



CSC 621/351: Database Management Systems

Professors: Mary Krueger & Alan Nochenson

DBM Project

SJU Scheduling System

Group Members:

Fatemah Alkhafaf

Julie Osborne

Jumanah Alshehri

December 6, 2016

Table of Contents

Table of Contents.....	1
Acknowledgment.....	2
Introduction.....	3
About the Purpose	3
Project Timeline.....	4
Project Ranking.....	4
Use-case Diagram.....	6
Entity Relationship Diagram.....	7
Relational Model.....	8
Functional Dependency.....	9
Table and View Creation.....	9
Index Selection	11
Constraints.....	12
Insertion Queries.....	13
System Interfaces	16
1. Homepage:	16
1.1. Homepage validations:.....	16
2. Employee Interfaces	17
2.1. Add:	18
2.2. Search:.....	27
2.3. Edit:.....	39
2.4. Edit profile:	45
3. Faculty and Student Interfaces.....	46
4. Guest Interfaces.....	53
References	56

Acknowledgment

We would like to thank Professor Mary Krueger and Professor Alan Nochenson for giving us the opportunity to work on this project and for helping us through the project steps. Also, we would like to thank our colleague Mohammed Alamoudi for helping us with the programming part.

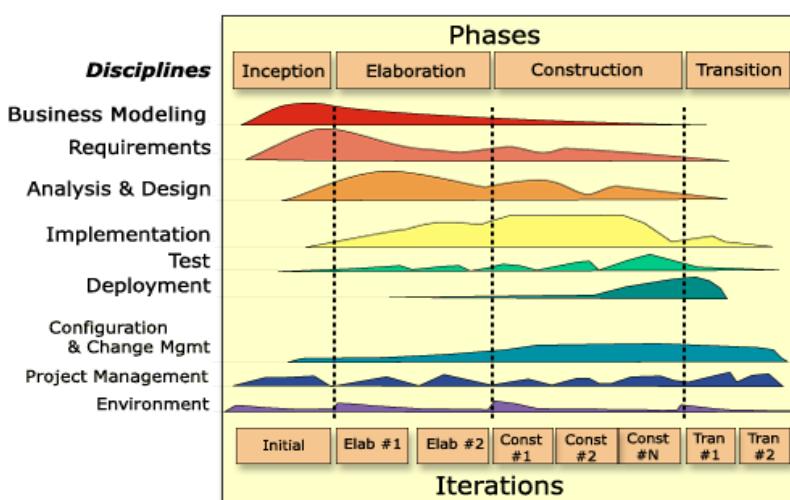
Introduction

As a requirement for the Database Management System course, we built a Database and interface for a certain system. Our idea was to build a Database for Saint Joseph's University, which will help the University Departments to schedule classes and search for students, faculty and classes. This report will summarize the project.

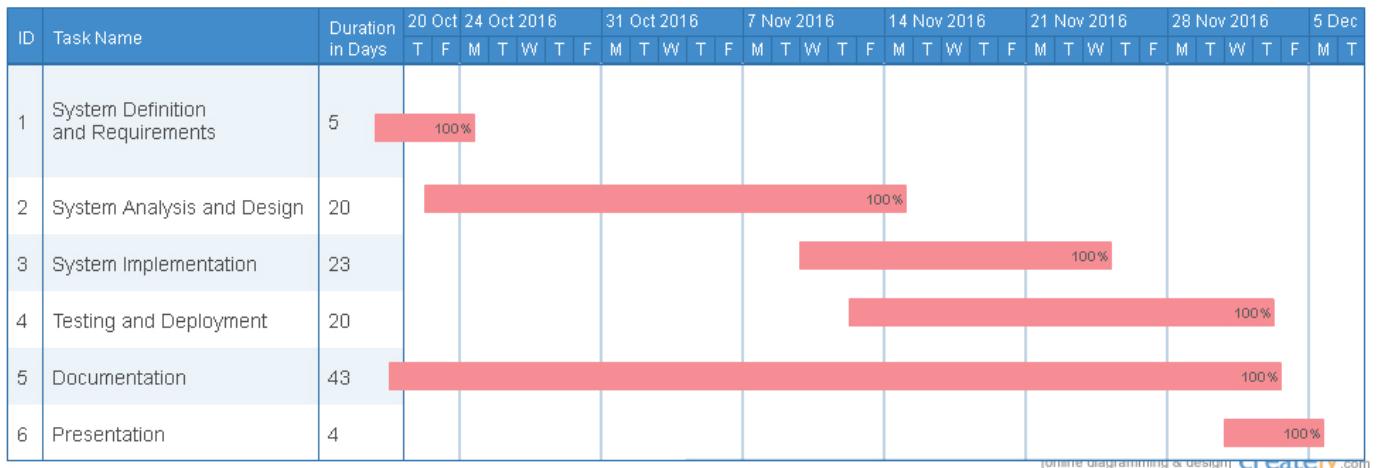
About the Purpose

The purpose of this application is to allow employees, faculty members, students, and guests of Saint Joseph's University to be able to access a scheduling system which contains information pertaining to various programs and courses. We would like this to be primarily be used for access to the information that focuses on the Computer Science department. We anticipate this application to be used among all four types of users to gain information regarding the department and its courses. Since this was the primary goal, our design was focused on making it accessible to each type of user. Each type of users has access and control to different information, so it was important in our design to have each user have a different type of portal that pertained specifically to that information. We designed our application with the purpose of being purposeful, yet simple, so it could be attainable to all different users. We hope that over time this application can contain information about all the colleges, programs, departments, and courses offered here at Saint Joe's.

For this project we analyzed our system using the Rational Unified Process (RUP), which has four different phases that are shown in the following figure.



Project Timeline



Project Ranking

- Member 1 : Fatemah Alkhafaf

My teammates and I agrees I did 33.33 % of the overall project. My specific tasks include

Task 1 : I helped define the idea

Task 2 : I helped create the ERD after discussing the system with teammates

Task 3 : I helped create the relational model and made modifications after peer review

Task 4 : I helped create the Use Case Diagram and made mods after peer review

Task 5 : I helped create the database after discussing the system with teammates

Task 6 : I helped code the form for the inserting, editing, deleting from the database

Task 7 : I tested the 2 of my teammates' forms offering suggestions on improvement

Task 8 : I helped specify constraints and indexing

Task 9: I helped write reports

Task 10 : I helped create the PowerPoint presentation.

Signature:

- Member 2 : Julie Osborne

My teammates and I agrees I did 33.33 % of the overall project. My specific tasks include

Task 1 : I helped define the idea

Task 2 : I helped create the ERD after discussing the system with teammates

Task 3 : I helped create the relational model and made mods after peer review

Task 4 : I helped create the Use Case Diagram and made mods after peer review

Task 5 : I helped create the database after discussing the system with teammates

Task 6 : I helped code the form for the inserting, editing, deleting from the database

Task 7 : I tested the 2 of my teammates' forms offering suggestions on improvement

Task 8 : I helped specify constraints and indexing

Task 9: I helped write reports

Task 10 : I helped create the PowerPoint presentation.

Signature:

- Member 3 : Juamanh Alshehri

My teammates and I agrees I did 33.33 % of the overall project. My specific tasks include

Task 1 : I helped define the idea

Task 2 : I helped create the ERD after discussing the system with teammates

Task 3 : I helped create the relational model and made mods after peer review

Task 4 : I helped create the Use Case Diagram and made mods after peer review

Task 5 : I helped create the database after discussing the system with teammates

Task 6 : I helped code the form for the inserting, editing, deleting from the database

Task 7 : I tested the 2 of my teammates' forms offering suggestions on improvement

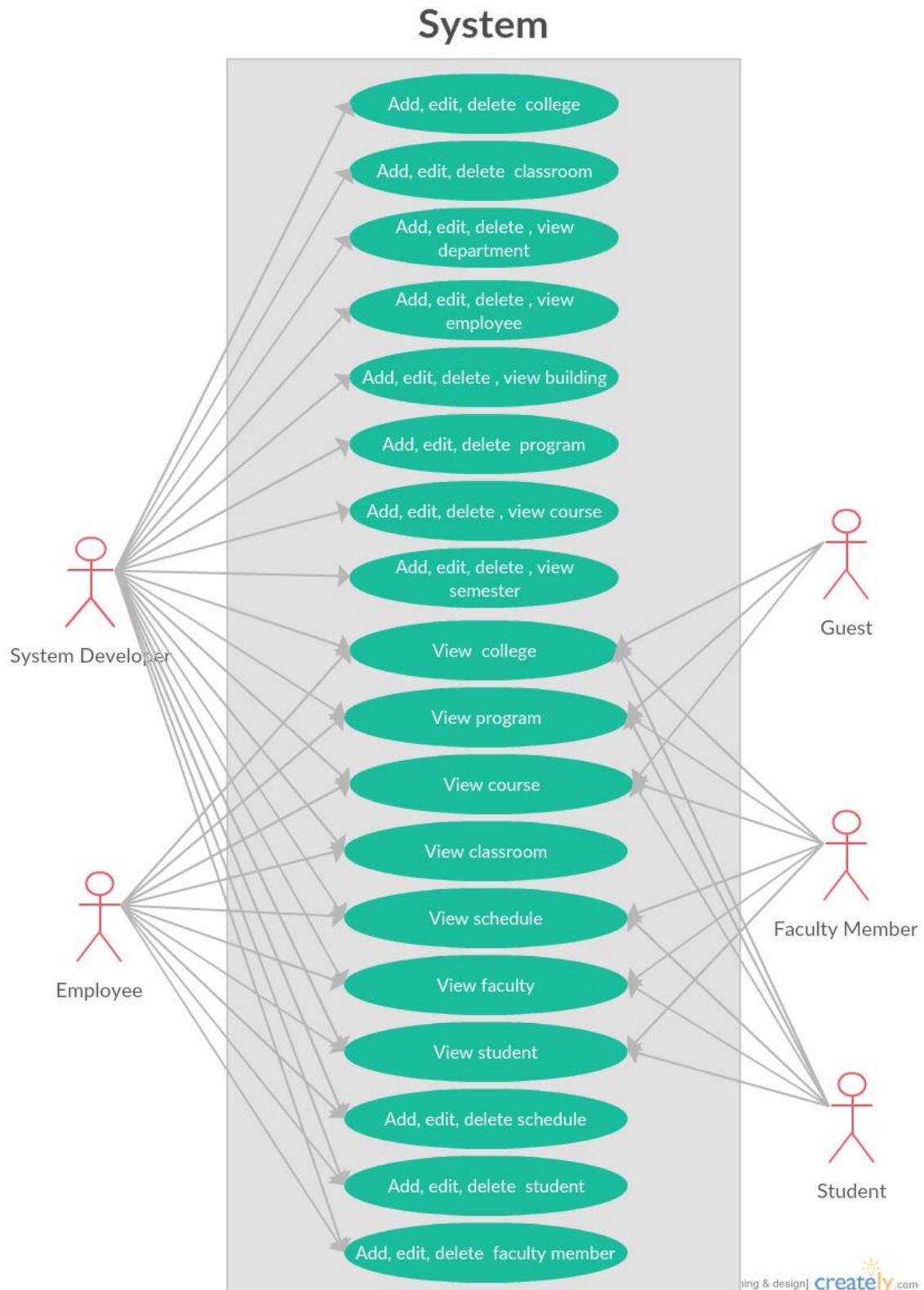
Task 8 : I helped specify constraints and indexing

Task 9: I helped write reports

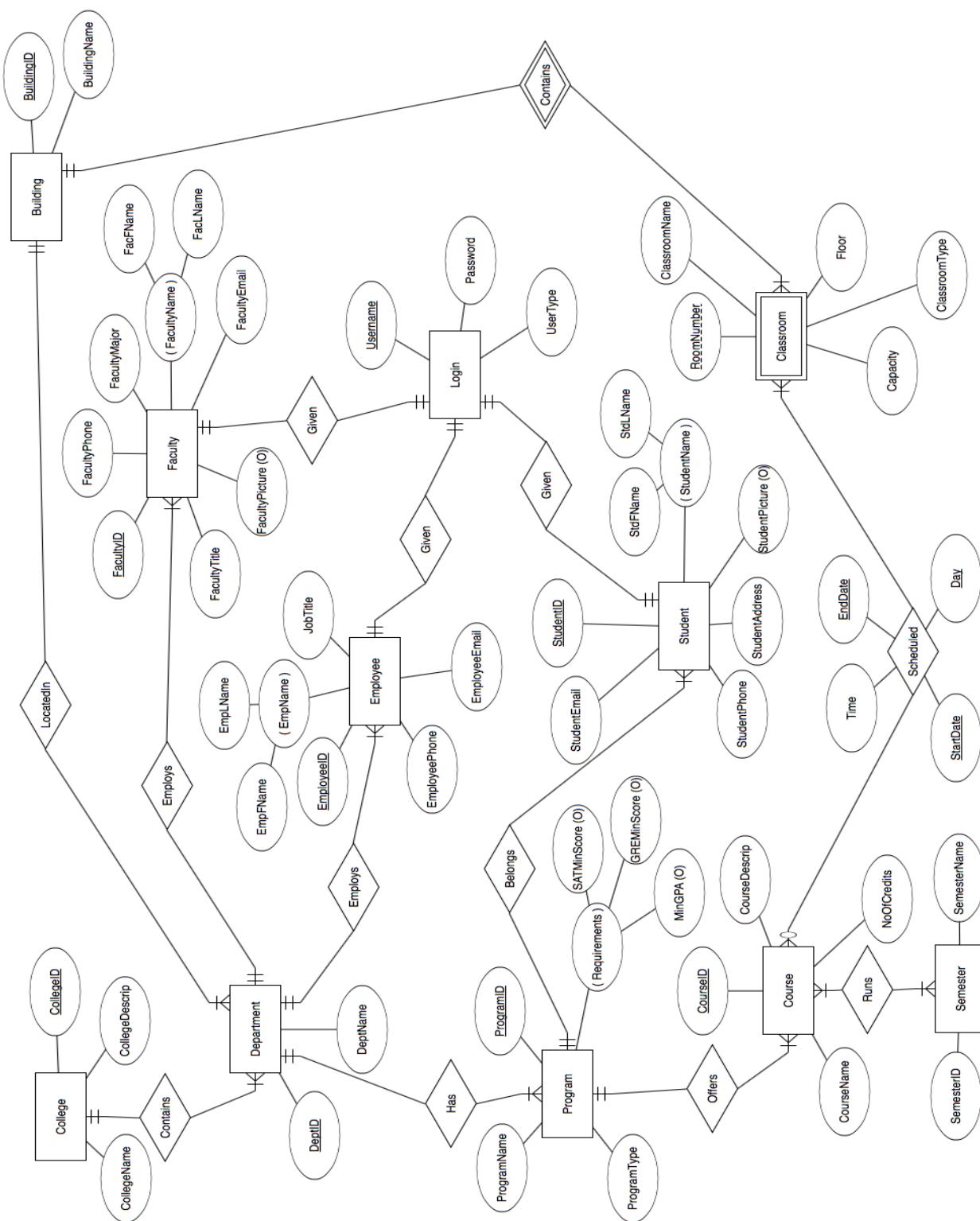
Task 10 : I helped create the PowerPoint presentation.

Signature:

Use-case Diagram



Entity Relationship Diagram



Relational Model



Functional Dependency

According to our analysis, we can say that our database is in 3rd Normalized Form. Therefore, we did avoid all types of functional dependency which they are:

- 1- Full functional dependency.
- 2- Partial functional dependency.
- 3- Transitive functional dependency.

Table and View Creation

No.	Table Name	SQL Query
1	building	CREATE TABLE `building` (`BID` varchar(5) NOT NULL, `BName` varchar(30) NOT NULL)
2	classroom	CREATE TABLE `classroom` (`BID` varchar(5) NOT NULL, `CRNum` int(3) NOT NULL, `CRName` varchar(10) NOT NULL, `Floor` int(2) NOT NULL, `CRTType` enum('Lecture','ComputerLab','ScienceLab','') NOT NULL, `Capacity` int(4) NOT NULL)
3	college	CREATE TABLE `college` (`ColgID` int(2) NOT NULL, `ColgName` varchar(50) NOT NULL, `ColgDesc` text NOT NULL)
4	course	CREATE TABLE `course` (`CID` varchar(6) NOT NULL, `CName` varchar(70) NOT NULL, `NoOfCredits` int(1) NOT NULL, `CDesc` text NOT NULL, `ProgID` int(2) NOT NULL)
5	department	CREATE TABLE `department` (`DeptID` int(3) NOT NULL, `DeptName` varchar(50) NOT NULL, `ColgID` int(2) NOT NULL, `BID` varchar(5) NOT NULL)

6	employee	CREATE TABLE `employee` (`EmpID` varchar(6) NOT NULL, `EmpFName` varchar(30) NOT NULL, `EmpLName` varchar(30) NOT NULL, `JobTitle` varchar(30) NOT NULL, `EmpPhone` varchar(10) NOT NULL DEFAULT '6106601000', `EmpEmail` varchar(30) NOT NULL, `DeptID` int(3) NOT NULL, `UName` varchar(10) NOT NULL)
7	faculty	CREATE TABLE `faculty` (`FID` varchar(6) NOT NULL, `FFName` varchar(30) NOT NULL, `FLName` varchar(30) NOT NULL, `FTitle` varchar(30) NOT NULL, `FMajor` varchar(30) NOT NULL, `FPhone` varchar(15) NOT NULL DEFAULT '6106601000', `FEmail` varchar(30) NOT NULL, `FPicture` blob, `DeptID` int(3) NOT NULL, `UName` varchar(10) NOT NULL)
8	student	CREATE TABLE `student` (`StdID` varchar(10) NOT NULL, `StdFName` varchar(30) NOT NULL, `StdLName` varchar(30) NOT NULL, `StdPhone` varchar(10) NOT NULL, `StdEmail` varchar(30) NOT NULL, `StdAddress` varchar(100) NOT NULL, `StdPicture` blob, `ProgID` int(2) NOT NULL, `UName` varchar(10) NOT NULL)
9	login	CREATE TABLE `login` (`UName` varchar(10) NOT NULL, `Pass` varchar(100) NOT NULL, `UType` varchar(10) NOT NULL)
	program	CREATE TABLE `program` (`ProgID` int(2) NOT NULL, `ProgName` varchar(50) NOT NULL, `ProgType` varchar(4) NOT NULL, `SATMinScore` int(4) DEFAULT NULL, `GREMinScore` int(3) DEFAULT NULL, `MinGPA` decimal(4,2) DEFAULT NULL, `DeptID` int(3) NOT NULL)

10	runs	CREATE TABLE `runs` (`CID` varchar(6) NOT NULL, `SID` int(5) NOT NULL)
11	schedule	CREATE TABLE `schedule` (`BID` varchar(5) NOT NULL, `CRNum` int(3) NOT NULL, `CID` varchar(6) NOT NULL, `Time` time NOT NULL, `Day` enum('Monday','Tuesday','Wednesday','Thursday','Friday') NOT NULL)
12	semester	CREATE TABLE `semester` (`SID` int(5) NOT NULL, `SName` varchar(10) NOT NULL, `Start_Date` date NOT NULL, `End_Date` date NOT NULL)
13	course_view	CREATE VIEW course_view AS (SELECT * FROM course;

Index Selection

Beside primary key index here is a list of the other indexes:

No.	Table	Indexes
1	classroom	ADD KEY `CRNum` (`CRNum`);
2	course	ADD KEY `ProgID` (`ProgID`);
3	department	ADD KEY `ColgID` (`ColgID`), ADD KEY `BID` (`BID`);
4	employee	ADD KEY `UName` (`UName`);
5	faculty	ADD KEY `DeptID` (`DeptID`), ADD KEY `UName` (`UName`);
6	student	ADD KEY `ProgID` (`ProgID`), ADD KEY `UName` (`UName`);
7	program	ADD KEY `DeptID` (`DeptID`);
8	runs	ADD KEY `SID` (`SID`);
9	schedule	ADD KEY `CID` (`CID`), ADD KEY `CRNum` (`CRNum`);

Constraints

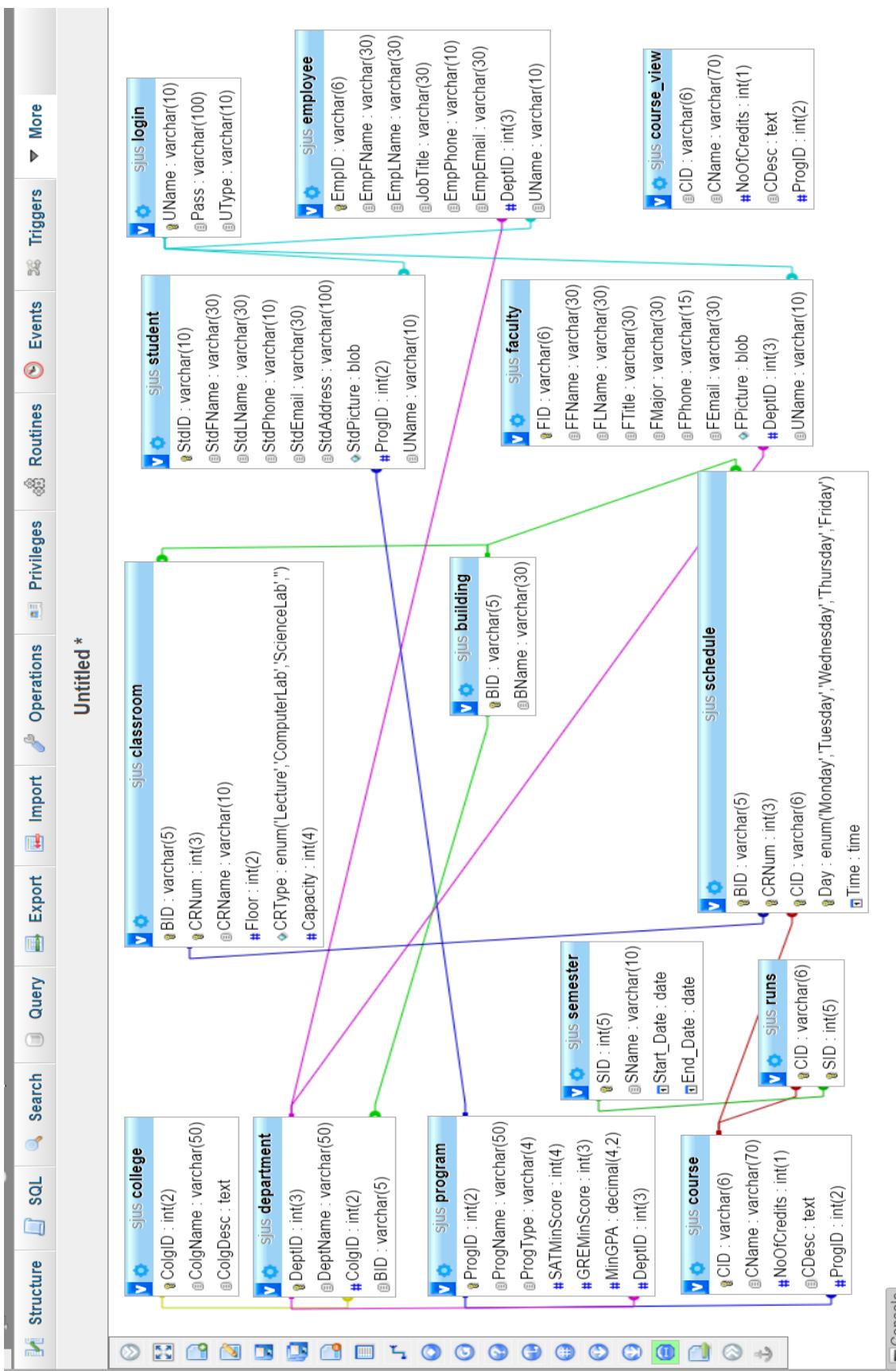
We implemented the following constraints to ensure all types of integrity.

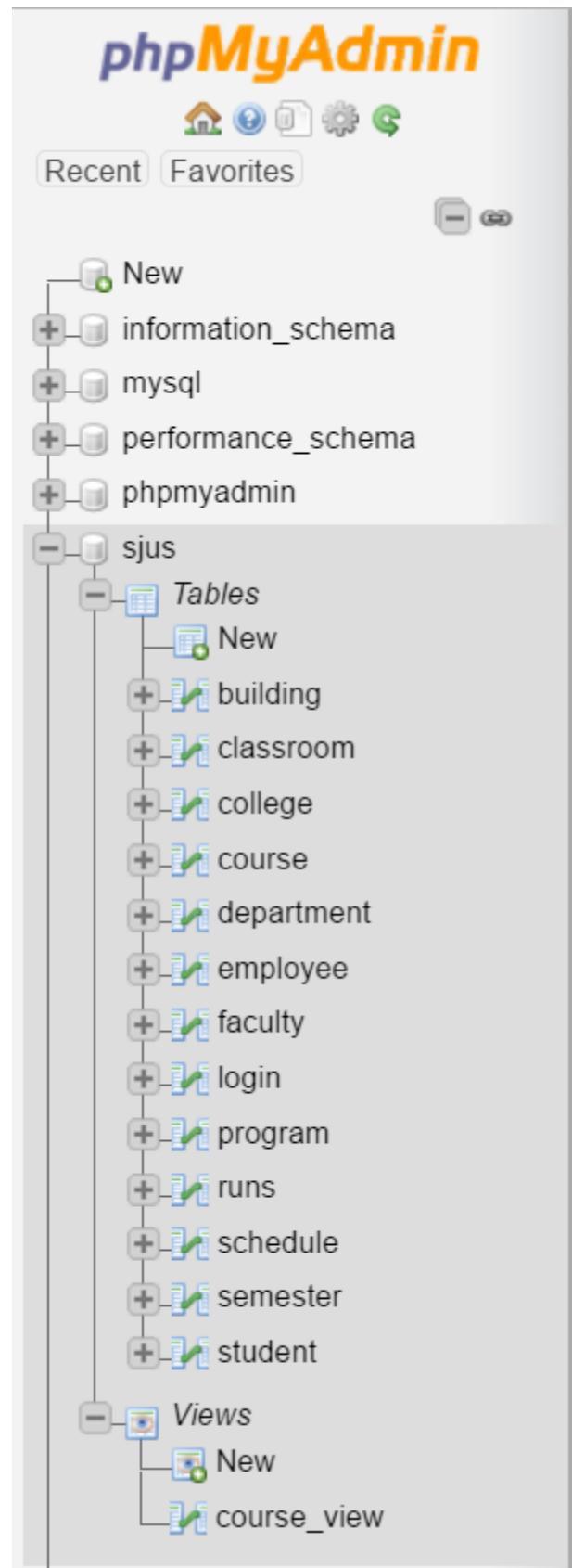
No.	Table	Constraints
1	Classroom	<ul style="list-style-type: none"> ADD CONSTRAINT `classroom_ibfk_1` FOREIGN KEY (`BID`) REFERENCES `building` (`BID`) ON DELETE CASCADE ON UPDATE CASCADE; ADD CONSTRAINT chk_capacity CHECK (Capacity>0);
2	Course	<ul style="list-style-type: none"> ADD CONSTRAINT `course_ibfk_1` FOREIGN KEY (`ProgID`) REFERENCES `program` (`ProgID`) ON DELETE CASCADE ON UPDATE CASCADE;
3	Department	<ul style="list-style-type: none"> ADD CONSTRAINT `department_ibfk_1` FOREIGN KEY (`ColgID`) REFERENCES `college` (`ColgID`) ON UPDATE CASCADE, ADD CONSTRAINT `department_ibfk_2` FOREIGN KEY (`BID`) REFERENCES `building` (`BID`) ON UPDATE CASCADE;
4	Employee	<ul style="list-style-type: none"> ADD CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`DeptID`) REFERENCES `department` (`DeptID`) ON UPDATE CASCADE, ADD CONSTRAINT `employee_ibfk_2` FOREIGN KEY (`UName`) REFERENCES `login` (`UName`) ON DELETE CASCADE ON UPDATE CASCADE;
5	Faculty	<ul style="list-style-type: none"> ADD CONSTRAINT `faculty_ibfk_1` FOREIGN KEY (`DeptID`) REFERENCES `department` (`DeptID`) ON UPDATE CASCADE, ADD CONSTRAINT `faculty_ibfk_2` FOREIGN KEY (`UName`) REFERENCES `login` (`UName`) ON DELETE CASCADE ON UPDATE CASCADE;
6	Student	<ul style="list-style-type: none"> ADD CONSTRAINT `student_ibfk_1` FOREIGN KEY (`ProgID`) REFERENCES `program` (`ProgID`) ON UPDATE CASCADE, ADD CONSTRAINT `student_ibfk_2` FOREIGN KEY (`UName`) REFERENCES `login` (`UName`) ON DELETE CASCADE ON UPDATE CASCADE;
7	Program	<ul style="list-style-type: none"> ADD CONSTRAINT `program_ibfk_1` FOREIGN KEY (`DeptID`) REFERENCES `department` (`DeptID`) ON UPDATE CASCADE;
8	Runs	<ul style="list-style-type: none"> ADD CONSTRAINT `runs_ibfk_1` FOREIGN KEY (`CID`) REFERENCES `course` (`CID`) ON UPDATE CASCADE, ADD CONSTRAINT `runs_ibfk_2` FOREIGN KEY (`SID`) REFERENCES `semester` (`SID`) ON UPDATE CASCADE;
9	Schedule	<ul style="list-style-type: none"> ADD CONSTRAINT `building` FOREIGN KEY (`BID`) REFERENCES `building` (`BID`) ON UPDATE CASCADE, ADD CONSTRAINT `classroom` FOREIGN KEY (`CRNum`) REFERENCES `classroom` (`CRNum`) ON UPDATE CASCADE, ADD CONSTRAINT `course` FOREIGN KEY (`CID`) REFERENCES `course` (`CID`) ON UPDATE CASCADE;
10	Semester	<ul style="list-style-type: none"> ADD CONSTRAINT chk_date CHECK (Start_Date<End_Date);

Insertion Queries

Here are some insertion queries:

No.	Table	Constraints
1	Classroom	<pre>INSERT INTO `classroom` ('BID', 'CRNum', 'CRName', 'Floor', 'CRTypE', 'Capacity') VALUES ('BN', 110, 'Lab 110', 1, 'ComputerLab', 30), ('BN', 111, 'Lab 111', 1, 'ScienceLab', 20), ('BN', 221, 'Class 221', 2, 'Lecture', 40), ('BN', 222, 'Class 222', 2, 'Lecture', 30), ('BN', 223, 'Class 223', 2, 'Lecture', 50), ('BN', 224, 'Lab 224', 2, 'ComputerLab', 25), ('BN', 225, 'Lab 225', 2, 'ComputerLab', 30), ('MV', 110, 'Class 110', 1, 'Lecture', 50), ('MV', 120, 'Class 120', 1, 'Lecture', 50), ('MV', 300, 'Class 300', 3, 'Lecture', 60), ('MV', 310, 'Class 310', 3, 'Lecture', 100), ('MV', 320, 'Lab 320', 3, 'ComputerLab', 25), ('MV', 330, 'Lab 330', 3, 'ComputerLab', 25), ('SC', 111, 'Lab 111', 1, 'ComputerLab', 20), ('SC', 112, 'Lab 112', 1, 'ScienceLab', 20), ('SC', 221, 'Class 221', 2, 'Lecture', 30), ('SC', 222, 'Class 222', 2, 'Lecture', 50), ('SC', 223, 'Lab 110', 2, 'ScienceLab', 20), ('SC', 224, 'Class 224', 2, 'Lecture', 40), ('SC', 225, 'Lab 225', 2, 'ScienceLab', 20);</pre>
2	Department	<pre>INSERT INTO `department` ('DeptID', 'DeptName', 'ColgID', 'BID') VALUES (11, 'Computer Science', 1, 'BN'), (23, 'Decision Support Science', 2, 'MV'), (24, 'Marketing', 2, 'MV'), (25, 'Accounting', 2, 'MV'), (26, 'Mathematics ', 1, 'SC'), (27, 'Physics', 1, 'SC'), (28, 'Finance', 2, 'MV'), (29, 'Food Marketing', 2, 'MV');</pre>
3	Program	<pre>INSERT INTO `program` ('ProgID', 'ProgName', 'ProgType', 'SATMinScore', 'GREMinScore', 'MinGPA', 'DeptID') VALUES (1, 'BS.Computer Science', 'BS', 1080, NULL, '3.27', 11), (2, 'MS.Computer Science', 'MS', NULL, NULL, '3.00', 11), (3, 'MS. in Business Intelligence and Analytics', 'MS', NULL, NULL, '3.00', 23), (4, 'BS. in Business Intelligence and Analytics', 'BS', 1100, NULL, '3.00', 23), (5, 'MS. in Marketing', 'MS', NULL, 160, '3.00', 24), (6, 'BS. in Marketing', 'BS', 1080, NULL, '3.00', 24);</pre>
4	Semester	<pre>INSERT INTO `semester` ('SID', 'SName', 'Start_Date', 'End_Date') VALUES (4, 'Spring16', '2016-01-01', '2016-05-01'), (5, 'Summer16', '2016-06-01', '2016-08-01'), (7, 'Fall16', '2016-09-01', '2016-12-01'), (8, 'Spring17', '2017-01-01', '2017-05-01');</pre>

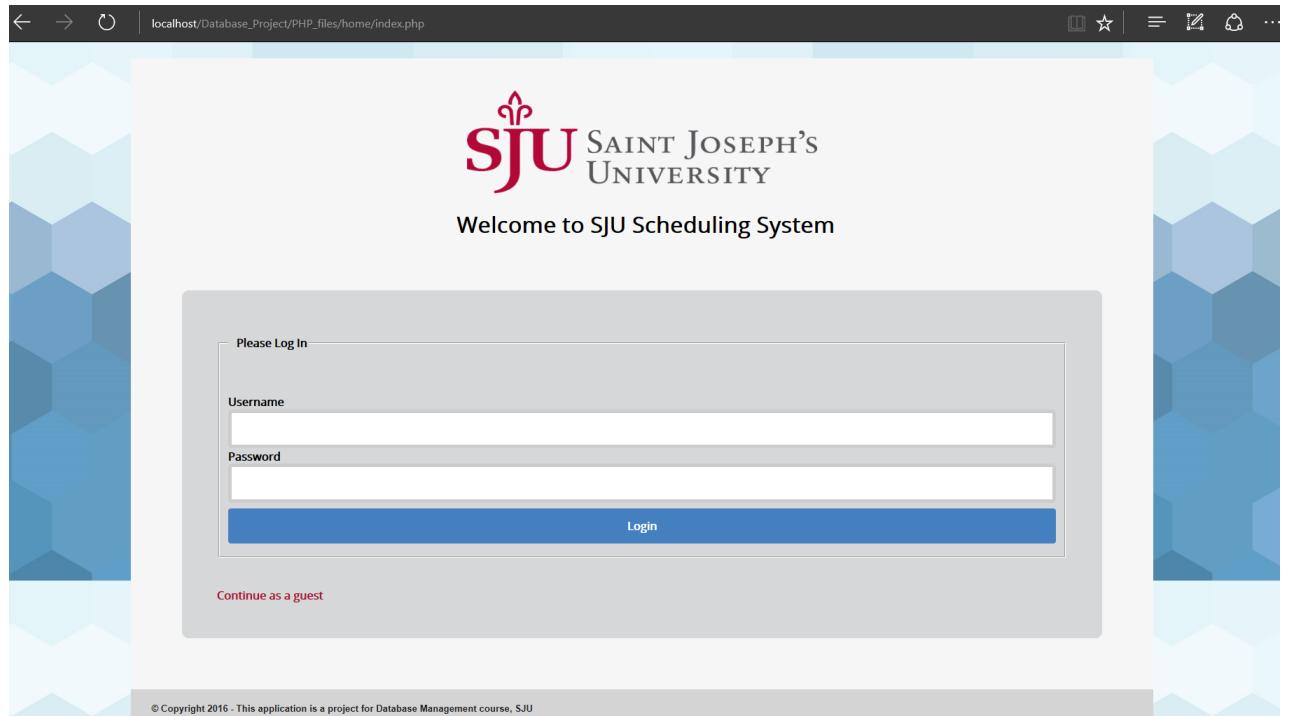




System Interfaces

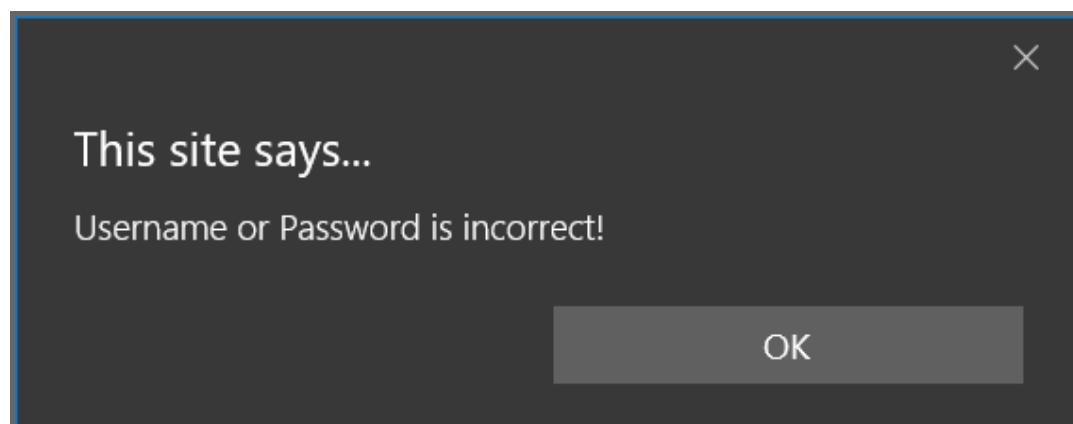
1. Homepage:

All users must login through this page by typing username and password. If the user is not an SJU member, they can browse some limited function as a Guest.



1.1. Homepage validations:

- If the user entered wrong username or password.



- If the user leaves the username or password field empty.

The screenshot shows a "Please Log In" form. The "Username" field is empty and highlighted with a red border. A red horizontal line also highlights the entire row of the "Username" field. A tooltip message "This is a required field" is displayed below the "Username" input field. Below the form is a blue "Login" button. At the bottom left of the page, there is a link "Continue as a guest".

2. Employee Interfaces

The screenshot shows the "Employee's Portal" main page. At the top, there is a navigation bar with links for "Home", "Add", "Search", and "Edit". On the right side of the navigation bar, it says "Welcome, Jumanah" and has "Edit Profile" and "Logout" buttons. The main content area features a large image of a university campus with a prominent Gothic-style tower. Above the image, the text "Employee's Portal" is centered. At the bottom of the page, there is a copyright notice: "© Copyright 2016 - This application is a project for Database Management course, SJU".

Employee's Portal is the Employee's main page and through this page Employees can interact with the system by adding, viewing, editing and deleting to and from the database. When the Employee logs in to the system a session will start and their name will be shown next to the welcome

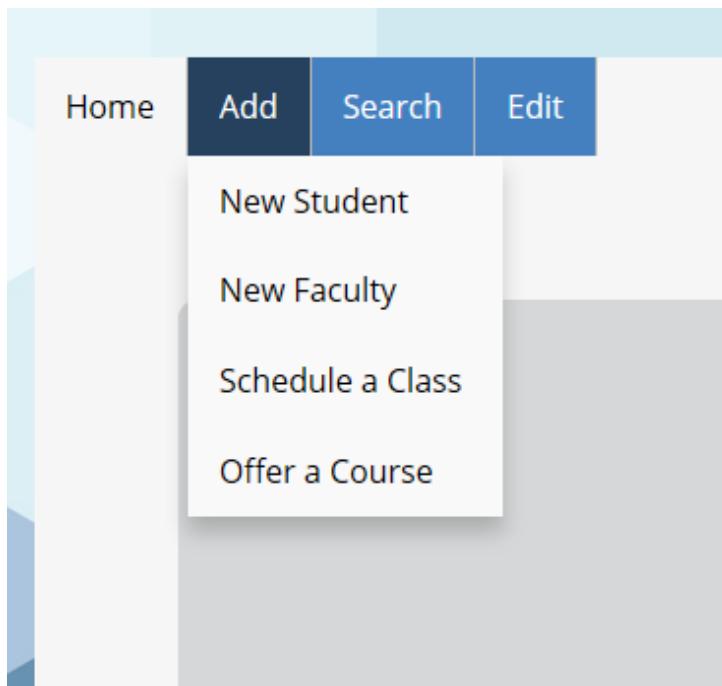
message. The session will remain on until the user is logged out. The following code shows how the session will start.

```
else if($count == 1 && $row['Pass']==$UPass && $row['UType']=="Employee")
{
    $EmpID = $Name;
    $ses_sql=mysqli_query($conn, "select * from employee where EmpID='$EmpID'");
    $row_sql=mysqli_fetch_array($ses_sql);
    $count_sql = mysqli_num_rows($ses_sql);
    if($count_sql == 1)
    {
        $_SESSION['EmpFName'] = $row_sql['EmpFName'];
    }

    $_SESSION['Employee'] = $EmpID;
    header("Location: ../employee/employee.php");
}
```

2.1. Add:

Employees can add new faculty, add new student, offer courses to a specific semester and schedule a course.



- Add new student:

The screenshot shows a web-based application interface for adding a new student. At the top, there is a navigation bar with links for Home, Add, Search, and Edit. On the right side of the top bar, it says "Welcome, Jumanah" and provides links for Edit Profile and Logout.

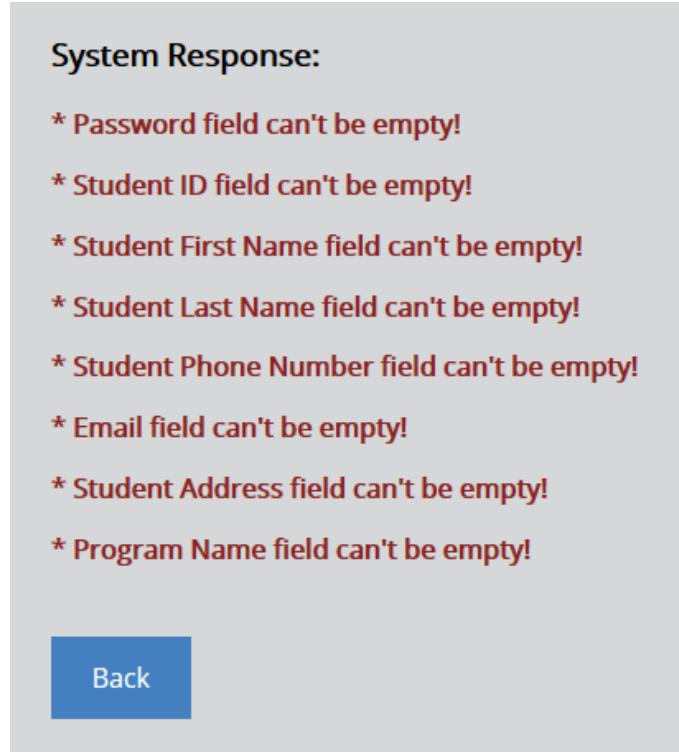
The main content area is titled "Add new Student". It contains several input fields:

- Student ID:** std000
- First Name:** (empty field)
- Last Name:** (empty field)
- Phone Number:** 0000000000
- Email:** example@example.com
- Address:** 5600 City Ave. Philadelphia, PA
- Picture:** (empty field)
- Email:** example@example.com
- Address:** 5600 City Ave. Philadelphia, PA
- Picture:** (empty field) with a "Browse..." button
- Program:** Choose one program (dropdown menu)
- Password:** (empty field) with a note: "Password should be between 6-15 characters"
- User Type:** Radio buttons for "Student" and "Faculty Member"

At the bottom of the form is a large blue "Submit" button.

In the footer, there is a copyright notice: "© Copyright 2016 - This application is a project for Database Management course, SJU".

All fields are required, except for the student picture, if the employee missed any of the fields the following message will appear.



The insertion query is written as the following:

```
$query = "INSERT INTO login (UName, Pass, UType)
VALUES ('$stdid', '$pass', '$utype');
INSERT INTO student(StdID, StdFName, StdLName, StdPhone, StdEmail, StdAddress, StdPicture, ProgID, UName)
VALUES ('$stdid', '$stdfn', '$stdln', '$pnum', '$email', '$stdadd', '$data', '$progid','$stdid');"

if(mysqli_multi_query($conn, $query))
{
    echo "* The record was added successfully.";
    $test = true;
}
else
{
    //echo "ERROR: Could not able to execute $insert1 \n $insert2 " . mysqli_error($conn);
    echo "<p class='errors'>* This username is already taken, please choose another one</p>";
    $test = false;
}
```

To make our database more secure we stored a hashed password in the database using the md5 algorithm.

	<input type="button" value="T"/>	UName	Pass	UType
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	emp001	e10adc3949ba59abbe56e057f20f883e	Employee
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	fm001	e10adc3949ba59abbe56e057f20f883e	Faculty
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	fm002	e10adc3949ba59abbe56e057f20f883e	Faculty
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	fm003	e10adc3949ba59abbe56e057f20f883e	Faculty
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	fm004	e10adc3949ba59abbe56e057f20f883e	Faculty

- Add new faculty:

The screenshot shows a web-based application interface. At the top, there is a navigation bar with buttons for 'Home', 'Add', 'Search', and 'Edit'. On the right side of the navigation bar, it says 'Welcome, Jumanah' and has links for 'Edit Profile' and 'Logout'. Below the navigation bar, the main content area has a title 'Add new Faculty Member'. The form itself is contained within a large grey box. It includes fields for 'Faculty ID' (with 'fm000' entered), 'First Name' (empty), 'Last Name' (empty), 'Title' (a dropdown menu showing 'Choose one title'), 'Major' (a dropdown menu showing 'Choose one Major'), and 'Phone Number' (with '0000000000' entered). The entire application has a light blue and white color scheme.

Home Add Search Edit

Welcome, Jumanah Edit Profile Logout

Add new Faculty Member

Faculty ID:
fm000

First Name:

Last Name:

Title:
Choose one title

Major:
Choose one Major

Phone Number:
0000000000

Email:
example@example.com

Picture:
Picture maximum size is 25KB

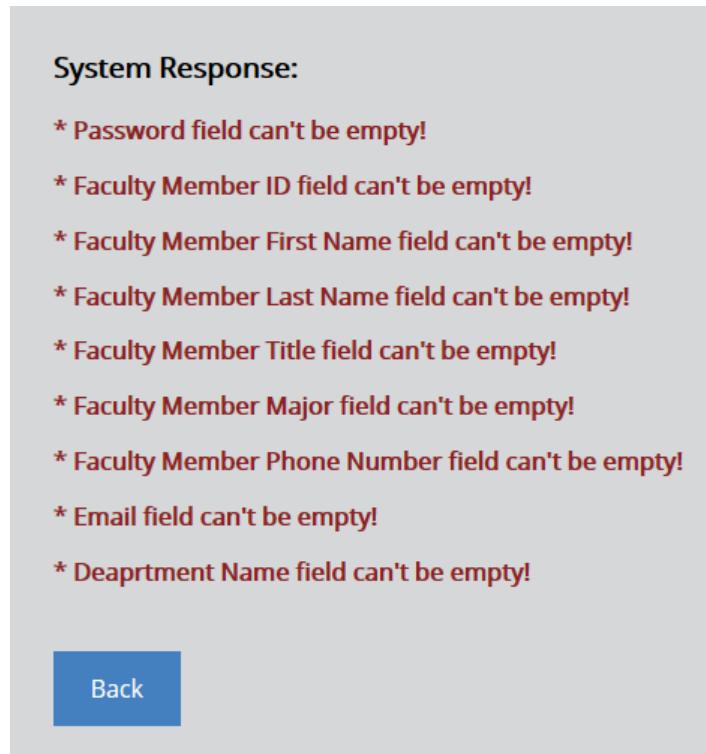
Department:
Choose one department

Password:
Password should be between 6-15 characters

User Type: Student Faculty Member

© Copyright 2016 - This application is a project for Database Management course, SJU

All fields are required, except for the faculty picture, if the employee missed any of the fields the following message will appear.



The insertion query is written as the following:

```
$query = "INSERT INTO login (UName, Pass, UType)
VALUES ('$fmid', '$pass', '$utype');
INSERT INTO faculty(FID, FFName, FLName, FTitle, FMajor, FPhone, FEmail, FPicture, DeptID, UName)
VALUES ('$fmid', '$fmfn', '$fmln', '$fmttitle', '$fmmajor', '$pnum', '$email', '$data', '$deptid','$fmid');";

if(mysqli_multi_query($conn, $query))
{
    echo "* The record was added successfully.";
    $test = true;
}
else
{
    //echo "ERROR: Could not able to execute $insert1 \n $insert2 " . mysqli_error($conn);
    echo "<p class='errors'>* This username is already taken, please choose another one</p>";
    $test = false;
}
```

- Offer a course:

The screenshot shows a web application interface for offering a course. At the top, there's a navigation bar with links for Home, Add, Search, Edit, Welcome, Jumanah, Edit Profile, and Logout. The main content area is titled "Offer a Course". It features three dropdown menus: "Semester" (with placeholder "choose a semester"), "Program" (with placeholder "Select a Program"), and "Course" (with placeholder "Select Program First"). Below these dropdowns is a large blue "Submit" button. At the bottom of the page, there's a small copyright notice: "© Copyright 2016 - This application is a project for Database Management course, SJU".

Through this page the employee can offer a course in a semester. The employee must choose the semester first then choose the program. After choosing the program the system will show a list of all courses that belongs to that program. This feature was implanted using AJAX and JQuery.

```

$(document).ready(function(){
    $('#program').on('change',function(){
        var ProgID = $(this).val();
        if(ProgID){
            $.ajax({
                type:'POST',
                url:'../employee/ajaxDataProgramCourse.php',
                data:'ProgID='+ ProgID,
                success:function(html){
                    $('#course').html(html);
                }
            });
        }else{
            $('#course').html('<option value="">Select Program First</option>');
        }
    });
});

```

```

<?php
//Include database configuration file
include_once('../connection.php');

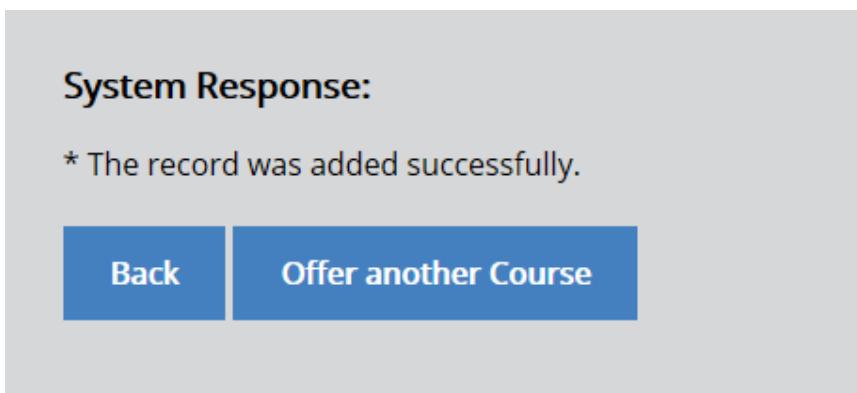
if(isset($_POST["ProgID"]) && !empty($_POST["ProgID"])){
    //Get all courses data
    $query = $conn->query("SELECT * FROM course WHERE ProgID='".$_POST['ProgID']."' ORDER BY CName ASC");

    //Count total number of rows
    $rowCount = $query->num_rows;

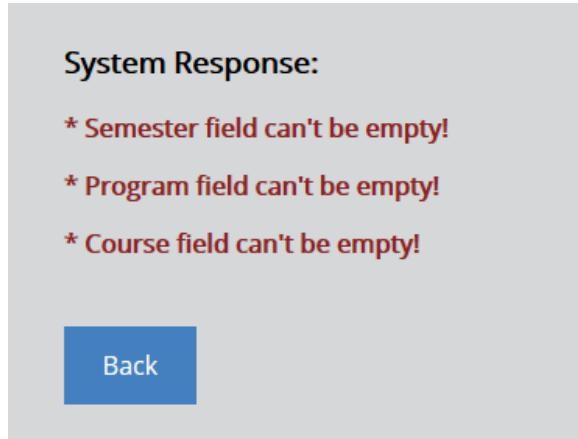
    //Display courses list
    if($rowCount > 0)
    {
        echo "<option value='>Select a Course</option>";
        while($row = $query->fetch_assoc())
        {
            echo "<option value='".$row['CID']."'>".$row['CName']."' </option>";
        }
    }else
    {
        echo '<option value="">Courses are not Available</option>';
    }
}
?>

```

If the course was added successfully the following message will appear:



Otherwise, an error message will appear:



- Schedule a course

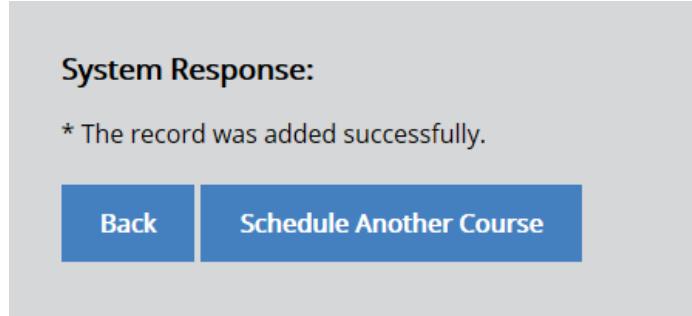
One of the main features in our system, through this page employees can schedule courses in a certain day, time and classroom.

The screenshot shows a "Schedule a Class" form page with a light gray background. The form fields are arranged vertically:

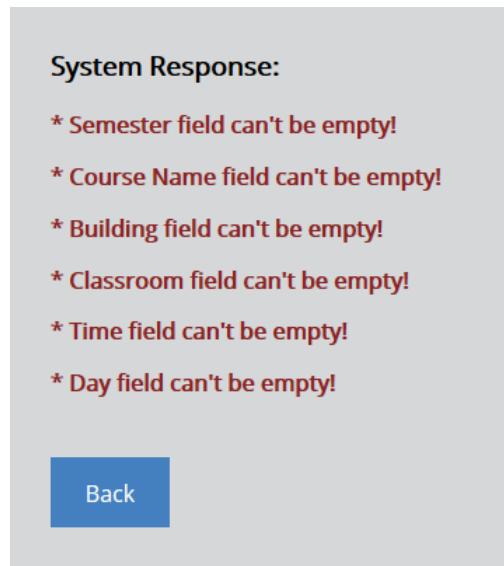
- Semester: A dropdown menu labeled "choose a semester".
- Course: A dropdown menu labeled "select a semester first".
- Building: A dropdown menu labeled "Select Building".
- Classroom: A dropdown menu labeled "select Building First".
- Time: A dropdown menu labeled "Choose Time".
- Day: A row of five radio buttons labeled "Monday", "Tuesday", "Wednesday", "Thursday", and "Friday".
- Submit: A large blue rectangular button at the bottom.

Also in this page, we used AJAX and JQuery to view all classes that in a certain building and to show offered courses in the selected semester.

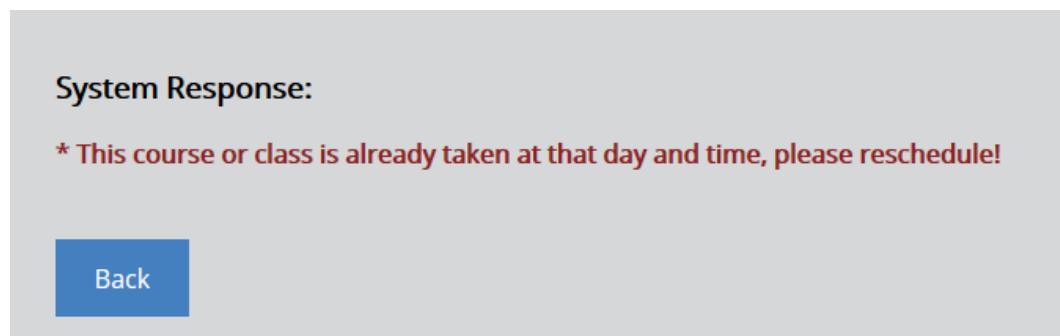
If the course was scheduled successfully the following message will appear:



Otherwise, an error message will appear.



Employee cannot schedule a course in two classes at the same time and day. Also, the employee cannot schedule two courses in the same classroom at the same time and day. If so, the following error message will appear:

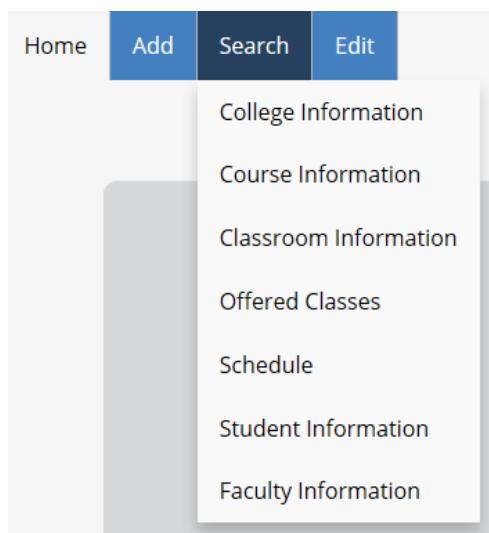


All dropdown lists are fetched from the database. For example, getting the programs and semester list.

```
<?php  
  
    $query = "SELECT ProgId,ProgName FROM program";  
    $result = mysqli_query($conn,$query);  
    $options ="";  
    while ($row = mysqli_fetch_array($result))  
    {  
        $options = $options."<option value = \"$row[0]\">$row[1]</option>";  
    }  
?  
  
<?php  
  
    //get all Semsters  
    $sqquery = "SELECT * FROM semester ORDER BY SID DESC";  
    $sresult = mysqli_query($conn,$sqquery);  
    $soptions ="";  
    while ($srow = mysqli_fetch_array($sresult))  
    {  
        $soptions = $soptions."<option value = \"$srow['SID']\">>".$srow['SName']."</option>";  
    }
```

2.2. Search:

Employees can search for college information, course information, classroom information, offered courses, schedule, student information and faculty information



- College Information: From this page Employee can get information about a specific college.

Please Select a College :

choose a college

Submit

College Name	College Description	Number of Departments
School of Business	The Erivan K. Haub School of Business is an international AACSB-accredited institution in business and accounting offering programs at the bachelor's, master's and executive master's levels. U.S. News & World Report has ranked the school's part-time MBA, risk-management and insurance, marketing, management, and finance programs among the top in the nation. The Haub School was recently named one of the nation's top business schools in the Princeton Review's "The Best 294 Business Schools," and the undergraduate program was recently recognized by Bloomberg's Businessweek.	5

Back

The query for this form is shown as the following:

```
<?php
if(isset($_POST['college']))
{
    $college = mysqli_real_escape_string($conn, $_REQUEST['college']);

    $query="SELECT c.*, COUNT(d.DeptID) AS Num_of_depts
           FROM college c, department d
           WHERE c.ColgID = d.ColgID AND
                 c.ColgID = '$college'";
    $result = mysqli_query($conn,$query);
}
?>
```

- Search for course description

The screenshot shows a web application interface. At the top, there is a navigation bar with buttons for "Home", "Add", "Search", and "Edit". On the right side of the navigation bar, it says "Welcome, Jumanah" and has links for "Edit Profile" and "Logout". Below the navigation bar, the main content area has a title "Search for Courses Description". Inside this area, there is a form with three dropdown menus. The first dropdown is labeled "Please Select a Department:" with the placeholder "choose a department". The second dropdown is labeled "Please Select a Program:" with the placeholder "Select College First". The third dropdown is labeled "Please Select a Course:" with the placeholder "Select Program First". Below these dropdowns is a blue "Submit" button. At the bottom left of the form area is a "Back" button.

The screenshot shows a table of course information. The table has four columns: "Course ID", "Course Name", "Credit Hours", and "Description". There is one row of data: CSC621, Database Systems, 3, and a detailed description. The description states: "This course covers the concepts and structures necessary to design and implement a database management system. Topics to be covered: data models (entity-relationship and relational), SQL, normalization, storage structures, enterprise applications and database integrity. Prerequisite: CSC 550 Object Oriented Design and Data Structures". At the bottom left of the table area is a "Back" button.

Course ID	Course Name	Credit Hours	Description
CSC621	Database Systems	3	This course covers the concepts and structures necessary to design and implement a database management system. Topics to be covered: data models (entity-relationship and relational), SQL, normalization, storage structures, enterprise applications and database integrity. Prerequisite: CSC 550 Object Oriented Design and Data Structures

Query used in this form:

```
$course = mysql_real_escape_string($_REQUEST['course']);

$query="SELECT *
        FROM course
        WHERE CID = '$course'";
$result = mysqli_query($conn,$query);
```

- Classroom Information: If an employee wants to schedule a course and he/she needs more information about the classrooms. Though this form he/she can get information about any classroom by selecting the building name.

The image consists of two vertically stacked screenshots of a web application. Both screenshots have a header bar at the top with buttons for Home, Add, Search, and Edit on the left, and Welcome, Jumanah, Edit Profile, and Logout on the right. The main content area has a title "Search for Classrooms Information".

Screenshot 1: This screenshot shows a single input field labeled "choose a building" with a dropdown arrow icon. Below it is a blue "Submit" button. At the bottom left is a "Back" button.

Screenshot 2: This screenshot shows the same interface but includes a table below the input field. The table has columns: Building, Classroom, Floor, Classroom Type, and Capacity. The data is as follows:

Building	Classroom	Floor	Classroom Type	Capacity
Barbelin	Lab 110	1	ComputerLab	30
Barbelin	Lab 111	1	ScienceLab	20
Barbelin	Class 221	2	Lecture	40
Barbelin	Class 222	2	Lecture	30
Barbelin	Class 223	2	Lecture	50
Barbelin	Lab 224	2	ComputerLab	25
Barbelin	Lab 225	2	ComputerLab	30

At the bottom left of this screenshot is a "Back" button.

Query used in this form:

```
if(isset($_POST['building']))
{
    $building = mysql_real_escape_string($_REQUEST['building']);

    $query="SELECT *
            FROM classroom c , building b
           WHERE c.BID = b.BID and
                 c.BID LIKE '$building'
           ORDER BY Floor, CRNum, Capacity ASC";
    $result = mysqli_query($conn,$query);
}
```

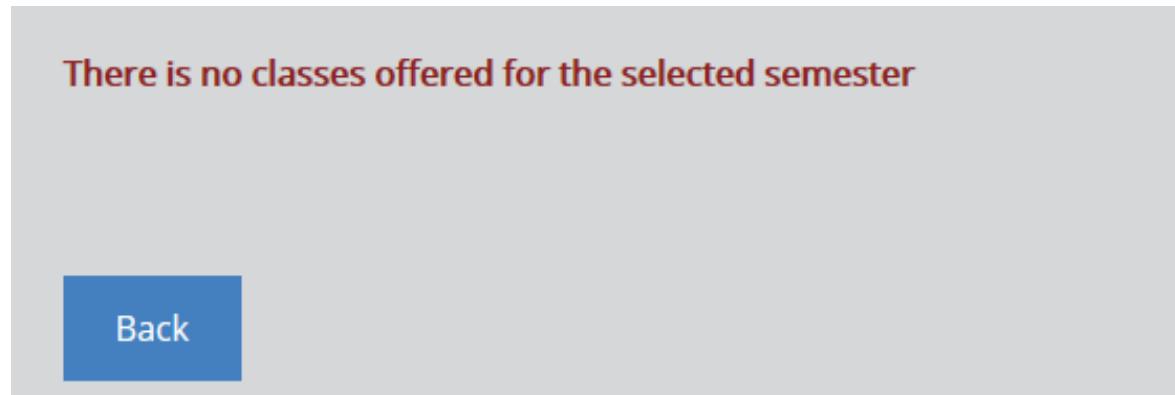
- Search for offered courses

The screenshot shows a web application interface for searching offered courses. At the top, there is a navigation bar with links for Home, Add, Search, and Edit. On the right side of the navigation bar, it says "Welcome, Jumanah" and provides links for Edit Profile and Logout. Below the navigation bar, the main content area has a title "Search for Offered Courses". A sub-section titled "Please Select a Semester:" contains a dropdown menu with the placeholder text "choose a semester". Below the dropdown is a blue "Submit" button. Underneath the "Submit" button is a "Back" button. At the bottom of the page, there is a copyright notice: "© Copyright 2016 - This application is a project for Database Management course, SJU". The lower half of the screenshot displays a table of course offerings:

Course ID	Course Name
CSC120	Computer Science I
CSC241	Discrete Structures II
CSC310	Computer Systems
CSC502	Introductory and Intermediate Computer Programming 2
CSC550	Object Oriented Design and Data Structures
CSC610	Software Engineering
CSC681	Programming Paradigms
DSS610	Business Analytics for BI
DSS650	Business Process Modeling & Analysis

At the bottom left of this table section is another "Back" button.

If no courses were offered in that semester the following message will appear:



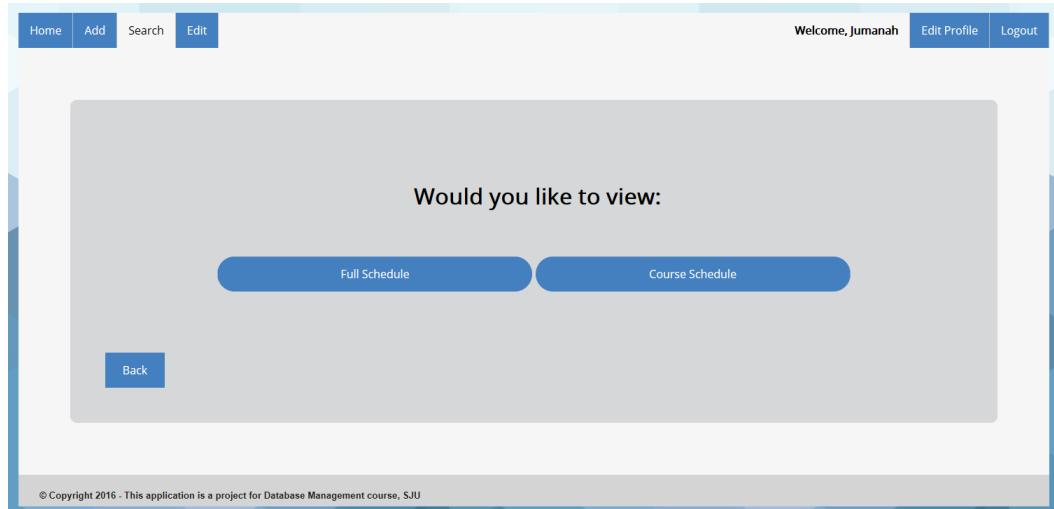
Query used in this form:

```
$query="SELECT *
        FROM runs r, semester s, course c
        WHERE r.CID = c.CID and
              r.SID = s.SID and
              s.SID = '$semester';
$result = mysqli_query($conn,$query);

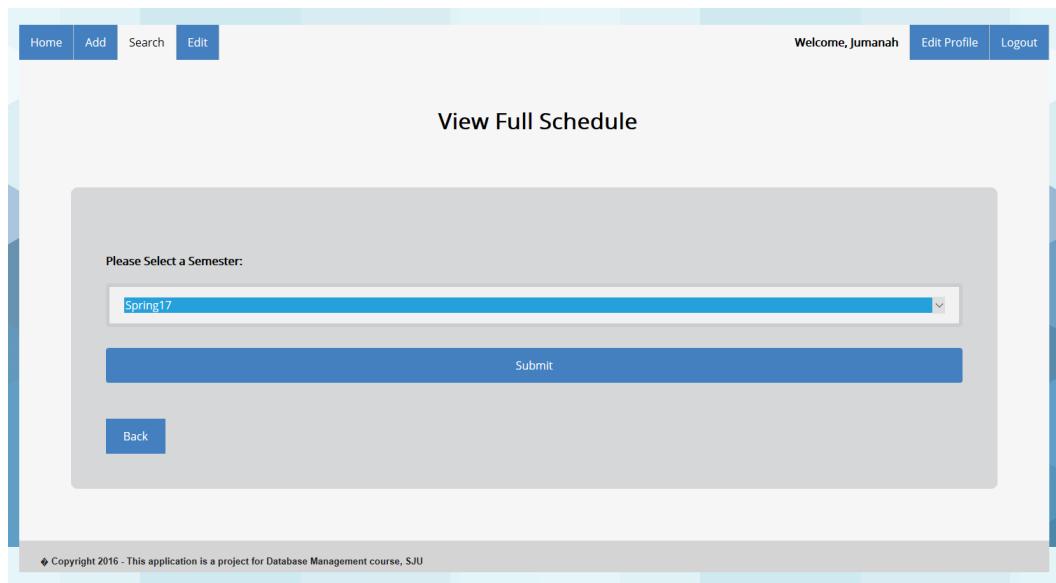
$rowCount = $result->num_rows;

if ($rowCount === 0)
{
    echo "<p class='errors'>There is no classes offered for
    echo "<br></br>";
}
```

- View schedule: Employees can view full schedule by choosing the semester or they can view a course schedule by choosing a course.



- View full schedule:



The screenshot shows a web-based application interface. At the top, there is a navigation bar with buttons for "Home", "Add", "Search", and "Edit". On the right side of the top bar, it says "Welcome, Jumanah" followed by "Edit Profile" and "Logout". Below the navigation bar, a button labeled "View Full Schedule" is visible. A large central box contains the following content:

Please Select a Semester:

choose a semester

Submit

Your Search Result is:

Day	Time	Course ID	Course Name	Building	Classroom
Monday	14:00:00	CSC502	Introductory and Intermediate Computer Programming 2	Barbelin	Class 223
Tuesday	09:00:00	CSC120	Computer Science I	Barbelin	Lab 110
Tuesday	13:00:00	CSC681	Programming Paradigms	Science Center	Lab 112
Wednesday	13:00:00	DSS610	Business Analytics for BI	Mandeville	Class 120
Wednesday	15:00:00	CSC241	Discrete Structures II	Mandeville	Lab 320
Thursday	12:00:00	CSC120	Computer Science I	Barbelin	Lab 110
Friday	16:00:00	CSC241	Discrete Structures II	Barbelin	Class 221

Back

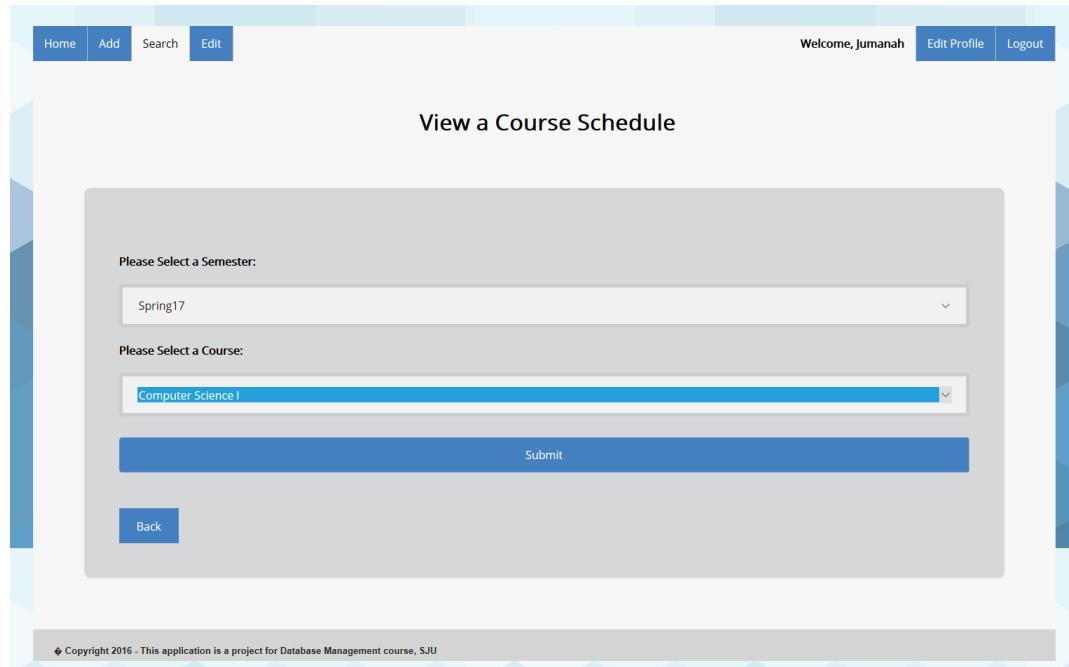
If there is no schedule classes in the chosen semester following message will be shown:

The screenshot shows a web-based application interface. A large red message box in the center contains the text "There is no classes scheduled for this semster". Below this message is a blue button labeled "Back".

Query used in this form:

```
<?php  
  
if(isset($_POST['semester']))  
{  
    $semester = mysql_real_escape_string($_REQUEST['semester']);  
  
    $query= " SELECT *  
             FROM schedule s, building b, course c, classroom cr  
            WHERE s.BID = b.BID AND  
                  s.CRNum = cr.CRNum AND  
                  s.CID = c.CID AND  
                  b.BID = cr.BID AND  
                  s.CID IN (   SELECT CID  
                           FROM runs  
                          WHERE SID = '$semester')  
            ORDER BY s.Day, s.Time ASC";  
  
    $result = mysqli_query($conn,$query);  
  
    $rowCount = $result->num_rows;  
  
    if ($rowCount === 0)  
    {  
        echo "<p class='errors'>There is no classes scheduled for this semester</p>";  
        echo "<br></br>";  
    }  
    else  
    {  
  
?>
```

o View course schedule:



The screenshot shows a web application interface for viewing course schedules. At the top, there is a navigation bar with links for Home, Add, Search, and Edit. On the right side of the header, there are links for Welcome, Jumanah, Edit Profile, and Logout. The main content area has a title "View a Course Schedule". Below the title, there are two dropdown menus: one for selecting a semester (with "Spring1" selected) and one for selecting a course (with "Computer Science I" selected). A large blue "Submit" button is positioned below the dropdowns. At the bottom left of the content area, there is a "Back" button. At the very bottom of the page, there is a footer bar with the copyright notice: "Copyright 2016 - This application is a project for Database Management course, SJU".

Home Add Search Edit Welcome, Jumanah Edit Profile Logout

View a Course Schedule

Please Select a Semester:

choose a semester

Please Select a Course:

select semester first

Submit

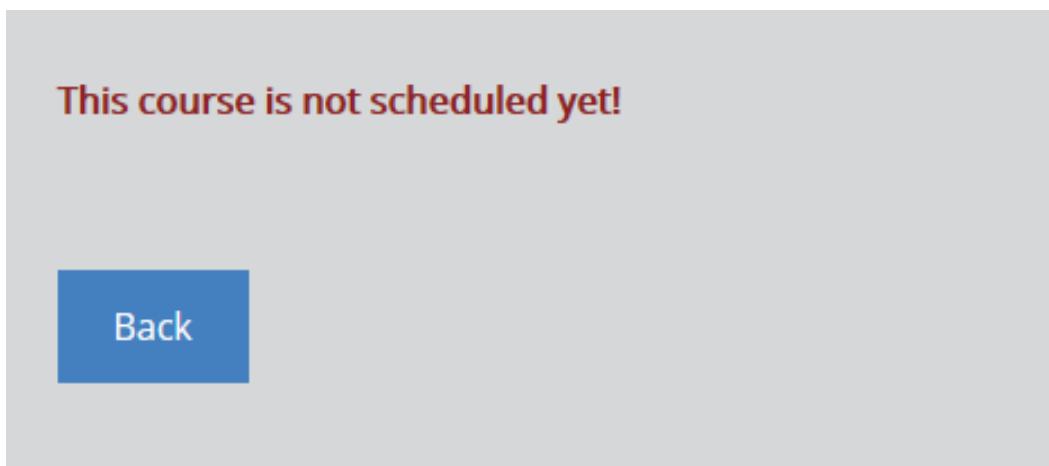
Your Search Result is:

Day	Time	Course ID	Course Name	Building	Classroom
Tuesday	09:00:00	CSC120	Computer Science I	Barbelin	Lab 110
Thursday	12:00:00	CSC120	Computer Science I	Barbelin	Lab 110

Back

This screenshot shows a user interface for viewing course schedules. At the top, there's a navigation bar with links for Home, Add, Search, and Edit, and a welcome message for the user 'Jumanah' along with links for Edit Profile and Logout. Below the navigation is a section titled 'View a Course Schedule'. It contains two dropdown menus: one for selecting a semester and another for selecting a course. Both dropdowns currently show placeholder text ('choose a semester' and 'select semester first'). A blue 'Submit' button is positioned below these fields. Underneath, a heading says 'Your Search Result is:' followed by a table. The table has columns for Day, Time, Course ID, Course Name, Building, and Classroom. It lists two entries: one for Tuesday at 09:00:00 and another for Thursday at 12:00:00, both for the course CSC120 titled 'Computer Science I' in the 'Barbelin' building, room 'Lab 110'. At the bottom left of the main content area is a blue 'Back' button.

If the course is not schedule yet the following message will be shown:



Query used in this form:

```
<?php

if(isset($_POST['course']))
{
    $course = mysql_real_escape_string($_REQUEST['course']);

    $query= " SELECT *
              FROM schedule s, building b, course c, classroom cr
              WHERE s.BID = b.BID AND
                    s.CRNum = cr.CRNum AND
                    s.CID = c.CID AND
                    b.BID = cr.BID AND
                    s.CID = '$course'
              ORDER BY s.Day, s.Time ASC";

    $result = mysqli_query($conn,$query);

    $rowCount = $result->num_rows;

    if ($rowCount === 0)
    {
        echo "<p class='errors'>This course is not scheduled yet!</p>";
        echo "<br><br>";
    }
    else
    {

?>
```

- Search for student information: Employee can view student information by searching for that student by their last name.

The screenshot shows a web application interface for searching student information. At the top, there is a navigation bar with links for Home, Add, Search, and Edit. On the right side of the navigation bar, it says "Welcome, Jumanah" and provides links for Edit Profile and Logout. Below the navigation bar, the main content area has a title "Search for Student". A form is present with the placeholder text "Please Enter Student Last Name :" and a text input field containing "enter the last name here". A blue "Submit" button is located below the input field. Further down, the text "Your Search Result is:" is displayed above a table. The table has columns for First Name, Last Name, Email, Program, and Picture. One row is shown in the table, corresponding to the search term "enter the last name here". The row contains the values: First Name - Fatemah, Last Name - Alkhalf, Email - falkhalaf@sju.edu, Program - MS.Computer Science, and Picture - a small placeholder icon. At the bottom left of the search result area, there is a "Back" button.

First Name	Last Name	Email	Program	Picture
Fatemah	Alkhalf	falkhalaf@sju.edu	MS.Computer Science	

Query used in this form:

```
$query="SELECT *
      FROM student s, program p
      WHERE s.ProgID = p.ProgID and
            StdLName like '$stdlname'";
$result = mysqli_query($conn,$query);
```

- Search for faculty information: Employee can view faculty information by searching for that faculty by their last name.

The image displays two screenshots of a web application interface. The top screenshot shows a search form titled "Search for Faculty Member" with a text input field for "Please Enter Faculty Last Name:" containing "enter the last name here". A blue "Submit" button is below the input field. The bottom screenshot shows the search results table with the following data:

First Name	Last Name	Title	Major	Email	Department	Picture
Mary	Krueger	Assistant Professor	Database	mkrueger@sju.edu	Computer Science	

A "Back" button is located at the bottom left of the results table. The bottom screenshot also has a header with "Welcome, Jumanah" and navigation links for "Edit Profile" and "Logout".

The second screenshot shows a similar search results table with one result:

First Name	Last Name	Title	Major	Email	Department	Picture
Alan	Nochenson	Assistant Professor	Computer Science	anochens@sju.edu	Computer Science	

A "Back" button is located at the bottom left of the results table.

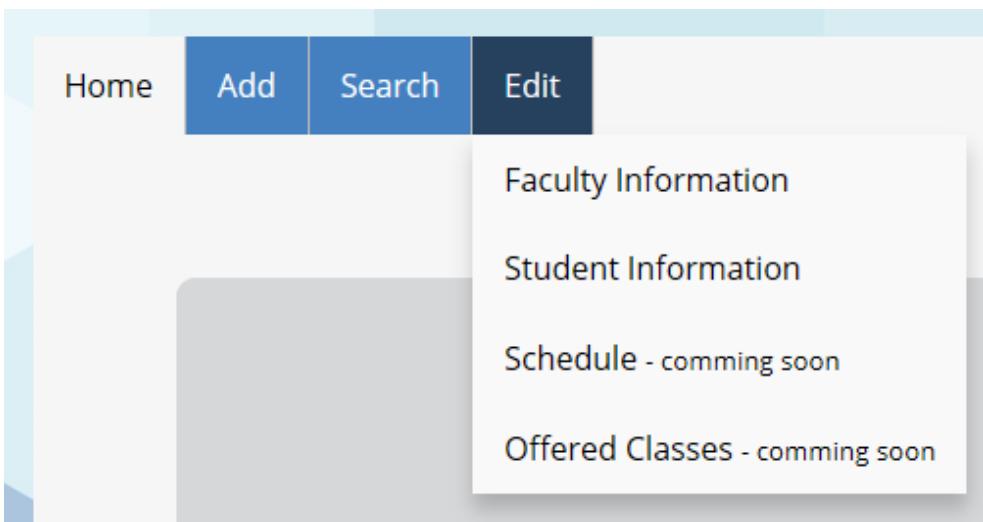
Your Search Result is:						
First Name	Last Name	Title	Major	Email	Department	Picture
Babak	Forouraghi	Professor	Computer Engineering	bforoura@sju.edu	Computer Science	
Back						

Query used in this form:

```
$query="SELECT *
      FROM faculty f, department d
      WHERE f.DeptID = d.DeptID and
            FLName like '$fname'";
$result = mysqli_query($conn,$query);
```

2.3. Edit:

Employee can edit and delete a faculty member, a student, and in the future he/she will be able to delete edit or delete offered course or course schedule.



- Edit/delete faculty: First the employee must search for that faculty and then choose the edit or delete action.

The screenshot shows a web application interface for searching faculty members. At the top, there is a navigation bar with buttons for Home, Add, Search, and Edit. On the right side of the header, it says "Welcome, Jumanah" and provides links for Edit Profile and Logout. The main content area has a title "Search for Faculty Member". Below the title, there is a form field labeled "Please Enter Faculty Last Name:" containing the value "alshehri". A blue "Submit" button is positioned below the input field. At the bottom left of the content area is a "Back" button. At the very bottom of the page, there is a copyright notice: "© Copyright 2016 - This application is a project for Database Management course, SJU".

This screenshot shows the same web application interface as the previous one, but with an empty search result. The search input field contains the placeholder text "enter the last name here". The rest of the interface, including the navigation bar, search title, and back button, is identical to the first screenshot.

If the employee chose the edit action, the form will retrieve this faculty information and allow the employee to edit the information. Note that the employee can edit

some of the information, but not everything, for example the employee cannot edit the faculty password.

Edit Faculty Member Information

First Name:
Jumanah

Last Name:
Alshehri

Major:
Computer Science

Phone Number:
6106602000

Email:
jshehri@sju.edu

Picture:
 [Browse...](#)
Picture maximum size is 25KB

Submit

If the update was completed successfully the following message will appear.

Otherwise, an error message will appear.

* The faculty member information was updated successfully.

First Name:
Jumanah

Last Name:
Alshehri

Major:
Computer Science

Phone Number:
2672529814

To delete this faculty, the employee must choose the delete action and then the faculty will be deleted from the database. And a message will appear in the delete action occurred or not.

Query used in this form:

```
<?php
if (isset($_REQUEST['deleteID']))
{
    $FID = $_REQUEST['deleteID'];
    $deleteQuery = "DELETE FROM login WHERE UName='$FID'";

    if (mysqli_query($conn, $deleteQuery) === TRUE)
    {
        echo "<p class='notice'>* Faculty member was deleted successfully.</p>";
    }
    else
    {
        echo "<p class='errors'>* System could not delete this faculty!</p>";
    }
}
?>
```

Since we used the constraint (Delete CASCADE) we only need to delete the faculty from the login table. And then it will be deleted automatically from the faculty table.

```
if($_FILES['FPicture']['tmp_name'] == null)
{
    $query = "UPDATE faculty SET FFName='$FFName', FLName='$FLName', FPhone='$FPhone', FEmail = '$FEmail' WHERE FID='$edit'";
    if(mysqli_query($conn, $query))
    {
        echo "<p class='notice'>* The faculty member information was updated successfully.</p>";
    }
    else
    {
        echo "<p class='errors'>* The system could not updated the faculty member information</p>";
    }
}
else
{
    $tmpName = $_FILES['FPicture']['tmp_name'];
    // Read the file
    $fp = fopen($tmpName, 'r');
    $data = fread($fp, filesize($tmpName));
    $data = addslashes($data);
    fclose($fp);
    $query = "UPDATE faculty SET FFName='$FFName', FLName='$FLName', FPhone='$FPhone', FEmail = '$FEmail', FPicture='$data' WHERE FID='$edit'";
    if(mysqli_query($conn, $query))
    {
        echo "<p class='notice'>* The faculty member information was updated successfully.</p>";
    }
    else
    {
        echo "<p class='errors'>* The system could not updated the faculty member information</p>";
    }
}
```

If the faculty was not found in the database, the following message will appear:

Search for Faculty Member

Please Enter Faculty Last Name :

Submit

There is no faculty member with this last name! please try again

Back

- Edit/delete student: the steps are same as editing and deleting a faculty member.

Home | Add | Search | **Edit** | Welcome, Jumanah | Edit Profile | Logout

Search for Student

Please Enter Student Last Name :

Submit

Back

© Copyright 2016 - This application is a project for Database Management course, SJU

Home Add Search Edit

Welcome, Jumanah Edit Profile Logout

Search for Student

Please Enter Student Last Name :

enter the last name here

Submit

Your Search Result is:

First Name	Last Name	Phone	Email	Address	Picture	Action
Fatemah	Alkhafaf	987654321	falkhalaf@sju.edu	5100cityave		Edit Delete

[Back](#)

Edit Student Information

* Student information was updated successfully.

* Student was deleted successfully.

[Back](#)

2.4. Edit profile:

Employees can also edit their own profile. However, they cannot edit all their profile because some fields such as job title has to be edited by the dean. When the employee clicks on edit profile, the form will automatically retrieve his/her information according to the session.

The screenshot shows a web application interface. At the top, there is a navigation bar with buttons for 'Home', 'Add', 'Search', and 'Edit'. On the right side of the top bar, it says 'Welcome, Jumanah' and has links for 'Edit Profile' and 'Logout'. Below the navigation bar, the main content area has a title 'Edit Profile'. A form is displayed with the following fields:

- First Name: Jumanah
- Last Name: Alshehri
- Phone Number: 2672529814
- Email: jshehri@sju.edu
- Password: (empty field)

A blue 'Submit' button is located at the bottom of the form.

After that, a notify message will appear if the information was updated successfully. Otherwise an error message will appear.

The screenshot shows the same 'Edit Profile' page after a successful update. A green success message at the top of the form area reads '* Your profile was updated successfully.' The 'First Name' field now contains 'Jumanah'. The other fields (Last Name, Phone Number, Email, Password) remain empty or unchanged from the previous screenshot.

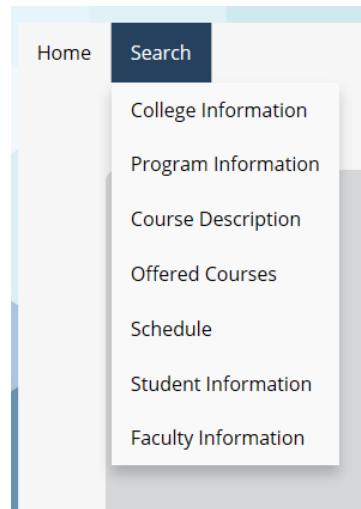
Query used in this form:

```
if(isset($_REQUEST['submit']))
{
    $EmpFName = mysqli_real_escape_string($conn, $_REQUEST['EmpFName']);
    $EmpLName = mysqli_real_escape_string($conn, $_REQUEST['EmpLName']);
    $EmpPhone = mysqli_real_escape_string($conn, $_REQUEST['EmpPhone']);
    $EmpEmail = mysqli_real_escape_string($conn, $_REQUEST['EmpEmail']);
    $fPass = $_REQUEST['fPass'];

    if($fPass == null)
    {
        $query = "UPDATE employee SET EmpFName='$EmpFName', EmpLName='$EmpLName', EmpPhone='$EmpPhone', EmpEmail='$EmpEmail' WHERE EmpID='$EmpID'";
        if(mysqli_query($conn, $query))
        {
            echo "<p class='notice'* Your profile was updated successfully.</p>";
        }
        else
        {
            echo "<p class='errors'* System could not updated your profile.</p>";
        }
    }
    else
    {
        $newPass = md5(mysqli_real_escape_string($conn, $_REQUEST['fPass']));
        $query = " UPDATE login l_employee e
                  SET l.Pass='$newPass',
                  e.EmpFName='$EmpFName',
                  e.EmpLName='$EmpLName',
                  e.EmpPhone='$EmpPhone',
                  e.EmpEmail='$EmpEmail'
                 WHERE e.EmpID = l.Uname AND
                       l.Uname = '$EmpID';";
        if(mysqli_query($conn, $query))
        {
            echo "<p class='notice'* Your profile was updated successfully.</p>";
        }
        else
        {
            echo "ERROR: Could not able to execute " . mysqli_error($conn);
            echo "<p class='errors'* System could not updated your profile.</p>";
        }
    }
}
```

3. Faculty and Student Interfaces

Faculty members and students do not have the permission to add, edit or delete anything in our system, except for editing their own profile. However, they can search for college information, program information, course description, offered classes, view full schedule, view course schedule, view student information and view faculty information.



Home Search

Welcome, Julie Edit Profile Logout

Faculty's Portal



© Copyright 2016 - This application is a project for Database Management course, SJU

Home Search

Welcome, Fatemah Edit Profile Logout

Student's Portal



© Copyright 2016 - This application is a project for Database Management course, SJU

- Search for college information

Please Select a College :

choose a college

Submit

College Name	College Description	Number of Departments
College of Art and Sciences	The College of Arts & Sciences at Saint Joseph's University offers programs in the humanities, natural sciences and social sciences for students pursuing bachelor's degrees, master's degrees, professional certificates and certifications, and interdisciplinary doctorates in Education Leadership. The College also provides the liberal arts foundation that is the heart of every student's education at Saint Joseph's, whatever your major or level.	3

Back

- Search for program requirements

Please Select a Department :

choose a department

Please Select a Program :

Select department First

Submit

Program Name	Program Type	GPA	SAT	GRE
MS. In Business Intelligence and Analytics	MS	3.00	-	-

Back

- Search for course description

Search for Courses Description

Please Select a Department:

Please Select a Program:

Please Select a Course:

Submit

Course ID	Course Name	Credit Hours	Description
CSC551	Design and Analysis of Algorithms	3	Concepts of program complexity; basic approaches to complexity reduction: data structures and techniques; worst cases and expected complexity. Topics to be covered may include sorting, set manipulation, graph algorithms, matrix multiplication, and finite Fourier transforms, polynomial arithmetic, and pattern matching. Prerequisite: CSC 550 Object Oriented Design and Data Structures

Back

- Search for offered courses

Home
Search
Welcome, Julie
Edit Profile
Logout

Search for Offered Courses

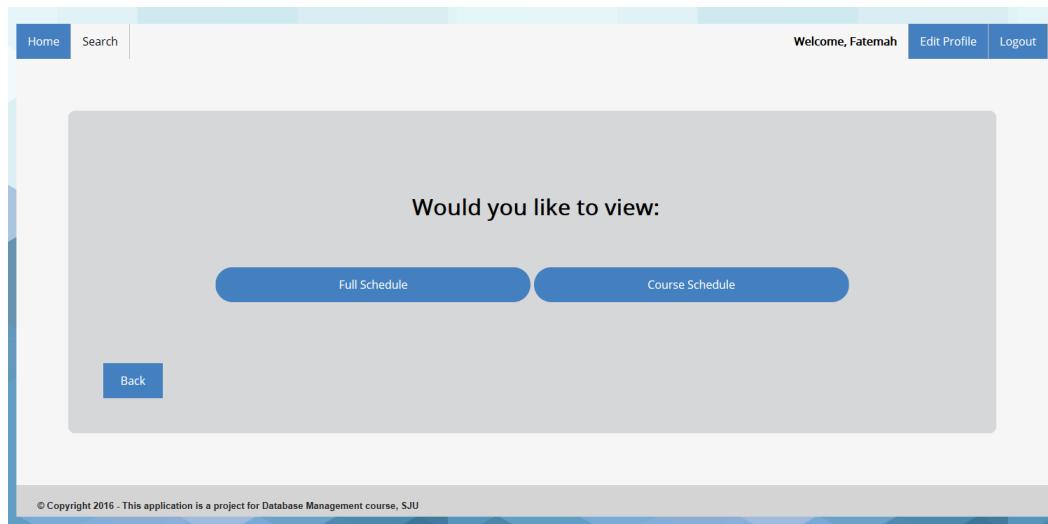
Please Select a Semester:

Submit

Course ID	Course Name
CSC120	Computer Science I
CSC241	Discrete Structures II
CSC310	Computer Systems
CSC502	Introductory and Intermediate Computer Programming 2
CSC550	Object Oriented Design and Data Structures
CSC610	Software Engineering
CSC681	Programming Paradigms
DS5610	Business Analytics for BI
DS5650	Business Process Modeling & Analysis

Back

- View schedule



The screenshot shows a top navigation bar with 'Home', 'Search', 'Welcome, Mary', 'Edit Profile', and 'Logout'. Below this is a section titled 'View Full Schedule'. Inside, there is a message 'Please Select a Semester:' above a dropdown menu with the placeholder 'choose a semester'. Below the dropdown is a blue 'Submit' button. Further down, the text 'Your Search Result is:' is followed by a table displaying course information. At the bottom left is a blue 'Back' button.

Day	Time	Course ID	Course Name	Building	Classroom
Monday	14:00:00	CSC502	Introductory and Intermediate Computer Programming 2	Barbelin	Class 223
Tuesday	09:00:00	CSC120	Computer Science I	Barbelin	Lab 110
Tuesday	13:00:00	CSC681	Programming Paradigms	Science Center	Lab 112
Wednesday	13:00:00	DSS610	Business Analytics for BI	Mandeville	Class 120
Wednesday	15:00:00	CSC241	Discrete Structures II	Mandeville	Lab 320
Thursday	12:00:00	CSC120	Computer Science I	Barbelin	Lab 110
Friday	16:00:00	CSC241	Discrete Structures II	Barbelin	Class 221

Home Search Welcome, Mary Edit Profile Logout

View a Course Schedule

Please Select a Semester:

choose a semester

Please Select a Course:

select semester first

Submit

Your Search Result is:

Day	Time	Course ID	Course Name	Building	Classroom
Wednesday	13:00:00	DSS610	Business Analytics for BI	Mandeville	Class 120

Back

- Search for student information

Home Search Welcome, Julie Edit Profile Logout

Search for Student

Please Enter Student Last Name :

enter the last name here

Submit

Your Search Result is:

First Name	Last Name	Email	Program	Picture
Fatemah	Alkhalf	falkhalaf@sj.edu	MS.Computer Science	

Back

- Search for faculty information

Home | Search | Welcome, Julie | Edit Profile | Logout

Search for Faculty Member

Please Enter Faculty Last Name :

Submit

Your Search Result is:

First Name	Last Name	Title	Major	Email	Department	Picture
George	Grevera	Associate Professor	Computer Vision	ggrevera@sju.edu	Computer Science	

Back

- Edit profile:

Home | Search | Welcome, Julie | Edit Profile | Logout

Edit Profile

* Your profile was updated successfully.

First Name:

Last Name:

Phone Number:

Email:

Password:

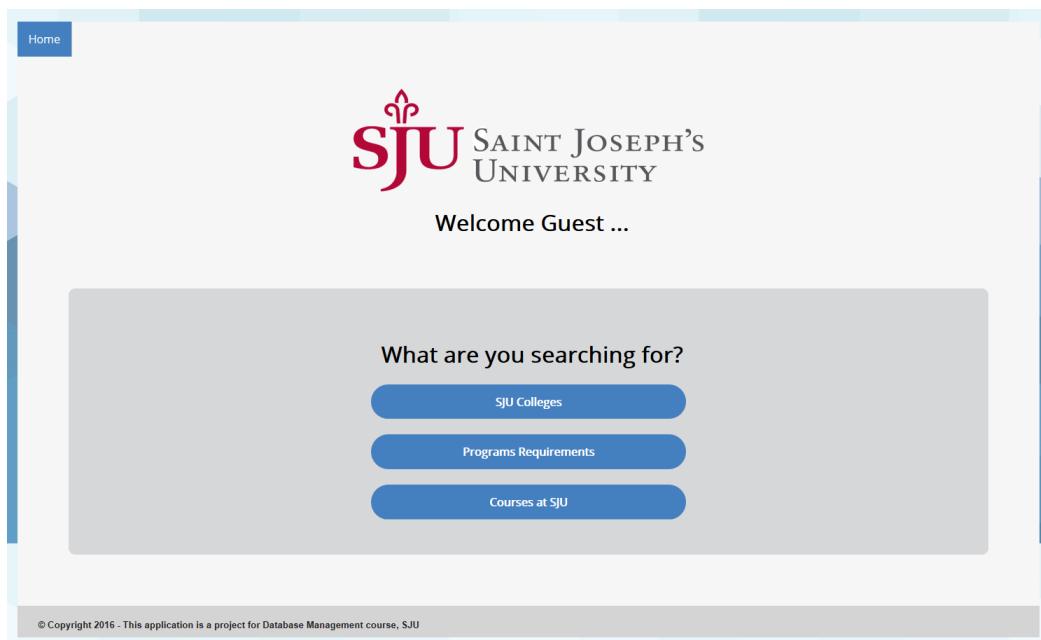
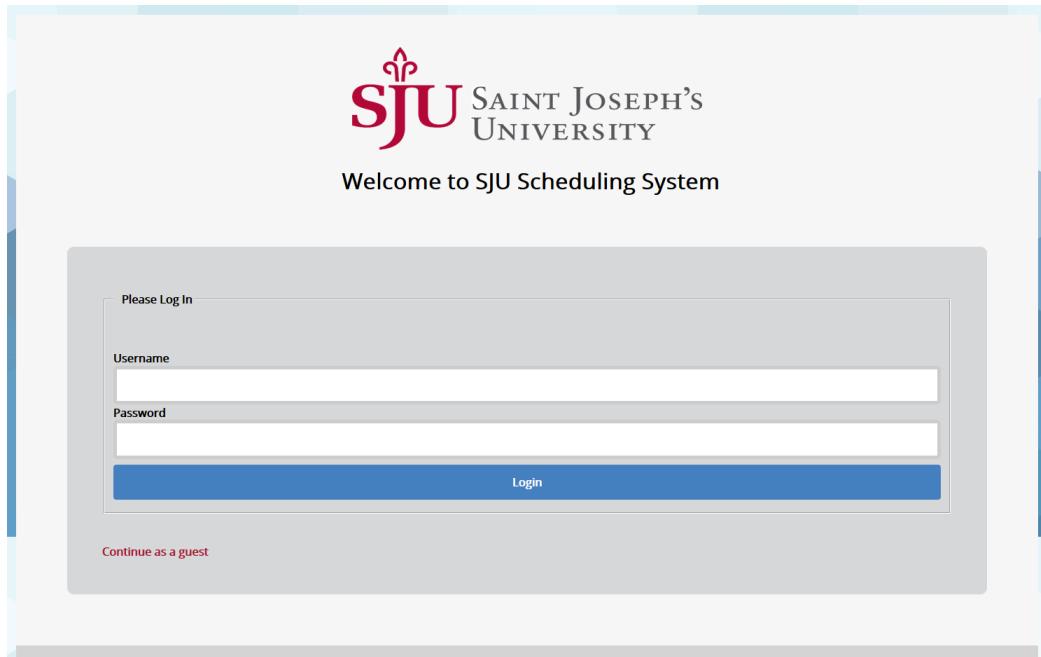
Picture:

Picture maximum size is 25KB

Submit

4. Guest Interfaces

Guests can browse our system by clicking on the “continue as a guest” link. Guests can only learn more about SJU colleges, view program requirements and view courses description a program.



Home

Search for College

Please Select a College :

choose a college

Submit

College Name	College Description	Number of Departments
School of Business	The Erivan K. Haub School of Business is an international AACSB-accredited institution in business and accounting offering programs at the bachelor's, master's and executive master's levels. U.S. News & World Report has ranked the school's part-time MBA, risk-management and insurance, marketing, management, and finance programs among the top in the nation. The Haub School was recently named one of the nation's top business schools in the Princeton Review's "The Best 294 Business Schools," and the undergraduate program was recently recognized by Bloomberg's Businessweek.	5

Back

Home

Search for Programs Requirements

Please Select a Department:

choose a department

Please Select a Program:

Select department First

Submit

Program Name	Program Type	GPA	SAT	GRE
BS. in Business Intelligence and Analytics	BS	3.00	1100	-

Back

Search for Courses Description

Please Select a Department:

choose a department

Please Select a Program:

Select Department First

Submit

Course ID	Course Name	Credit Hours	Description
CSC500	Discrete Structures	3	Mathematics needed for Computer Science. Topics covered include: functions, relations, propositional and first order predicate logic, set theory, proofs and their construction, counting and elementary probability. The course will use a declarative language as a tool to support concrete implementations of the mathematical ideas.
CSC501	Introductory and Intermediate Computer Programming 1	3	This is an intensive, one-semester two-course sequence intended to provide students with the necessary background in programming for the graduate program. The use of the computer to solve problems. Students will learn general principles of program design at first by using libraries of predefined program units, and later by constructing complete programs. Emphasis is on developing techniques for program design that lead to correct, readable and maintainable programs. Intermediate programming techniques including the use of recursion. An introduction to encapsulated data structures. Lists and list sorting will be used to introduce a discussion of algorithm efficiency.
CSC502	Introductory and Intermediate Computer Programming 2	3	This is an intensive, one-semester two-course sequence intended to provide students with the necessary background in programming for the graduate program. The use of the computer to solve problems. Students will learn general principles of program design at first by using libraries of predefined program units, and later by constructing complete programs. Emphasis is on developing techniques for program design that lead to correct, readable and maintainable programs. Intermediate programming techniques including the use of recursion. An introduction to encapsulated data structures. Lists and list sorting will be used to introduce a discussion of algorithm efficiency.

Since the course description will not be updated that often, we created a view for this form to retrieve the data faster. Query for this form:

```
$query="SELECT *
      FROM course_view
      WHERE progID = '$program'";
$result = mysqli_query($conn, $query);
```

References

- Dev, D. (n.d.). *Microsoft Access Query Types*. Retrieved from Database Dev:
<http://www.databasedev.co.uk/microsoft-access-query-types.html>
- Jukic , N., Vrbsky , S., & Nestorov, S. (2016). *Database Systems: Introduction to Databases and Data Warehouses*. Danvers: Prospect Press.
- tutorial, W. 2. (n.d.). *Oracle SQL Introduction*. Retrieved from Way 2 tutorial:
http://www.way2tutorial.com/sql/oracle_sql_introduction_type_of_sql_statement.php
- Wales, M. (2014, December 8). *3 Web Dev Careers Decoded: Front-End vs Back-End vs Full Stack*. Retrieved from Udacity: <http://blog.udacity.com/2014/12/front-end-vs-back-end-vs-full-stack-web-developers.html>
- W3Schools. (n.d.). *THE WORLD'S LARGEST WEB DEVELOPER SITE* <http://www.w3schools.com/>