CSC350 Group D
Professor Kok
Spring 2020

Keyshawn Harinarain
Shirley Ni
Juan Peguero

PDF version of all deliverables.

Tasks

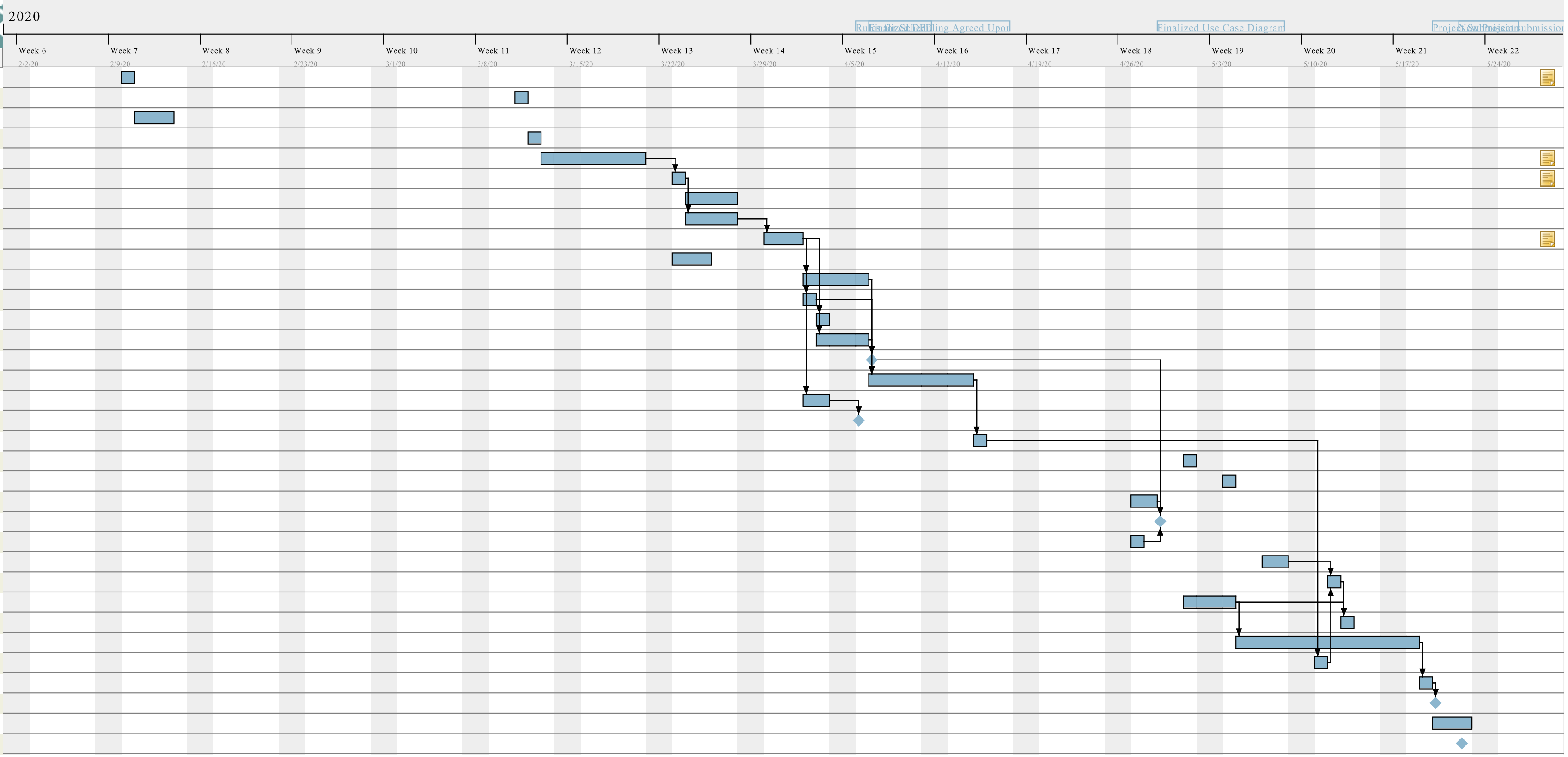
Name	Begin date	End date
Setup Group comSetup Group communication munication *Text group chat*	2/10/20	2/10/20
GitHub Repository	3/11/20	3/11/20
Interview Questions 1	2/11/20	2/13/20
Upload ScreenShot and README	3/12/20	3/12/20
Brainstorm Solutions *General brainstorming, getting ready for first meeting.*	3/13/20	3/20/20
First Meeting on Discord *Discussion, gathering an approach. Putting out ideas regarding the project with what we know at that time.*	3/23/20	3/23/20
Excel Sheet of classes uploaded	3/24/20	3/27/20
Idea of HTML form structure	3/24/20	3/27/20
Hand Scheduling 1 *Doing the scheduling by hand to get a feel for what the algorithim would look like.*	3/30/20	4/1/20
Grid Project	3/23/20	3/25/20
work on DFD	4/2/20	4/6/20
Scheduling Rules	4/2/20	4/2/20
Revise Scheduling Rules	4/3/20	4/3/20
Scheduling Logic Diagram	4/3/20	4/6/20
Finalized DFD	4/7/20	4/7/20
Schema design-Normalization	4/7/20	4/14/20
Hand Scheduling 2	4/2/20	4/3/20
Rules for Scheduling Agreed Upon	4/6/20	4/6/20
Schema Implementation	4/15/20	4/15/20
UI Development Sketches	5/1/20	5/1/20
Interview Questions 2	5/4/20	5/4/20
Use Case Diagram Sketches	4/27/20	4/28/20
Finalized Use Case Diagram	4/29/20	4/29/20
UI Review with professor	4/27/20	4/27/20
updated normalization	5/7/20	5/8/20
Database Dictionary	5/12/20	5/12/20

Tasks

Name	Begin date	End date
Finished Test Data	5/1/20	5/4/20
Connecting Database to PHP	5/13/20	5/13/20
Debugging Code	5/5/20	5/18/20
ERD	5/11/20	5/11/20
Finalized Deliverables	5/19/20	5/19/20
Project Submission	5/20/20	5/20/20
Project Extension	5/20/20	5/22/20
New Project submission	5/22/20	5/22/20

Gantt Chart

Name	Begin date	End date
Setup Group comSetu...	2/10/20	2/10/20
GitHub Repository	3/11/20	3/11/20
Interview Questions 1	2/11/20	2/13/20
Upload ScreenShot an...	3/12/20	3/12/20
Brainstorm Solutions	3/13/20	3/20/20
First Meeting on Disco...	3/23/20	3/23/20
Excel Sheet of classes ...	3/24/20	3/27/20
Idea of HTML form str...	3/24/20	3/27/20
Hand Scheduling 1	3/30/20	4/1/20
Grid Project	3/23/20	3/25/20
work on DFD	4/2/20	4/6/20
Scheduling Rules	4/2/20	4/2/20
Revise Scheduling Rules	4/3/20	4/3/20
Scheduling Logic Diag...	4/3/20	4/6/20
Finalized DFD	4/7/20	4/7/20
Schema design-Norm...	4/7/20	4/14/20
Hand Scheduling 2	4/2/20	4/3/20
Rules for Scheduling A...	4/6/20	4/6/20
Schema Implementati...	4/15/20	4/15/20
UI Development Sketc...	5/1/20	5/1/20
Interview Questions 2	5/4/20	5/4/20
Use Case Diagram Sket...	4/27/20	4/28/20
Finalized Use Case Dia...	4/29/20	4/29/20
UI Review with profess...	4/27/20	4/27/20
updated normalization	5/7/20	5/8/20
Database Dictionary	5/12/20	5/12/20
Finished Test Data	5/1/20	5/4/20
Connecting Database t...	5/13/20	5/13/20
Debugging Code	5/5/20	5/18/20
ERD	5/11/20	5/11/20
Finalized Deliverables	5/19/20	5/19/20
Project Submission	5/20/20	5/20/20
Project Extension	5/20/20	5/22/20
New Project submission	5/22/20	5/22/20



Keyshawn Harinarain

Shirley Ni

Juan Peguero

Group D Project Description

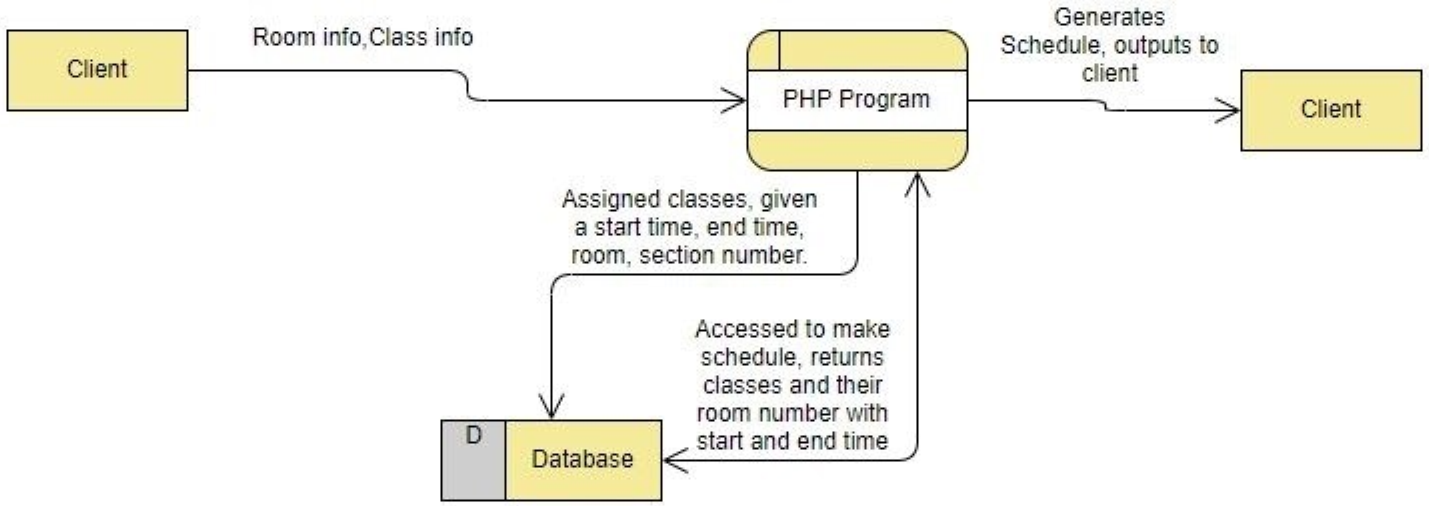
Professor Kok

Project Purpose:

To create a program that takes a user input of room numbers, class numbers, and days a class is to be scheduled, and output a class schedule for the rooms given. This project is also, to determine our understanding of the software development process and the different phases of development. This begins with planning and organizing our approach as well as consistently updating a GANTT chart to plan, manage, and control the time for different phases of the process. This is a base level of our project and provides the ability to set time expectations and deadlines, as well as account for various setbacks that can occur along the process.

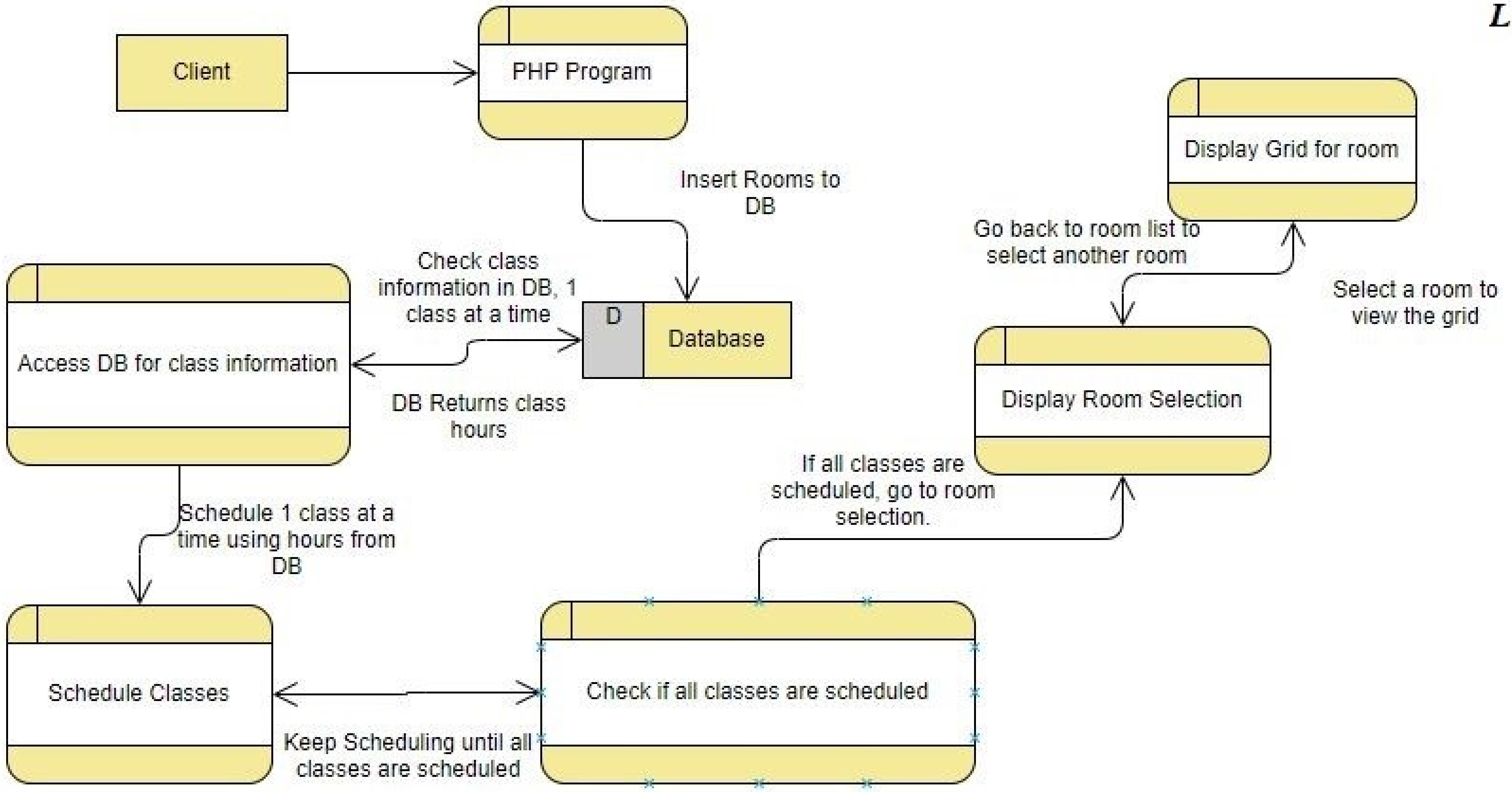
This project then goes into the development process itself, which involves designing a program at the very base level, this starts with how data flows in our program. To accomplish this, we use three levels of Data Flow Diagrams. The base level, or context, shows a high-level view of the data flow process, and how a user data moves through our program. The next level, level zero, shows a more detailed view of the process and allows a use to follow the data as it goes from the start of its flow, to the end output, and the various processes that occur in between. The level one diagram is an in-depth view of our scheduling process, how we manage data, what happens if an input meets a certain criteria, how does this class number, days to be scheduled, and room number actually turn into a scheduled class.

This isn't our only process, software development occurs in different stages, and diagrams like entity-relationship diagrams and use-case diagrams show how users interact with our software. How does a client use our program? What happens when a client selects a certain button or uploads their information? How does this entity (user or client) engage with our software? These diagrams showcase this interaction and it is crucial to the design process to understand how a user can interact with our software. This project is a look into the software development process, from how users interact with our program, to the planning and time management aspects, testing, users' interactions, and the actual user interface, this project is an entire view of how software goes from an idea from a brainstorming session, to a final product.

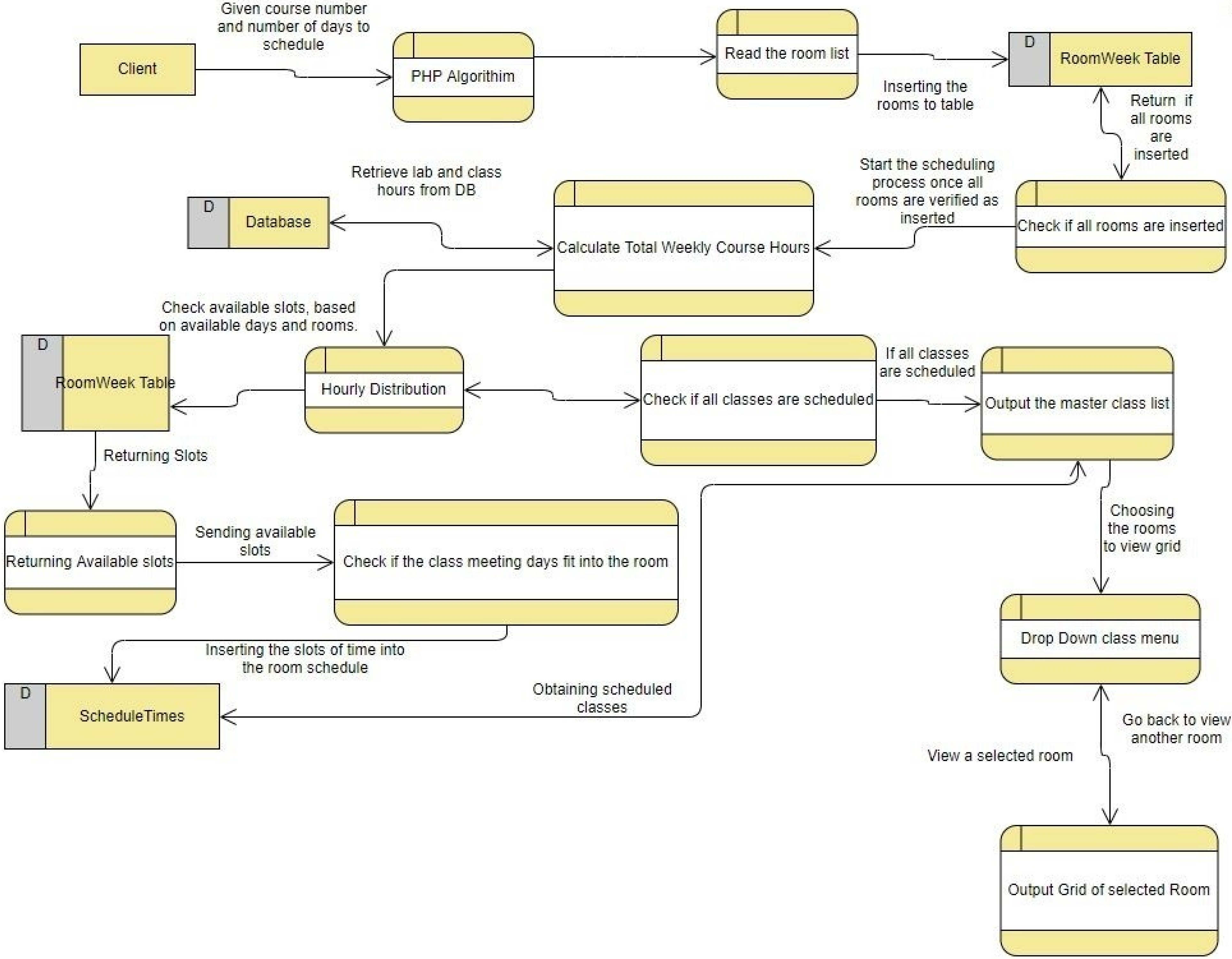

Client
Room info,Class info
PHP Program
Generates Schedule, outputs to client
Client
Assigned classes, given a start time, end time, room, section number.
Accessed to make schedule, returns classes and their room number with start and end time
D
Database

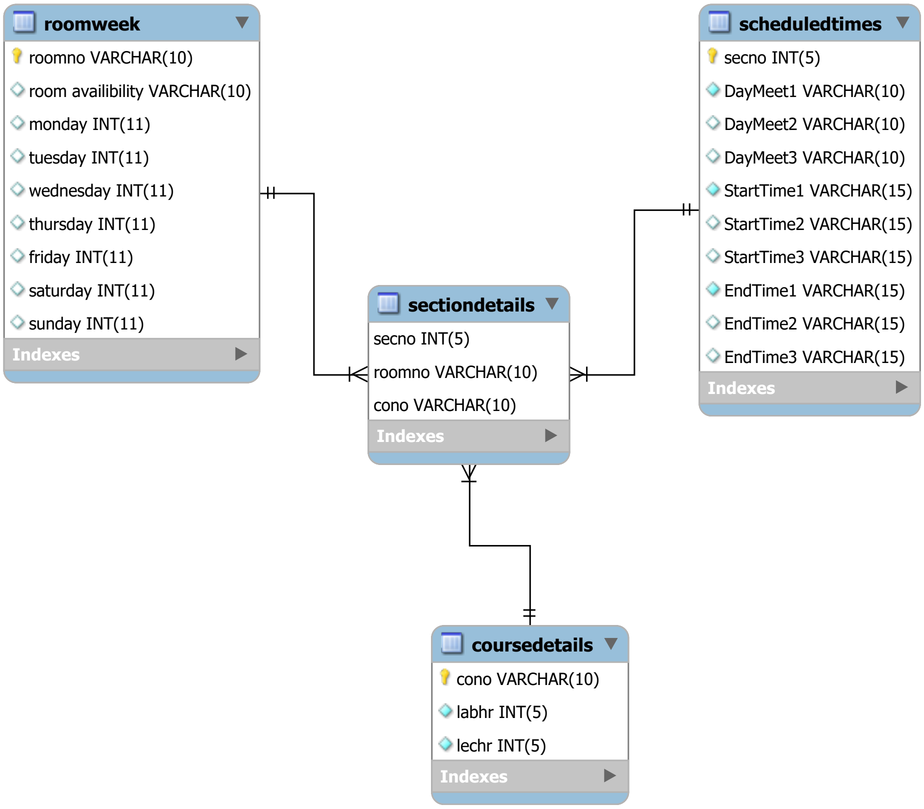
Context

Level 0


Client
PHP Program
Insert Rooms to DB
D
Database
Check class information in DB, 1 class at a time
Access DB for class information
DB Returns class hours
Schedule 1 class at a time using hours from DB
Schedule Classes
Keep Scheduling until all classes are scheduled
Check if all classes are scheduled
If all classes are scheduled, go to room selection.
Display Room Selection
Go back to room list to select another room
Display Grid for room
Select a room to view the grid

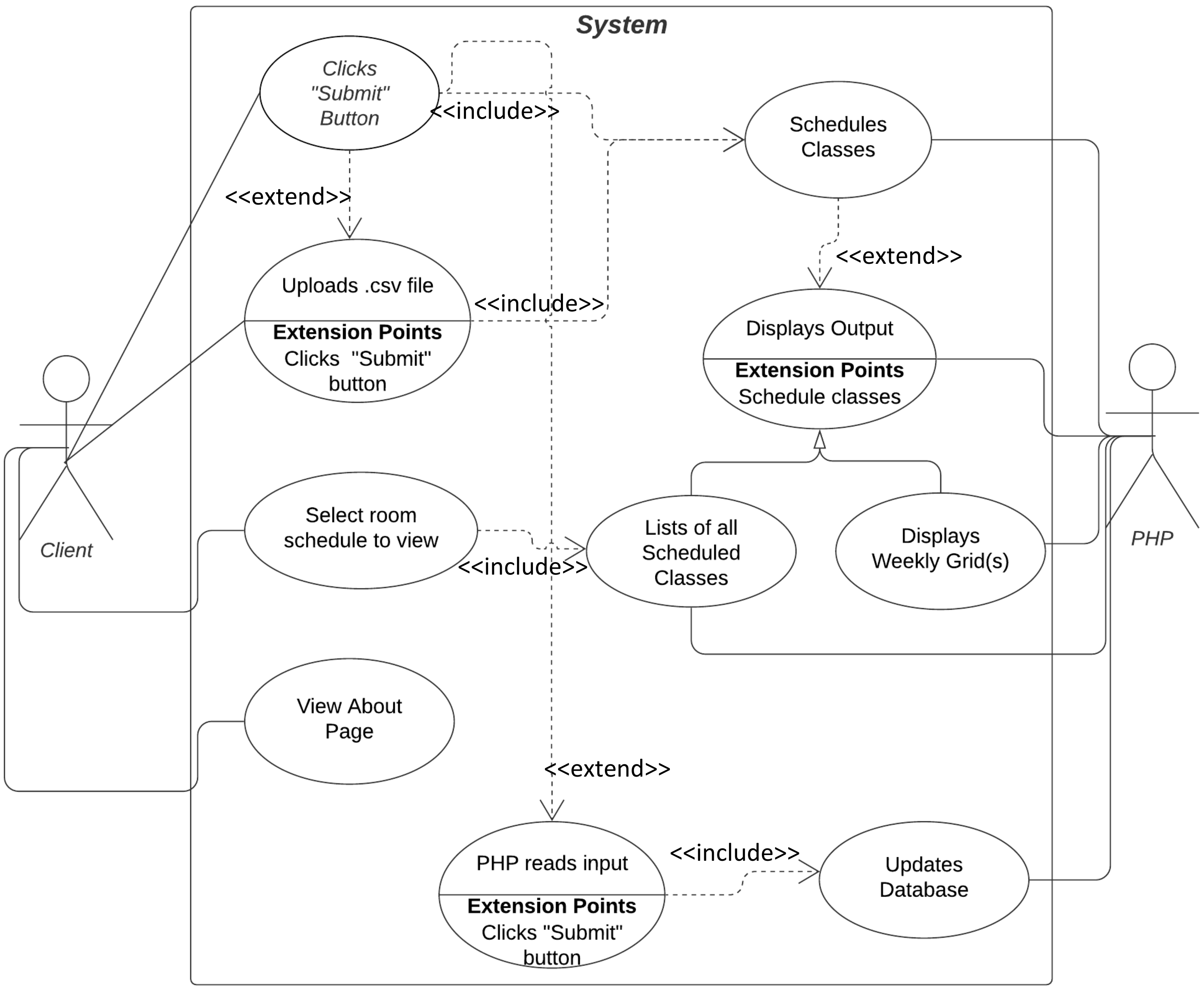
Level 1


Client
Given course number and number of days to schedule
PHP Algorithim
Read the room list
Inserting the rooms to table
D
RoomWeek Table
Return if all rooms are inserted
Check if all rooms are inserted
Start the scheduling process once all rooms are verified as inserted
Calculate Total Weekly Course Hours
Retrieve lab and class hours from DB
D
Database
Check available slots, based on available days and rooms.
Hourly Distribution
D
RoomWeek Table
Check if all classes are scheduled
If all classes are scheduled
Output the master class list
Returning Slots
Returning Available slots
Sending available slots
Check if the class meeting days fit into the room
Inserting the slots of time into the room schedule
D
ScheduleTimes
Obtaining scheduled classes
Choosing the rooms to view grid
Drop Down class menu
Go back to view another room
View a selected room
Output Grid of selected Room

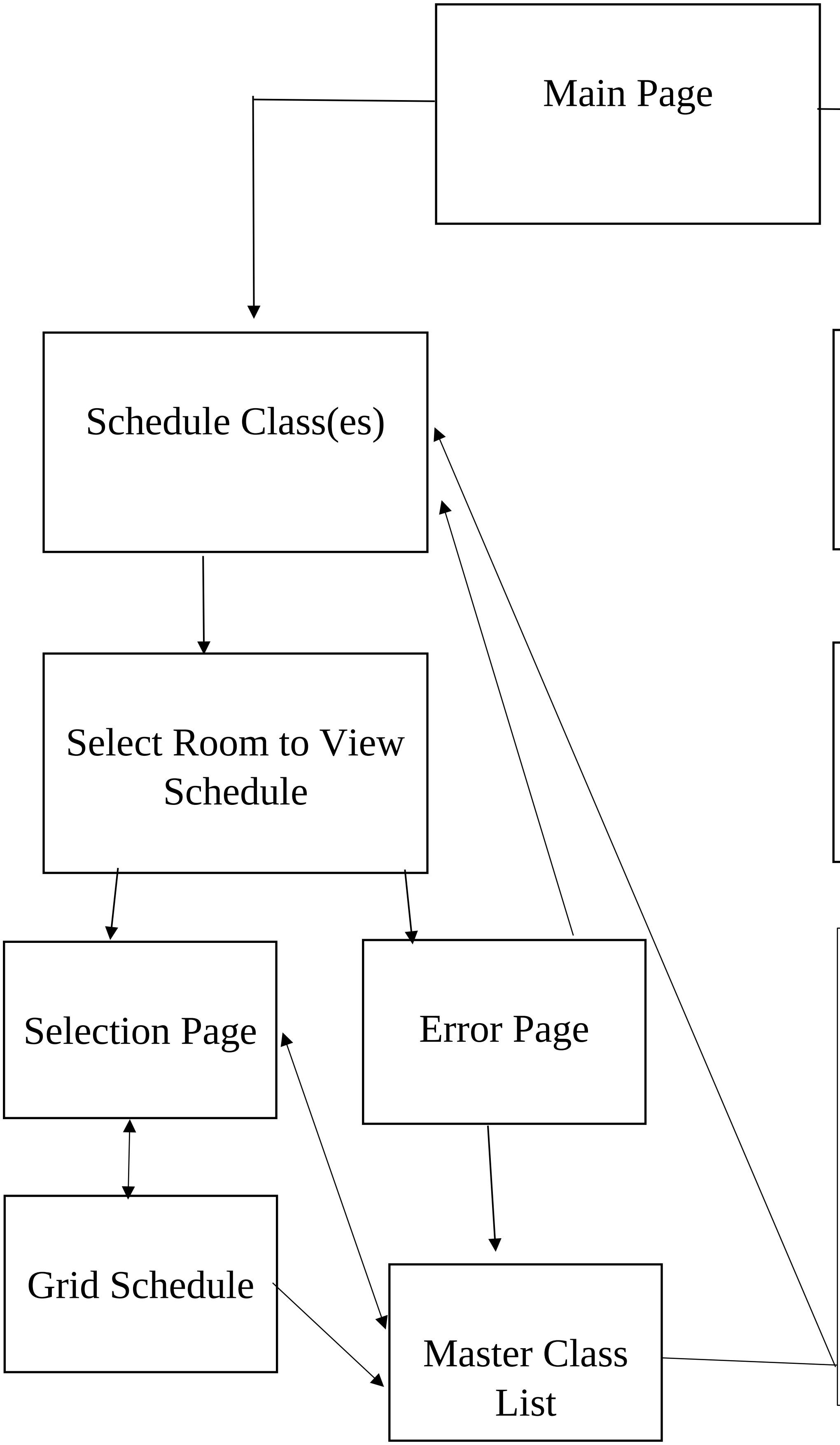

roomweek
roomno VARCHAR(10)
room availibility VARCHAR(10)
monday INT(11)
tuesday INT(11)
wednesday INT(11)
thursday INT(11)
friday INT(11)
saturday INT(11)
sunday INT(11)
Indexes
scheduledtimes
secno INT(5)
DayMeet1 VARCHAR(10)
DayMeet2 VARCHAR(10)
DayMeet3 VARCHAR(10)
StartTime1 VARCHAR(15)
StartTime2 VARCHAR(15)
StartTime3 VARCHAR(15)
EndTime1 VARCHAR(15)
EndTime2 VARCHAR(15)
EndTime3 VARCHAR(15)
Indexes
sectiondetails
secno INT(5)
roomno VARCHAR(10)
cono VARCHAR(10)
Indexes
coursedetails
cono VARCHAR(10)
labhr INT(5)
lechr INT(5)
Indexes

Group D
Term Project
CSC 350-1200
Professor Kok
May 18, 2020

Use Case Diagram



Logic Diagram of User Interface



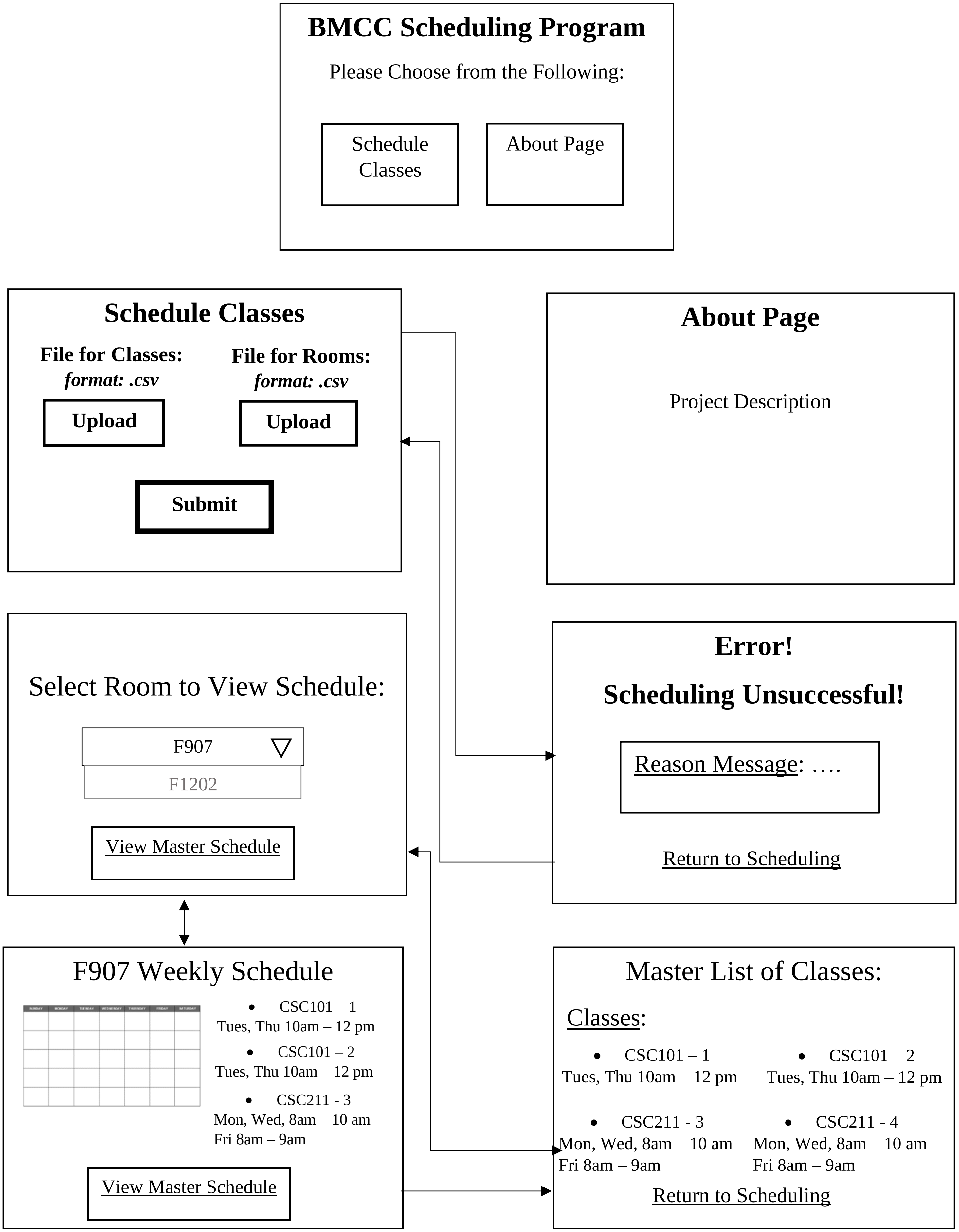
Notes:

Clicking on "BMCC Class Scheduling" link at the top of every page returns user to Main Page.

The first level is the main page.

The second level are the button options from the main page. Based on what the user selects, he/she will be lead to various webpages.

Third level and fourth level are the web pages displayed and their respective sub-webpages.



BMCC Scheduling Program

Please Choose from the Following:

Schedule Classes

File for Classes: ***format: .csv***

File for Rooms: ***format: .csv***

About Page

Project Description

Error!

Scheduling Unsuccessful!

Reason Message: ….

Select Room to View Schedule:

F907 Weekly Schedule

- CSC101 – 1
  Tues, Thu 10am – 12 pm
- CSC101 – 2
  Tues, Thu 10am – 12 pm
- CSC211 - 3
  Mon, Wed, 8am – 10 am
  Fri 8am – 9am

Master List of Classes:

Classes:

- CSC101 – 1
  Tues, Thu 10am – 12 pm
- CSC101 – 2
  Tues, Thu 10am – 12 pm
- CSC211 - 3
  Mon, Wed, 8am – 10 am
  Fri 8am – 9am
- CSC211 - 4
  Mon, Wed, 8am – 10 am
  Fri 8am – 9am

Group D
CSC 350-1200
Professor Kok
May 20, 2020

Database Dictionary

Table: coursedetails

Field Name	Data Type	Field Size	Description	Example
cono	varchar	10	Course number from BMCC CIS Department	CSC350
labhr	int	5	Total weekly lab hours for a specific course within CIS Department	3
lechr	int	5	Total weekly lecture hours for a specific course within CIS Department	2

Table: scheduledtimes

Field Name	Data Type	Field Size	Description	Example
secno	int	5	Unique section number for specific class and is attached to end of specific course type that was scheduled	1
DayMeet1	varchar	10	First day that a specific class meets	Monday

DayMeet2	varchar	10	Second day that a specific class meets	Wednesday
DayMeet3	varchar	10	Third day that a specific class meets	Friday
StartTime1	varchar	15	Starting time on first day of the class meets	8:00 AM
StartTime2	varchar	15	Start time on second day of the class meets	9:00 AM
StartTime3	varchar	15	Start time on third day of the class meets	10:00 AM
EndTime1	varchar	15	End time on first day of the class meets	10:00 AM
EndTime2	varchar	15	End time on second day of the class meets	11:00 AM
EndTime3	varchar	15	End time on third day of the class meets	12:00 PM

Table: sectiondetails

Field Name	**Data Type**	**Field Size**	**Description**	**Example**
secno	int	5	Unique section number for specific class and is attached to	1

			end of specific course type that was scheduled	
roomno	varchar	10	Room number that a class section meets in. For all days that a class section meets, it will be meeting in the same room unless otherwise stated.	F906
cono	varchar	10	Course number from BMCC CIS Department	CSC350

Table: roomweek

Field Name	**Data Type**	**Field Size**	**Description**	**Example**
roomno	varchar	10	Room number that a class section meets in. For all days that a class section meets, it will be meeting in the same room unless otherwise stated.	F906
roomavailability	varchar	10	Indicates the availability of a specific room based on the cumulative percentage of hourly time slots occupied over the total number of hourly time slots per room	Yes/No

monday	int	11	Stores the cumulative number of occupied hourly time slots on Monday based on the number of classes scheduled in specific room this far	13
tuesday	int	11	Stores the cumulative number of occupied hourly time slots on Tuesday based on the number of classes scheduled in specific room this far	8
wednesday	int	11	Stores the cumulative number of occupied hourly time slots on Wednesday based on the number of classes scheduled in specific room this far	12
thursday	int	11	Stores the cumulative number of occupied hourly time slots on Thursday based on the number of classes scheduled in specific room this far	10
friday	int	11	Stores the cumulative number of occupied hourly time slots on Friday based on the number	6

			of classes scheduled in specific room this far	
saturday	int	11	Stores the cumulative number of occupied hourly time slots on Saturday based on the number of classes scheduled in specific room this far	6
sunday	int	11	Stores the cumulative number of occupied hourly time slots on Sunday based on the number of classes scheduled in specific room this far	0

CSC211	3
CSC410	2
CIS445	1
CIS220	3
CIS100	3
CSC211	2
CSC111	2
CSC111	3
CSC331	1
CSC350	1
CIS490	3
CSC110	3
CSC110	1
CSC210	2
CSC210	2
CIS235	2
CIS359	3
CIS385	3
CIS440	2
CIS475	1
CIS475	1
CSC310	2
CSC310	3
CSC430	3
CIS395	3
CIS316	2
CSC230	2
CSC230	2
CIS490	2
CIS316	3
CSC215	3
CSC210	3
CSC331	3
CSC430	1
CIS495	1
CIS335	1
CIS490	2
CSC331	3
CSC410	3
CSC101	2
CSC101	3
CSC101	1
CSC230	1
CIS395	1
CIS325	3
CIS220	3
CIS180	2

CIS420	2
CSC331	2
CSC330	3
CSC330	2
CSC350	1
CSC350	3
CSC350	2
CIS100	3
CIS100	3
CIS362	3
CIS359	3
CIS316	2
CIS100	2
CSC470	3
CSC470	1
CSC410	1
CSC430	1
CSC101	3
CSC110	3
CSC215	3
CSC215	3
CSC215	2
CIS465	2
CIS475	2
CSC410	3
CSC110	3
CSC111	3
CSC111	1
CSC310	1
CSC310	1
CSC350	1
CSC430	1
CSC215	2
CIS359	3
CIS255	3
CIS100	2
CIS100	2
CIS100	2
CSC450	1
CSC450	1
CSC210	1
CSC215	3
CSC215	3
CSC211	1
CIS280	2
CIS317	3
CIS385	3

CIS370	2
CIS395	3
CSC210	2
CSC350	2
CSC330	2
CIS440	1

F1201
F1001
F907
M1010
F1102
M907
F1204
F1202
F805
F901
F701
M1207
F910

```
//CSC350GroupD_Index.php source code for PDF

<html>
<head>
    <title>BMCC Scheduling Program</title>
    <style>
            h1
            {
                text-align: center;
                font-size: 40px;
            }

            .Submitbtn
            {
                font-size: 32px;
                font-family: "Times New Roman";
                text-align: center;
                height:150px;
                width:650px;
                float:left;
                margin: 0 auto;

            }

            p.paragraph
            {
                font-size: 32px;
                font-family: "Times New Roman";
                text-align: center;
                text-decoration: underline;
            }

    </style>
</head>

<h1><a href= "CSC350GroupD_Index.php">BMCC Scheduling Program </a></h1>
<body>
<p class = "paragraph"> Please Choose From the following: </p>
    <form action="CSC350GroupD_Index.php" method="post">
        <tr>
                <tr>
                    <td colspan = "2">&nbsp;
                    <a href= "CSC350GroupD_scheduling.php">
                    <input type="button" value="Schedule Classes"
name="ScheduleClasses" class = "Submitbtn"/></a></td>
                </tr>

                <tr>
                    <td colspan = "2">&nbsp;
                    <a href= "CSC350GroupD_aboutpage.php">
                    <input type="button" value="About Page"
name="AboutPage" class = "Submitbtn"/></a></td>
                </tr>
            </tr>
        <table>

    </form>


<?php

?>


</body>
</html>
```

//CSC350GroupD_aboutpage.php source code for PDF

```
<html>
<head>
    <title>BMCC Scheduling Program</title>
    <style>
        h1
        {
            text-align: center;
            font-size: 40px;
        }

        h2
        {
            text-align: center;
            font-size: 26px;
        }

        p.paragraph
        {
            text-align: center;
            font-size: 20px;
            line-height: 1.6;
            font-family: "Times New Roman";
        }

        p.ptitle
        {
            text-align: center;
            font-size: 20px;
            text-decoration: underline;
            font-family: "Times New Roman";
        }

    </style>
</head>

<h1><a href= "CSC350GroupD_Index.php">BMCC Scheduling Program </a></h1>
<h2>About Page</h2>

<body>
    <p>
        <br>Keyshawn Harinarain
```

```html
                <br>Shirley Ni
                <br>Juan Peguero
                <br>Group D Project Description
                <br>Professor Kok</br>
        </p>

        <p class = "ptitle"><br>Project Purpose:</br></p>
        <p class = "paragraph">

        <br>
        To create a program that takes a user input of room numbers, class numbers, and days a class is
        to be scheduled, and output a class schedule for the rooms given. This project is also, to determine our
        understanding of the software development process and the different phases of development. This
        begins with planning and organizing our approach as well as consistently updating a GANTT chart to
        plan, manage, and control the time for different phases of the process. This is a base level of our project
        and provides the ability to set time expectations and deadlines, as well as account for various setbacks
        that can occur along the process.
        </br>

        <br>
        This project then goes into the development process itself, which involves designing a program
        at the very base level, this starts with how data flows in our program. To accomplish this, we use three
        levels of Data Flow Diagrams. The base level, or context, shows a high-level view of the data flow
        process, and how a user data moves through our program. The next level, level zero, shows a more
        detailed view of the process and allows a use to follow the data as it goes from the start of its flow, to
        the end output, and the various processes that occur in between. The level one diagram is an in-depth
        view of our scheduling process, how we manage data, what happens if an input meets a certain criteria,
        how does this class number, days to be scheduled, and room number actually turn into a scheduled
        class.
```

```
	</br>

	<br>
	This isn't our only process, software development occurs in different stages, and
diagrams like
	entity-relationship diagrams and use-case diagrams show how users interact with our
software. How
	does a client use our program? What happens when a client selects a certain button or
uploads their
	information? How does this entity (user or client) engage with our software? These
diagrams showcase
	this interaction and it is crucial to the design process to understand how a user can
interact with our
	software. This project is a look into the software development process, from how users
interact with our
	program, to the planning and time management aspects, testing, users’ interactions, and
the actual user
	interface, this project is an entire view of how software goes from an idea from a
brainstorming session,
	to a final product.
	</br>
	</p>
</body>
</html>
```

```
// CSC350GroupD_scheduling.php source code for PDF

<html>
	<head>
		<title>BMCC Scheduling Program</title>
		<style>
			h1
			{
				text-align: center;
				font-size: 40px;
			}
			table
			{
				text-align: center;
				font-size: 32px;
				width: 100px;
				height: 100px;
				margin: 0 auto;
			}
			.Uploadbtn
			{
				font-size: 20px;
				font-family: "Times New Roman";

				height: 34px;
				margin: 0 auto;
				float:left;
			}

			.Submitbtn
			{
				font-size: 32px;
				font-family: "Times New Roman";
				text-align: center;
				height:50px;
				width:650px;
				float:left;
				margin: 0 auto;

			}

			body
			{
```

```
                text-align: center;
            }

        </style>
    </head>
<body>
<h1><a href= "CSC350GroupD_Index.php">BMCC Scheduling Program </a></h1>
<h1>Schedule Classes</h1>
<form action="CSC350GroupD_ProcessScheduling.php" method="post"
enctype="multipart/form-data">
    <table>
            <tr>
                <td><label for="file">File for Classes</label></td>
                <td><label for="file">File for Rooms</label></td>
            </tr>

            <tr>
                <td><label for="file">format: .csv</label></td>
                <td><label for="file">format: .csv</label></td>
            </tr>

            <tr>
                <!-- only allows .csv to be be uploaded -->
                <td><input type="file" accept = ".csv" name="classes_to_upload"
id="classes_to_upload" class ="Uploadbtn"/> </td>
                <td><input type="file" accept = ".csv" name="rooms_to_upload"
id="rooms_to_upload" class ="Uploadbtn"/> </td>
            </tr>


            <!-- merges 2 cells to fit submit button into -->
            <tr><td colspan = "2">&nbsp;<input type="submit" value="Submit"
name="Submit" class = "Submitbtn"/></td></tr>
    </table>

</form>
</body>
</html>
```

```
//CSC350GroupD_ProcessScheduling.php for pdf

<html>
<body>
<table border="1">

<?php
    class schedule
    {
        private $hr;
        private $dys;
        private $coName;

        private $servername = "localhost";
        private $username = "root";
        private $password = "";
        private $dbname = "scheduling";

        public function __construct()
        {
            // Create connection
            $conn = new mysqli($this->servername, $this->username,
$this->password, $this->dbname);
            // Check connection
            if ($conn->connect_error)
                die("Connection failed: " . $conn->connect_error);
            else
                echo "<br>Connection Successful</br>";
        }

        public function readClasses()
        {
            if(isset($_POST['Submit']))
            {

                $target_dir = "C:/xampp/htdocs/uploads";
                $target_file = $target_dir .
basename($_FILES["classes_to_upload"]["name"]);
                $uploadOk = 1;
                $image_file_type =
pathinfo($target_file,PATHINFO_EXTENSION);
```

```
                // Check if file already exists
                if (file_exists($target_file))
                {
                    echo "File already present.";
                    $uploadOk = 0;
                }

                // Check file size
                elseif ($_FILES["classes_to_upload"]["size"] > 1000000)
                {
                    echo "File too big.";
                    $uploadOk = 0;
                }

                // Check if $upload_ok is set to 0 by an error
                elseif ($uploadOk == 0)
                {
                    echo "Your file was not uploaded.";
                // If all the checks are passed, file is uploaded
                }

                else
                {
                    /*      creates a copy of file and renames it as
uploads[file_name]     */
                    if
(move_uploaded_file($_FILES["classes_to_upload"]["tmp_name"], $target_file))
                        echo "The file ". basename(
$_FILES["classes_to_upload"]["name"]). " was uploaded.";

                    else
                        echo "A error has occured uploading.";
                }

                $conn = mysqli_connect("localhost", "root", "", "scheduling");

                if($conn)
                {
                    $file = $_FILES["classes_to_upload"]["tmp_name"];
                    $handle = fopen($file,"r");
                    while(($cont=fgetcsv($handle,1000,","))!==false)
                    {
                        $hr = 0;
```

```
/*      Selects data for hourly distribution     */
$sql = "SELECT cono, labhr+lechr AS total FROM coursedetails WHERE  cono = '$cont[0]'";
if($result = mysqli_query($conn, $sql))
{

if (mysqli_num_rows($result) > 0)
{
        echo "<table border = 1>";
        echo "<tr>";
            echo "<th>cono</th>";
            echo "<th>total</th>";
            //echo "<th>labhr</th>";
            //echo "<th>lechr</th>";
        echo "</tr>";
    while($row = mysqli_fetch_array($result))
     {
        echo "<tr>";
        echo "<td>" . $row['cono'] . "</td>";
        $hr = $row['total'];
        echo "<td>" . $row['total'] . "</td>";
     }


        // Free result set
        mysqli_free_result($result);


    /*Performing Hourly Distribution*/
    $dys = $cont[1];
    $coName = $cont[0];
    $daysDistr = array();
    if(!is_Numeric($hr) || !is_Numeric($dys))
    echo "Invalid input. Numeric values only for total course hours and days to be scheduled.";

    elseif(is_float($hr) || is_float($dys))
        echo "Invalid input. Decimal values are not accepted for total course hours and days to be scheduled.";

    elseif($hr <= 0 || $dys <= 0)
```

```
                                              echo "Invalid input. Numeric values
must be nonnegative values.";

                                        elseif($hr < $dys)
                                              echo "Total courses must be greater
than days to be scheduled.";

                                        else
                                        {
                                              $sum = 0; //holds value of
accumulated sum in array
                                                    $temp1 = $hr;
                                                    $temp2 = $dys;
                                                    for($i = 0; $i <= $dys - 1;
$i++)
                                                    {
                                                          $quotient =
round($temp1/$temp2, 0, PHP_ROUND_HALF_UP);
                                                          //echo
$temp1."/".$temp2."<br></br>";
                                                          $daysDistr[$i] =
$quotient;

                                                          $numerator = $temp1
- $quotient;
                                                          $denominator =
$temp2 - 1;
                                                          $temp1 =
$numerator;
                                                          $temp2 =
$denominator;
                                                    }
                                        }

                                        /* Print Hourly Distribution Array*/
                                        $withComma = implode(",", $daysDistr);
                                        //echo "<tr>";
                                              echo "<td>" . $withComma . "</td>";
                                        echo "</tr>";
                                        echo "</table>";
                                        echo "<br></br>";
```

```
                            /*starts scheduling sections by going into
roomweek and checking for available hourly time slots*/
                            /*$sql2 = "SELECT roomno FROM
roomweek WHERE roomavailability = 'Yes' ";
                            if($result = mysqli_query($conn, $sql))
                            {
                                if (mysqli_num_rows($result) > 0)
                                {
                                    for($i = 0; i <
sizeof($daysDistr); $i++)
                                    {
                                        $daysofWeek =
array("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday");
                                        for($j = 0; j <
sizeof($daysofWeek); $j++)
                                        {
                                            $counter = 0;


while($row['$dayofWeek[$j]'] <= 10)
                                            {


$counter += $daysDistr[$i];

$insertSQL = "INSERT INTO roomweek ('$j') VALUES('$row['$dayofWeek[$j]']')";

                                                break;
                                            }
                                        }
                                    }
                                }
                        } */
                    }

                        else
                            echo "0 results";

                        }
                    }
                }
            }
        }
        public function readRooms()
        {
            if(isset($_POST['Submit']))
            {
                $target_dir = "C:/xampp/htdocs/uploads";
                $target_file = $target_dir .
basename($_FILES["rooms_to_upload"]["name"]);
                $uploadOk = 1;
                $image_file_type =
pathinfo($target_file,PATHINFO_EXTENSION);


                // Check if file already exists
                if (file_exists($target_file))
                {
                    echo "File already present.";
                    $uploadOk = 0;
                }


                // Check file size
                elseif ($_FILES["rooms_to_upload"]["size"] > 1000000)
                {
                    echo "File too big.";
                    $uploadOk = 0;
                }

                // Check if $upload_ok is set to 0 by an error
                elseif ($uploadOk == 0)
                {
                    echo "Your file was not uploaded.";
                // If all the checks are passed, file is uploaded
                }

                else
                {
                    /*      creates a copy of file and renames it as
uploads[file_name]    */
                    if
(move_uploaded_file($_FILES["rooms_to_upload"]["tmp_name"], $target_file))
                        echo "The file ". basename(
$_FILES["rooms_to_upload"]["name"]). " was uploaded.";
```

```
                    else
                        echo "A error has occured uploading.";
                }
            $conn = mysqli_connect("localhost", "root", "", "scheduling");

            if($conn)
            {
                $file = $_FILES["rooms_to_upload"]["tmp_name"];
                $handle = fopen($file,"r");
                while(($cont=fgetcsv($handle,1000,","))!==false)
                {

                    $table=rtrim($_FILES["rooms_to_upload"]["tmp_name"],
".csv");

                        $query = "INSERT INTO roomweek (roomno,
roomavailibility, monday, tuesday, wednesday, thursday, friday, saturday, sunday)
                        VALUES ('$cont[0]', 'Yes', '0', '0', '0', '0', '0', '0', '0')";
                        echo $query;
                        mysqli_query($conn, $query);
                }
            }
            }
        }

        public function makeGrid()
        {
            $daysofWeek = array(" ", "Monday", "Tuesday", "Wednesday",
                                    "Thursday", "Friday", "Saturday",
"Sunday");


            $hours = array("8:00 am", "9:00 am", "10:00 am", "11:00 am", "12:00 pm",
                                "1:00 pm", "2:00 pm", "3:00 pm", "4:00 pm",
"5:00 pm",
                                "6:00 pm", "7:00 pm", "8:00 pm", "9:00 pm",
"10:00 pm");

            for ($j = 0; $j <= sizeof($daysofWeek) - 1; $j++)
                echo "<td><b>$daysofWeek[$j]</b></td>";

            for ($i = 0; $i <= sizeof($hours) - 1 ; $i++)
```

```
                echo "<tr></tr> <td><b>$hours[$i]</b></td> <td></td>
                        <td></td> <td></td>
                        <td></td> <td></td>
                        <td></td> <td></td>";
        }



        public function printMasterList()
        {
            $conn = mysqli_connect("localhost", "root", "", "scheduling");

            if($conn)
            {

                $query1 = "SELECT *, sectiondetails.cono FROM scheduledtimes
JOIN sectiondetails ON sectiondetails.secno = scheduledtimes.secno";
                $result = mysqli_query($conn, $query1);


                echo "<table border = 1>"; // start a table tag in the HTML

                while($row = mysqli_fetch_array($result))
                {   //Creates a loop to loop through results
                    echo "<tr><td>" . $row['cono'] . "</td><td>" . $row['secno'] .
"</td><td>" . $row['DayMeet1'] .
                        "</td><td>" . $row['DayMeet2'] . "</td><td>".
$row['DayMeet3'] .
                        "</td><td>". $row['StartTime1'] . "</td><td>".
$row['StartTime2'] .
                        "</td><td>".$row['StartTime3'] . "</td><td>".
$row['EndTime1'] .
                        "</td><td>".$row['EndTime2'] .
"</td><td>".$row['EndTime3'] . "</td></tr>";
                }

                echo "</table>"; //Close the table in HTML
            }
            //mysql_close(); //Make sure to close out the database connection
        }


    }
```

```
/*      link: https://www.w3schools.com/php/php_mysql_select.asp
        username and password is same as MySQL workbench root login info
        $dbname is scheduling
*/

if(isset($_POST['Submit']))
{


/*      After uploading both classes.csv and rooms.csv files onto the upload page, it will take
        a few minutes for the file to appear in htdocs. Once both files appear in the htdocs folder,
        then the other parts of the schedule processing will run smoothly.*/

        $class1 = new schedule();
        $class1->readClasses();
        $class1->readRooms();


        //$class1->printMasterList();
}

?>

</table>
</body>
</html>
```

```
-- phpMyAdmin SQL Dump
-- version 5.0.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: May 19, 2020 at 04:17 AM
-- Server version: 10.4.11-MariaDB
-- PHP Version: 7.4.1

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `scheduling`
--

-- --------------------------------------------------------

--
-- Table structure for table `coursedetails`
--

CREATE TABLE `coursedetails` (
  `cono` varchar(10) NOT NULL,
  `labhr` int(5) NOT NULL,
  `lechr` int(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `coursedetails`
--

INSERT INTO `coursedetails` (`cono`, `labhr`, `lechr`) VALUES
('CIS100', 0, 4),
('CIS115', 2, 2),
('CIS120', 0, 3),
('CIS140', 0, 3),
('CIS155', 2, 3),
('CIS160', 0, 3),
('CIS165', 2, 2),
('CIS180', 0, 4),
('CIS200', 2, 2),
('CIS207', 4, 2),
('CIS220', 0, 4),
('CIS235', 0, 5),
('CIS255', 2, 4),
('CIS280', 0, 4),
('CIS316', 2, 2),
('CIS317', 0, 4),
('CIS325', 0, 4),
('CIS335', 0, 5),
('CIS345', 3, 2),
('CIS359', 2, 2),
('CIS362', 2, 2),
('CIS364', 2, 2),
('CIS365', 0, 5),
('CIS370', 2, 2),
('CIS385', 2, 2),
('CIS390', 2, 2),
('CIS395', 3, 2),
('CIS420', 0, 5),
('CIS440', 0, 4),
('CIS445', 3, 2),
('CIS459', 3, 2),
('CIS465', 5, 5),
('CIS475', 2, 3),
('CIS480', 0, 3),
('CIS485', 2, 2),
('CIS490', 2, 2),
('CIS495', 2, 2),
('CSC101', 2, 2),
('CSC110', 0, 5),
('CSC111', 2, 3),
('CSC210', 3, 2),
('CSC211', 3, 2),
('CSC215', 2, 2),
('CSC230', 3, 3),
('CSC231', 0, 4),
('CSC310', 0, 4),
('CSC330', 0, 4),
('CSC331', 4, 1),
('CSC350', 3, 2),
('CSC410', 0, 4),
('CSC430', 0, 4),
('CSC450', 0, 4),
('CSC470', 2, 3),
('GIS101', 2, 2),
('GIS201', 3, 3),
('GIS261', 2, 2),
('GIS325', 0, 15),
('GIS361', 2, 2);

-- --------------------------------------------------------

--
-- Table structure for table `roomweek`
--

CREATE TABLE `roomweek` (
  `roomno` varchar(10) NOT NULL,
  `roomavailibility` varchar(10) DEFAULT NULL,
  `monday` int(11) DEFAULT NULL,
  `tuesday` int(11) DEFAULT NULL,
  `wednesday` int(11) DEFAULT NULL,
  `thursday` int(11) DEFAULT NULL,
  `friday` int(11) DEFAULT NULL,
  `saturday` int(11) DEFAULT NULL,
  `sunday` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- --------------------------------------------------------

--
-- Table structure for table `scheduledtimes`
--

CREATE TABLE `scheduledtimes` (
  `secno` int(5) NOT NULL,
  `DayMeet1` varchar(10) NOT NULL,
  `DayMeet2` varchar(10) DEFAULT NULL,
  `DayMeet3` varchar(10) DEFAULT NULL,
  `StartTime1` varchar(15) NOT NULL,
  `StartTime2` varchar(15) DEFAULT NULL,
  `StartTime3` varchar(15) DEFAULT NULL,
  `EndTime1` varchar(15) NOT NULL,
  `EndTime2` varchar(15) DEFAULT NULL,
  `EndTime3` varchar(15) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- --------------------------------------------------------

--
-- Table structure for table `sectiondetails`
--

CREATE TABLE `sectiondetails` (
  `secno` int(5) NOT NULL,
  `roomno` varchar(10) NOT NULL,
  `cono` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Indexes for dumped tables
--

--
-- Indexes for table `coursedetails`
--
ALTER TABLE `coursedetails`
  ADD PRIMARY KEY (`cono`);

--
-- Indexes for table `roomweek`
--
ALTER TABLE `roomweek`
  ADD PRIMARY KEY (`roomno`);

--
-- Indexes for table `scheduledtimes`
--
ALTER TABLE `scheduledtimes`
  ADD PRIMARY KEY (`secno`);

--
```

```
-- Indexes for table `sectiondetails`
--
ALTER TABLE `sectiondetails`
  ADD PRIMARY KEY (`secno`,`roomno`,`cono`),
  ADD KEY `roomno_idx` (`roomno`),
  ADD KEY `cono_idx` (`cono`);

--
-- Constraints for dumped tables
--

--
-- Constraints for table `sectiondetails`
--
ALTER TABLE `sectiondetails`
  ADD CONSTRAINT `cono_idx` FOREIGN KEY (`cono`) REFERENCES `coursedetails`
(`cono`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `roomno_idx` FOREIGN KEY (`roomno`) REFERENCES `roomweek`
(`roomno`),
  ADD CONSTRAINT `secno_idx` FOREIGN KEY (`secno`) REFERENCES `scheduledtimes`
(`secno`) ON UPDATE CASCADE;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```