot-net-programming ,05.02.19
13. https://www.tutorialspoint.com/sql/ ,05.05.19

Appendix

Source Code

Student Registration

```
using SUDMS.DAC;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace SUDMS.RegLogin
{
    public partial class StdntRegistration : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                BindDepartmentList();
                BindSessionList();


            }
        }

        private void BindSessionList()
        {
            AppDB db = new AppDB();
            ddlSession.DataSource = db.GetSession();
            ddlSession.DataBind();
            ddlSession.Items.Add(new System.Web.UI.WebControls.ListItem("All", "0"));
            ddlSession.SelectedValue = "0";
        }

        private void BindDepartmentList()
        {
            AppDB db = new AppDB();
            ddlDepartment.DataSource = db.GetDepartment();
            ddlDepartment.DataBind();
        }

        protected void Regbtn_Click(object sender, EventArgs e)
        {
            string UserName = txtUserName.Text;
            int DepartmentID = Convert.ToInt32(ddlDepartment.SelectedValue);
            int SessionID = Convert.ToInt32(ddlSession.SelectedValue);
            string Email = txtEmail.Text;
            string ContactNo = txtContactNo.Text;
```

```
            string LoginID = txtLoginID.Text;
            string Password = txtPassword.Text;
            string ConfirmPassword = txtConfirmPassword.Text;
            lblMessage.Text = "";
            if (Page.IsValid)
            {

                if (txtPassword.Text != txtConfirmPassword.Text)
                {
                    lblMessage.Text = "Password was not confirmed.";
                }

                else
                {

                    UserDB db = new UserDB();
                    try
                    {
                        db.InsertStdntUser(UserName, DepartmentID, SessionID,
Email,ContactNo, LoginID, Password, ConfirmPassword);
                        txtUserName.Text = "";
                        txtEmail.Text = "";
                        txtLoginID.Text = "";
                        lblMessage.Text = "Registration complete";
                        lblMessage.CssClass = "alert-success";

                    }

                    catch (SqlException ex)
                    {
                        //lblMessage.Text = "Connection Failed";
                        lblMessage.Text = ex.Message;
                    }
                }
            }
        }
    }
}
```

Result Published

```
using SUDMS.DAC;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.IO;
using System.Linq;
using System.Text;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Xml;

namespace SUDMS
{
```

```
public partial class WebForm14 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            BindUser();
            BindSessionList();
            BindCabinetList();
            BindCourse();

        }
    }

    private void BindCourse()
    {
        AppDB db = new AppDB();
        ddlCourse.DataSource = db.GetCourse();
        ddlCourse.DataBind();
    }

    private void BindCabinetList()
    {
        AppDB db = new AppDB();
        ddlCabinet.DataSource = db.GetCabinet();
        ddlCabinet.DataBind();
    }

    private void BindUser()
    {
        UserDB db = new UserDB();
        MyDataGrid1.DataSource = db.BindUser();
        MyDataGrid1.DataBind();

    }

    private void BindSessionList()
    {
        UserDB db = new UserDB();
        ddlSession.DataSource = db.GetSessionSearch();
        ddlSession.DataBind();
        ddlSession.Items.Add(new System.Web.UI.WebControls.ListItem("All", "0"));
        ddlSession.SelectedValue = "0";
    }

    protected void btnFind_Click(object sender, EventArgs e)
    {
        int SessionID = Int32.Parse(ddlSession.SelectedItem.Value);
        AppDB db = new AppDB();
        MyDataGrid1.DataSource = db.GetStudentBySessionID(SessionID);
        MyDataGrid1.DataBind();

    }

    public void btnSaveResult_Click(object sender, EventArgs e)
    {

        int CabinetID = Int32.Parse(ddlCabinet.SelectedValue);
```

```
            int CourseID = Int32.Parse(ddlCourse.SelectedItem.Value);

            StringBuilder xmlResult = new StringBuilder();
            using (XmlWriter writer = new XmlTextWriter(new StringWriter(xmlResult)))
            {
                writer.WriteStartElement("publishResult");
                foreach (DataGridItem row in MyDataGrid1.Items)
                {
                    writer.WriteStartElement("results");

                    string UserID = MyDataGrid1.DataKeys[row.ItemIndex].ToString();


                    TextBox txtAssingment = row.Cells[1].FindControl("txtAssingment") as
TextBox;
                    TextBox txtClassTest = row.Cells[2].FindControl("txtClassTest") as
TextBox;
                    TextBox txtAttendance = row.Cells[3].FindControl("txtAttendance") as
TextBox;
                    TextBox txtMid = row.Cells[4].FindControl("txtMid") as TextBox;
                    TextBox txtFinal = row.Cells[5].FindControl("txtFinal") as TextBox;


                    //writer.WriteStartElement("minnute");
                    //writer.WriteAttributeString("MinID",
dk.Values["MinID"].ToString());
                    writer.WriteAttributeString("StdID", UserID.Trim().ToString());

                    writer.WriteAttributeString("Assingment",
txtAssingment.Text.Trim().ToString());
                    writer.WriteAttributeString("ClassTest",
txtClassTest.Text.Trim().ToString());
                    writer.WriteAttributeString("Attendance",
txtAttendance.Text.Trim().ToString());
                    writer.WriteAttributeString("Mid", txtMid.Text.Trim().ToString());
                    writer.WriteAttributeString("Final",
txtFinal.Text.Trim().ToString());
                    writer.WriteEndElement();
                }
                writer.WriteEndElement();

            }
            AppDB db = new AppDB();
            db.SaveResult(xmlResult.ToString(), CabinetID, CourseID);
            lblMessage.Text = "Successfully result save";
            BindUser();


    }
}
```

Results View

```
using SUDMS.DAC;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace SUDMS
{
    public partial class WebForm13 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                //BindUser();
                //BindSessionList();
                BindCabinetList();
                BindCourse();
                int UserID = Int32.Parse(Request.Cookies["UserID"].Value);
                int SessionID = Int32.Parse(Request.Cookies["SessionID"].Value);
                if (SessionID == 0)
                {
                    BindResult();
                }
                else
                {
                    BindResultByStudent(UserID);
                }


            }
        }

        private void BindResultByStudent(int UserID)
        {
            AppDB db = new AppDB();
            GridView2.DataSource = db.BindResultByStudent(UserID);
            GridView2.DataBind();
        }

        private void BindResult()
        {
            AppDB db = new AppDB();
            GridView2.DataSource = db.BindResult();
            GridView2.DataBind();
        }

        private void BindCourse()
        {
            AppDB db = new AppDB();
            ddlCourse.DataSource = db.GetCourse();
            ddlCourse.DataBind();
            ddlCourse.Items.Add(new System.Web.UI.WebControls.ListItem("All", "0"));
            ddlCourse.SelectedValue = "0";
```

```
        }

        private void BindCabinetList()
        {
            AppDB db = new AppDB();
            ddlCabinet.DataSource = db.GetCabinet();
            ddlCabinet.DataBind();
            //ddlCabinet.Items.Add(new System.Web.UI.WebControls.ListItem("All", "0"));
            //ddlCabinet.SelectedValue = "0";
        }


        protected void GridView1_PageIndexChanging(object sender, GridViewPageEventArgs
e)
        {
            GridView2.PageIndex = e.NewPageIndex;
            BindResult();
            GridView2.DataBind();
        }

        protected void GridView2_RowCancelingEdit(object sender,
GridViewCancelEditEventArgs e)
        {
            GridView2.EditIndex = -1;
            BindResult();
        }

        protected void GridView2_RowDeleting(object sender, GridViewDeleteEventArgs e)
        {


        }

        protected void GridView2_RowEditing(object sender, GridViewEditEventArgs e)
        {
            GridView2.EditIndex = e.NewEditIndex;
            BindResult();
        }

        protected void GridView2_RowUpdating(object sender, GridViewUpdateEventArgs e)
        {

        }

        protected void GridView1_RowCancelingEdit(object sender,
GridViewCancelEditEventArgs e)
        {
            GridView2.EditIndex = -1;
            BindResult();
        }

        protected void SearchResult_Click(object sender, EventArgs e)
        {
            int UserID = Int32.Parse(Request.Cookies["UserID"].Value);
            int SessionID = Int32.Parse(Request.Cookies["SessionID"].Value);
            int CabinetID = Int32.Parse(ddlCabinet.SelectedItem.Value);
            int CourseID = Int32.Parse(ddlCourse.SelectedItem.Value);
            AppDB db = new AppDB();
```

```
            if(SessionID==0)
            {
                GridView2.DataSource = db.SearchResult(CabinetID, CourseID);
                GridView2.DataBind();
            }
            else
            {
                GridView2.DataSource = db.SearchResultByStudent(CabinetID,
CourseID,UserID);
                GridView2.DataBind();
            }


        }


    }
}
```

Create table and store procedure

```
USE [DMS]
GO
/****** Object:  Table [dbo].[DMS_Result]    Script Date: 5/15/2019 11:18:02 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[DMS_Result](
	[ResultID] [int] IDENTITY(1,1) NOT NULL,
	[StudentID] [int] NULL,
	[SessionID] [int] NULL,
	[SessionDetailsID] [int] NULL,
	[SessionDetailsName] [varchar](50) NULL,
	[CourseID] [int] NULL,
	[Credit] [float] NULL,
	[Assingment] [int] NULL,
	[ClassTest] [int] NULL,
	[Attendance] [int] NULL,
	[Mid] [int] NULL,
	[Final] [int] NULL,
	[Marks]  AS ((((([Assingment]+[ClassTest])+[Attendance])+[Mid])+[Final]),
	[GPA] [float] NULL,
	[Grade] [varchar](10) NULL,
	[Status] [bit] NULL,
 CONSTRAINT [PK_DMS_Result] PRIMARY KEY CLUSTERED
(
	[ResultID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
GO
SET ANSI_PADDING OFF

GO
/****** Object:  StoredProcedure [dbo].[DMS_GetResultByStudent]    Script Date: 5/15/2019
11:18:02 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DMS_GetResultByStudent]
      @UserID                     int
AS
BEGIN

      SELECT  [ResultID]
        ,S.LoginID
        ,S.UserName
        ,SE.SessionName Batch
        ,CB.CabinetName
      ,C.CourseName
        ,C.CourseCode
        ,C.CourseCredit
      ,[Assingment]
      ,[ClassTest]
      ,[Attendance]
      ,[Mid]
      ,[Final]
      ,[Marks]
      ,[GPA]
      ,[Grade]
  FROM [DMS].[dbo].[DMS_Result] R
  LEFT JOIN DMS.dbo.StdntRegistration S ON S.UserID=R.StudentID
  LEFT JOIN DMS.dbo.DMS_Course C ON C.CourseID=R.CourseID
  LEFT JOIN DMS.dbo.DMS_Cabinet CB ON CB.CabinetID=R.SessionDetailsID
  LEFT JOIN DMS.dbo.DMS_Session SE ON SE.SessionID=S.SessionID
  WHERE R.StudentID=@UserID


        --ORDER BY FileID DESC

      SET NOCOUNT ON;


END

GO
/****** Object:  StoredProcedure [dbo].[DMS_SaveResult]    Script Date: 5/15/2019
11:18:02 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO


CREATE PROCEDURE [dbo].[DMS_SaveResult]
```

```
(
        @xmlResult              xml,
        @CabinetID              int,
        @CourseID               int,
        @ResultID               int     OUTPUT
)
AS
BEGIN
    BEGIN TRANSACTION
        BEGIN
     INSERT INTO [DMS].[dbo].[DMS_Result]
     (
            StudentID,
            SessionDetailsID,
            CourseID,
            Assingment,
            ClassTest,
            Attendance,
            Mid,
            Final,
            [Status]

    )
        SELECT doc.c.value('@StdID', 'int'),
        @CabinetID,
      @CourseID,
        doc.c.value('@Assingment','int'),
        doc.c.value('@ClassTest','int'),
        doc.c.value('@Attendance','int'),
        doc.c.value('@Mid','int'),
        doc.c.value('@Final','int'),
        0

        FROM @xmlResult.nodes('/publishResult/results')doc(c)

        SET @ResultID = @@IDENTITY
        END
    COMMIT TRANSACTION

    BEGIN TRANSACTION

        UPDATE DMS_Result
        SET Grade = CASE
        WHEN Marks >= 80 THEN 'A+'
        WHEN Marks>=75 AND Marks<80 THEN 'A'
        WHEN Marks>=70 AND Marks<75 THEN 'A-'
        WHEN Marks>=65 AND Marks<70 THEN 'B+'
        WHEN Marks>=60 AND Marks<65 THEN 'B'
        WHEN Marks>=55 AND Marks<60 THEN 'B-'
        WHEN Marks>=50 AND Marks<55 THEN 'C+'
        WHEN Marks>=45 AND Marks<50 THEN 'C'
        WHEN Marks>=40 AND Marks<45 THEN 'D'
        ELSE 'F' END,
```

```
            GPA = CASE
            WHEN Marks >= 80 THEN 4.00
            WHEN Marks>=75 AND Marks<80 THEN 3.75
            WHEN Marks>=70 AND Marks<75 THEN 3.50
            WHEN Marks>=65 AND Marks<70 THEN 3.25
            WHEN Marks>=60 AND Marks<65 THEN 3.00
            WHEN Marks>=55 AND Marks<60 THEN 2.75
            WHEN Marks>=50 AND Marks<55 THEN 2.50
            WHEN Marks>=45 AND Marks<50 THEN 2.25
            WHEN Marks>=40 AND Marks<45 THEN 2.00
            ELSE 0.00 END
            WHERE GRADE IS NULL

        COMMIT TRANSACTION
END


GO
```