

CSCI 5633-SPRING 19

Meeting Room Booking System

Report

By

Group 3

Binal Patel (Id: 26)

Kushani Shah (Id: 28)

Arhat Baid (Id: 23)

Mayuri Amrutkar (Id: 13)

Kelly Corriea (Id: 15)

Parvez Salim (Id: 30)

Meeting Room Booking System

Abstract

Generally, being a employee in organization, there are lots of problems we face. One of the problems we encounter was whenever employee need conference room, each time they need to go to meet lab assistant. To overcome from that, we introduced an asp.net based web portal named “Meeting Room Booking System”, through which you can reserve your conference room online.

An online web portal developed in asp.net environment where employees can reserve the conference room in advance for various purposes such as project meetings, group meetings, project discussion. To reserve a conference room in advance, employees need to login with their organization email id, specify the purpose for reserving a conference room and select the date and time for when they need to reserve the conference room with required duration. Once the employees register themselves, the system will allocate them a conference room number. And once their allocated time gets completed, they will be automatically logged out from the register and the conference room gets allocated to the next employee request. This keeps the employees more organized with an ease of access making the best use out of the resources.

TABLE OF CONTENTS

Sr. No.	Title	Page No.
	Abstract	3
	List of Contents	4
1	Introduction	5
2	Testing user name/password	5
3	Special features	6
4	Lessons learned	6
5	Flow chart of Menus	7
6	Project Description	8
7	Steps for development	12
8	Steps for deploying	12
9	Group Home page	13
10	Scope for future Enhancements	13
11	Code	14

1. Introduction

‘Meeting Room Booking System’ is the web application in .NET framework which is the project that aims at creating an Online Room Reservation system which can be used by employees to reserve conference rooms in advance for various purposes such as group study, group meetings. This online system manages the study rooms at various locations for employees and managers. If the user logs into system he will be able to see all the reservations made on that day, month and year.

- Users can check the availability of conference rooms.
- The users can register and log into the system.
- The user will be able to know the details of reservations.
- User can be able to edit and delete the reservation

For this application the hardware and software requirements are:

- 1024x768 resolution or higher is required for the core products.
- Dual Core 1.6GHz or faster with RAM: 1 gigabyte (GB) (32-bit) or 2 GB (64-bit).
- Beta versions of Operating Systems or Web Browsers are not supported. All system requirements are subject to change based on product version.
- Operating System: Windows 7/8/10 OR ABOVE
- Server: MYSQL SERVER Database: SQL or MYSQL
- IDE: Visual Studio 2017

2. Testing user name or password

In order to login into the application, the user must enter the username, password and must select the class which they belong.

Login credentials for user:

Username: (example user name: binal) , Password: user’s password (example password: **1234**)
Username: (example user name: kushani) , Password: user’s password (example password: **1234**)
Username: (example user name: parvez) , Password: user’s password (example password: **1234**)
Username: (example user name: mayuri) , Password: user’s password (example password: **1234**)
Username: (example user name: kelly) , Password: user’s password (example password: **1234**)
Username: (example user name: arhat) , Password: user’s password (example password: **1234**)

If the user enters incorrect email id or password, it will send a show a notification on screen that contains some error message.

3. Special features of the project

USER MODULE:

- Check room accessibility.
- Fill enrollment structure.
- Can login with his/her own client id and secret phrase.
- Apply channels and view rooms accessible in every determination made.
- Get data about held rooms (if room is now saved).
- Check subtleties of rooms held for the present day.
- Check subtleties of rooms held for the coming days
- Update and view client subtleties.

4. Lessons learned

While actualizing the undertaking we took in a ton of things. Some of them are:

- How to assemble the necessities for the framework to execute.
- How to configuration, set up and distribute a database in the web.
- Creating and coordinating database.
- Storing information in the database and utilization of sessions.

- Dealing with database disappointments and mistake taking care of.
- Handling issues of information irregularity promotion memory distribution.

Obtained learning on occasion dealing with and code improvement strategies while actualizing in code behind.

5. Flow chart of menus

In this application at first, Home page shows up. In the event that the client doesn't have a record, at that point he should enlist himself. After enrollment he can ready to login into his record, at that point it will explore to room reservation page as indicated by the login accreditations. In the room reservation client can,

- Add a booking
- Delete a booking
- Edit a booking
- Move and expand a booking

When the client finishes their Registration procedure, an initiation will be sent by the framework to client's email id, from where they have to enact their record.

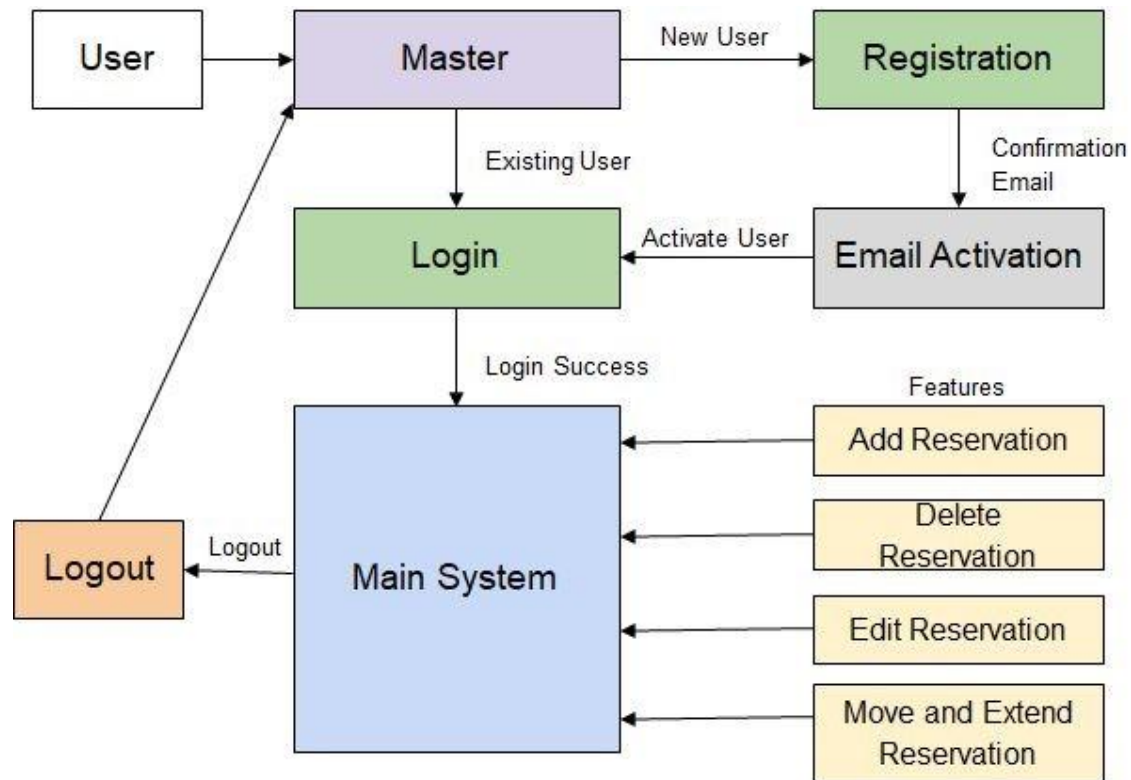


Figure: 1.1Flow Chart

6. Project Description

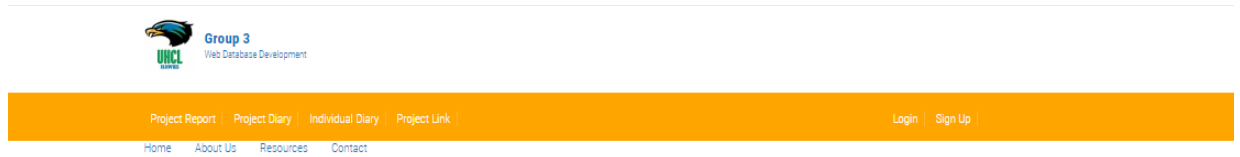
This application mainly consists of following,

1. Home page
2. Register Page
3. Login page
4. Room reservation page

Home Page:

In this page, user can able to know the details of our web product and the contact details.

Meeting Room Booking System



Meeting Room Booking System

An online web portal developed in asp.net environment where employees can reserve the meeting room in advance for various purposes such as project meeting, group meetings, project discussion. To reserve a meeting room in advance, employee need to login with their organization email id, specify the purpose for reserving a meeting room and select the date and time for when they need to reserve the meeting room with required duration. Once the employee register themselves, the system will allocate them a particular meeting room number. And once their allocated time gets completed they will be automatically logged out from the register and the meeting room gets allocated to the next employee request. This keeps the employee more organized with an ease of access making the best use out of the resources.



Login page:

This is the page in which the client can login into the application. Here are a few approvals like on the off chance that we enter wrong secret word or username it will show some message like 'Invalid Account'

Meeting Room Booking System

Username

Password

☐ Remember me

Login

New User Register Here!

Register Page:

This is the page in which the client can make a record into the application. Making a record is a 2-step process.

Step-1: User must enlist himself by entering his subtleties.

Step-2: After enrollment an actuation email sent to the email referenced in the register page. Client must enact their separate record utilizing that connect.

User Registration

Username

Password

Confirm Password

Email

Register

Login Here

Room reservation page:

Meeting Room Booking System

WDD SPRING19 Group 03 | UHCL | Logout

Show rooms: All

DEMO

Room	RoomCoc	Size	Capacity	Projector	Location	Status	28	29	30	1	2	3	4	5	6
Conference Room 1	1	50 sqfts	50 Persons	Y	Building A1	Ready				WDD Project Meeting (5/1/2019 - 5/2/2019) Checked out			Business Meeting (5/4/2019 - 5/5/2019) New		
											Testing Meeting (5/2/2019 - 5/3/2019) Expired (not confirmed in time)				
Conference Room 2	2	75 sqfts	60 Persons	Y	Building A2	Available			Business Meeting (4/30/2019 - 5/1/2019) Late arrival						
Conference Room 3	3	100 sqfts	85 Persons	Y	Building A3	Booked							TA Meeting (5/4/2019 - 5/5/2019) New		
Conference Room 4	4	120 sqfts	100 Persons	Y	Building A3	Ready						IEEE Meeting (5/3/2019 - 5/6/2019) Arrived			
Conference Room 5	5	125 sqfts	105 Persons	Y	Building A5	Ready		Project Meeting (4/29/2019) Late arrival							
Conference Room 6	6	60 sqfts	70 Persons	Y	Building A4	Booked						Development Meeting (5/3/2019 - 5/4/2019) Confirmed			
Conference Room 7	7	80 sqfts	89 Persons	N	Building A6	Available								User Experience Meeting (5/5/2019 - 5/6/2019) Confirmed	
Conference Room 8	8	10 sqfts	15 Persons	Y	Building A5	Available				Security Meeting (5/1/2019 - 5/2/2019) Late checkout					

Making a new reservation:

New Reservation

Start: 4/27/2018

Start Time: 4:00 PM

End: 4/28/2018

End Time: 7:00 PM

Room: BAYOU103

Name: Personal - Sheeyam

OK

Cancel

Edit current reservation:

Edit Reservation
[Delete](#)

Start: 4/29/2018 Start Time: 12:00 PM
 End: 4/30/2018 End Time: 9:00 PM

Resource: BAYOU101
 Name: Yoga Event for UG
 Reserved By: sheeyam
 Status: New

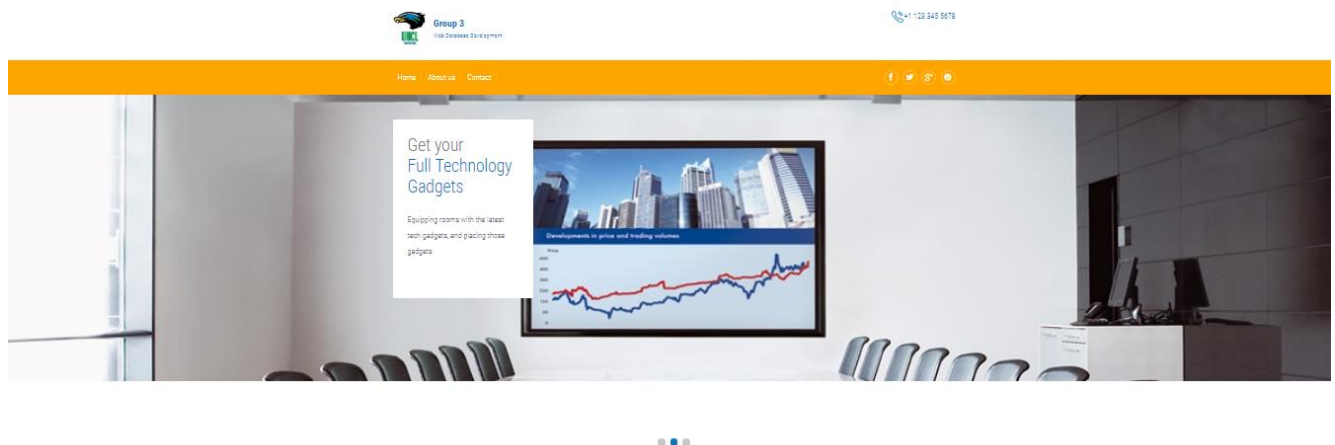
Review Reservation:

This page encourages the client to check the accessibility and hold the room, by entering the fundamental subtleties, for example, individuals, look at in and check date, email and phone.

On the off chance that the client isn't login out of the blue, at that point they can see the accessibility of the rooms as it were. They can't hold the room.

About us:

This page specifies the user, the details about the Study Room Reservation System.



Contact:

User can contact the management by entering the user details and a detailed message.

Our Team



Binal Patel
26



Kushani Shah
28



Arhat Baid
23



Mayuri Amrutkar
13



Kelly Correia
15



Parvez Salim
30

8. Steps for development

So as to build up an application utilizing .NET system we should introduce visual studio. This might be any rendition like visual studio 2017 or higher.

1. First we need to go to begin visual studio 2017 - > File - > New - > Project
2. At that point a menu will show up. In that select visual c # and ASP.NET web application
3. Click on web structures and snap on 'alright'.
4. Then compose your code in that venture

9. Steps for deploying the project

To deploy the web application, all the configuration about the web server should be configured using the publish menu. Select the project solution name, right click and select publish menu item. The publish web dialog will then open. Configure

the web server name as DCM Server and go to the connection tab to input the publish method as FTP, Server address pointing to the given web DCM server at uhcl.edu, give user login details and finally give the destination URL to open after publishing the web application. The user can click on validate connection button to test if the connection is successful with the given configuration settings.

After all the settings are given a click on the publish button to publish the web application. If the hosting is successful, then it opens the website URL and displays the contents designed using the ASP.NET. If there are any issues with the coding or hosting, then the URL returns an error code. Most of the issues can be easily found using the error code. The developer can also verify if the hosting is successful by going to the FTP path and check manually if it contains all the necessary files under pages folder.

10. Group Home Page

Group home page contains some details of group members like their images, name, presentation id, email id. For each member, there is a link called 'READ MORE' which gives additional information about the member.

This page also contains other links like the project, group diary, conceptualization. If we click on the project, it will redirect us to the project. The link group diary shows us the information of work done by the team members. The conceptualization link tells us about the project and its working. When we click on this link it will download some word document which contains all the information regarding the project.

11. Scope for future enhancement

In future we can implement this web application as mobile application. And we can also implement some additional features like our software would also keep a record of room resources and staff details in our database.

Code:

Login.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Login.aspx.cs"
Inherits="Login" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <title></title>
```

```

    <link href="CSS/bootstrap.css" rel="stylesheet" />
    <style>
body {
    background-image:url('media/SR01.jpg');
background-size: cover;
    }
    </style>

</head>
<body>
    <form id="form1" runat="server">
    <div>
        <div class="container-fluid">
            <div class="row">
                <div class="col-md-4"></div>
                <div class="col-md-4" style="margin-top:100px;">
                    <h2>Study Room Reservation System</h2><br />
                    <div class="form-group">
                        <label>Username</label>
                        <asp:TextBox ID="txtUsername" runat="server"
CssClass="form-control"></asp:TextBox>
                        <asp:RequiredFieldValidator
ID="RequiredFieldValidator1" runat="server" ErrorMessage="Username Required"
ForeColor="Red" ControlToValidate="txtUsername"></asp:RequiredFieldValidator>
                    </div>
                    <div class="form-group">
                        <label>Password</label>
                        <asp:TextBox ID="txtPassword" runat="server"
CssClass="form-control" TextMode="Password"></asp:TextBox>
                        <asp:RequiredFieldValidator
ID="RequiredFieldValidator2" runat="server" ErrorMessage="Password Required"
ForeColor="Red" ControlToValidate="txtPassword"></asp:RequiredFieldValidator>
                    </div>
                    <div class="checkbox">
                        <label><asp:CheckBox ID="chkRemember" runat="server"
CssClass="checkbox"/>Remember me</label>
                    </div>
                    <asp:Button ID="Button1" runat="server" Text="Login"
CssClass="btn btn-primary" OnClick="Validateuser"/>
                    <asp:LinkButton runat="server" id="SomeLinkButton"
href="Register.aspx" CssClass="btn btn-primary">New User Register
Here!</asp:LinkButton>
                    <asp:Label ID="lblError" runat="server" Text=""
ForeColor="Red"></asp:Label>
                </div>
                <div class="col-md-4"></div>
            </div>
        </div>
    </div>
    </form>
</body>
</html>

```

```

Login.aspx.cs using
System;
using
System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq; using
System.Web; using
System.Web.Security; using
System.Web.UI;
using System.Web.UI.WebControls;

public partial class Login : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Validateuser(object sender, EventArgs e)
    {
        int userId = 0;
        //string cs = "Data Source=.\SQLEXPRESS01;Initial
Catalog=SRR;Integrated Security=True";
        string cs = "Data Source=dcm.uhcl.edu;Initial
Catalog=c563318sp02g4;Persist Security Info=True;User
ID=c563318sp02g4;Password=7850477";
        using (SqlConnection con = new SqlConnection(cs))
        {
            using (SqlCommand cmd = new SqlCommand("validate_user", con))
            {
                cmd.CommandType = System.Data.CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@Username", txtUsername.Text);
cmd.Parameters.AddWithValue("@Password", txtPassword.Text);
con.Open();

                userId = Convert.ToInt32(cmd.ExecuteScalar());
            }
            switch (userId)
            {
                case -1:
lblError.Text = "Invalid Account";
                    break;
                case -2:
                    lblError.Text = "Account has not been
activated";
                    break;
                default:

//FormsAuthentication.RedirectFromLoginPage(txtUsername.Text,
chkRemember.Checked);
                    Session["UserName"] = txtUsername.Text;
Response.Redirect("Default.aspx");
break;
            }
        }
    }
}

```

Register.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Register.aspx.cs"
Inherits="Register" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <title></title>
    <link href="CSS/bootstrap.css" rel="stylesheet" />
    <style>
body {
            background-image:url('media/SR01.jpg');
background-repeat:no-repeat;                background-
attachment:fixed;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <div class="container">

                <div class="row" style="margin-top:100px;">

                    <div class="col-md-4"></div>

                    <div class="col-md-4">
                        <h2>User Registration</h2><br />
                        <div class="form-group">
                            <label>Username</label>
                            <asp:TextBox ID="txtUsername" runat="server"
CssClass="form-control"></asp:TextBox>
                            <asp:RequiredFieldValidator
ID="RequiredFieldValidator1" runat="server" ErrorMessage="Username Required"
ForeColor="Red" ControlToValidate="txtUsername"></asp:RequiredFieldValidator>
                        </div>

                        <div class="form-group">
                            <label>Password</label>
                            <asp:TextBox ID="txtPassword" runat="server"
CssClass="form-control" TextMode="Password"></asp:TextBox>
                            <asp:RequiredFieldValidator
ID="RequiredFieldValidator2" runat="server" ErrorMessage="Password Required"
ForeColor="Red" ControlToValidate="txtPassword"></asp:RequiredFieldValidator>
                        </div>

                        <div class="form-group">
                            <label>Confirm Password</label>
                            <asp:TextBox ID="txtConfirmPassword"
runat="server"
CssClass="form-control" TextMode="Password"></asp:TextBox>
                        </div>
                    </div>
                </div>
            </div>
        </form>
    </body>
</html>
```



```

        <asp:CompareValidator ID="CompareValidator1"
runat="server" ErrorMessage="Password do not match"
ControlToCompare="txtPassword" ControlToValidate="txtConfirmPassword"
ForeColor="Red"></asp:CompareValidator>
    </div>

    <div class="form-group">
        <label>Email</label>
        <asp:TextBox ID="txtEmail" runat="server"
CssClass="form-control" TextMode="Email"></asp:TextBox>
        <asp:RequiredFieldValidator
ID="RequiredFieldValidator4" runat="server" ErrorMessage="Email Required"
ForeColor="Red" ControlToValidate="txtEmail"
Display="Dynamic"></asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator
ID="RegularExpressionValidator1" runat="server" ErrorMessage="Invalid Email"
Display="Dynamic" ForeColor="Red" ControlToValidate="txtEmail"
ValidationExpression="\w+([-+.']\w+)*@\w+([-.\w+)*\.\w+([-
.\w+)*"></asp:Regul arExpressionValidator>
    </div>

    <asp:Button ID="Button1" runat="server"
Text="Register"
CssClass="btn btn-primary" OnClick="RegisterUser"/>
    <asp:LinkButton runat="server" id="SomeLinkButton"
href="Login.aspx" CssClass="btn btn-primary">Login Here</asp:LinkButton>
    <asp:Label ID="Label1" runat="server" Text=""
ForeColor="Red"></asp:Label>
    </div>

    <div class="col-md-4"></div>
</div>
</div>
</div>
</form>
</body>
</html>

```

Register.aspx.cs

```

using System;
using
System.Collections.Generic;
using System.Data; using
System.Data.SqlClient; using
System.Linq; using System.Net;
using System.Net.Mail;
using System.Web; using
System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Register : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)

```

```

{
}
protected void RegisterUser(object sender, EventArgs e)
{
    if (Page.IsValid)
    {
        int userId = 0;
        //string cs = "Data Source=.\SQLEXPRESS01;Initial
Catalog=SRR;Integrated Security=True";
        string cs = "Data Source=dcm.uhcl.edu;Initial
Catalog=c563318sp02g4;Persist Security Info=True;User
ID=c563318sp02g4;Password=7850477";
        using (SqlConnection con = new SqlConnection(cs))
        {
            using (SqlCommand cmd = new SqlCommand("insert_user"))
            {
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("@Username", txtUsername.Text);
                cmd.Parameters.AddWithValue("@Password", txtPassword.Text);
                cmd.Parameters.AddWithValue("@Email", txtEmail.Text);
                cmd.Connection = con;
                con.Open();
                userId = Convert.ToInt32(cmd.ExecuteScalar());
            }
        }
        switch (userId)
        {
        case -1:
            Label1.Text = "Username already
exists";
            break;
        case -
2:
            Label1.Text = "Email already
exists";
            break;
        default:
            SendActivationEmail(userId);
        }
        break;
    }
}
private void SendActivationEmail(int userId)
{
    //string cs = "Data Source=.\SQLEXPRESS01;Initial
Catalog=SRR;Integrated Security=True";
    string cs = "Data Source=dcm.uhcl.edu;Initial
Catalog=c563318sp02g4;Persist Security Info=True;User
ID=c563318sp02g4;Password=7850477";
    string activationCode = Guid.NewGuid().ToString();
    using (SqlConnection con = new SqlConnection(cs))
    {
        using (SqlCommand cmd = new SqlCommand("insert into
UserActivation values (@UserId, @ActCode)", con))
        {

```

```

        cmd.Parameters.AddWithValue("@UserId", userId);
cmd.Parameters.AddWithValue("@ActCode", activationCode);
con.Open();

        cmd.ExecuteNonQuery();
    }
}
using (MailMessage mail = new
MailMessage("myEmail@email.com", txtEmail.Text))
{
    mail.Subject = "SRRS - Activate Account";
    string body = "Hello " + txtUsername.Text.Trim() + ", ";
body += "<br/><br/>Welcome to Study Room Reservation System";
body += "<br/><br/><a href='" +
Request.Url.AbsoluteUri.Replace("Register.aspx", "Activation.aspx?uid="
+ activationCode + "'>Click here to activate your account</a>";
body += "<br/><br/>Thank You";          body += "<br/>Team SRRS";
mail.Body = body;          mail.IsBodyHtml = true;
    SmtpClient smtp = new SmtpClient();
smtp.Host = "smtp.gmail.com";
smtp.EnableSsl = true;
    NetworkCredential credentials = new
NetworkCredential("srrssystemuhcl@gmail.com",
"srrs@1234");          smtp.UseDefaultCredentials =
true;          smtp.Credentials = credentials;
smtp.Port = 587;          smtp.Send(mail);

        //Go to Login

Response.Redirect("http://dcm.uhcl.edu/c563318sp02g4/SRRS/Login.aspx");
    }
}
}

```

Default.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" MasterPageFile="./Site.master" Title="Study Room
Reservation System" %>
<%@ Register Assembly="DayPilot" Namespace="DayPilot.Web.Ui"
TagPrefix="DayPilot" %>
<asp:Content ID="HeaderContent" runat="server"
ContentPlaceHolderID="HeadContent">
    <script type="text/javascript" src="js/modal.js"></script>
    <link href='css/main.css' type="text/css" rel="stylesheet" />
    <script type="text/javascript">
var modal = new DayPilot.Modal();
modal.border = "10px solid #ccc";
modal.closed = function () {          if
(this.result == "OK") {
    dps.commandCallBack('refresh');
}
}
    };
    function createEvent(start, end, resource) {
        modal.height = 250;
        modal.showUrl("New.aspx?start=" + start.toStringSortable()
+ "&end=" + end.toStringSortable() + "&r=" + resource);    }

```

```

function editEvent(id) {
    modal.height = 300;
    modal.showUrl("Edit.aspx?id=" + id);
}
function afterRender(data) {
    };
function filter(property, value) {
    if (!dps.clientState.filter) {
        dps.clientState.filter = {};
    }
    if (dps.clientState.filter[property] != value) { // only
        refresh
when the value has changed
        dps.clientState.filter[property] = value;
        dps.commandCallBack('filter');
    }
}

</script>
<style type="text/css">
    .scheduler_default_rowheader .scheduler_default_rowheader_inner
    {
        border-right: 1px solid #aaa;
    }
    .scheduler_default_rowheader.scheduler_default_rowheadercol2
    {
        background: White;
    }
    .scheduler_default_rowheadercol2 .scheduler_default_rowheader_inner
    {
top: 2px;
bottom: 2px;
left: 2px;
        background-color: transparent;
        border-left: 5px solid #1a9d13; /* green */
border-right: 0px none;
    }
    .status_dirty.scheduler_default_rowheadercol2
.scheduler_default_rowheader_inner
    {
        border-left: 5px solid #ea3624; /* red */
    }
    .status_cleanup.scheduler_default_rowheadercol2
.scheduler_default_rowheader_inner
    {
        border-left: 5px solid #f9ba25; /* orange */
    }
</style>
</asp:Content>

<asp:Content ID="BodyContent" runat="server"
ContentPlaceHolderID="MainContent">
    <div style="margin-bottom: 5px;">

```

```

Show rooms:
    <asp:DropDownList ID="DropDownListFilter" runat="server"
onchange="filter('room', this.value)">
    <asp:ListItem Text="All" Value="0"></asp:ListItem>
    <asp:ListItem Text="BAYOU" Value="1"></asp:ListItem>
    <asp:ListItem Text="SSCB" Value="2"></asp:ListItem>
    <asp:ListItem Text="DELTA" Value="3"></asp:ListItem>
    <asp:ListItem Text="ARBOR" Value="4"></asp:ListItem>
    </asp:DropDownList>
</div>
<DayPilot:DayPilotScheduler
ID="DayPilotScheduler1"
    runat="server"

    DataStartField="ReservationStart"
    DataEndField="ReservationEnd"
    DataTextField="ReservationName"
    DataValueField="ReservationId"
    DataResourceField="RoomId"
    DataTagFields="ReservationStatus"

    ClientObjectName="dps"

    CellGroupBy="Month"
    CellDuration="1440"
    Days="365"

    CellWidth = "100"

    HeightSpec="Max"
    Height="550"
    Width="100%"
    HeaderFontSize="8pt"
    EventFontSize="8pt"

    EventMoveHandling="CallBack"
    OnEventMove="DayPilotScheduler1_EventMove"

    EventResizeHandling="CallBack"
    OnEventResize="DayPilotScheduler1_EventResize"

    TimeRangeSelectedHandling="JavaScript"
    TimeRangeSelectedJavaScript="createEvent(start, end, column);"

    OnCommand="DayPilotScheduler1_Command"

    EventClickHandling="JavaScript"
    EventClickJavaScript="editEvent(e.value());"

    AfterRenderJavaScript="afterRender(data);"

    OnBeforeEventRender="DayPilotScheduler1_BeforeEventRender"

```

```

OnBeforeCellRender="DayPilotScheduler1_BeforeCellRender"

    RowHeaderWidthAutoFit="true"
    EventHeight="50"
    DurationBarVisible="true"
    SyncResourceTree="false"

    OnBeforeResHeaderRender="DayPilotScheduler1_BeforeResHeaderRender"
>

<TimeHeaders>
    <DayPilot:TimeHeader GroupBy="Month" Format="MMMM yyyy" />
    <DayPilot:TimeHeader GroupBy="Day" />
</TimeHeaders>
<HeaderColumns>
    <DayPilot:RowHeaderColumn Title="Room" Width="80" />
    <DayPilot:RowHeaderColumn Title="RoomCode" Width="60" />
    <DayPilot:RowHeaderColumn Title="Size" Width="80" />
    <DayPilot:RowHeaderColumn Title="Capacity" Width="80" />
    <DayPilot:RowHeaderColumn Title="Projector" Width="60" />
    <DayPilot:RowHeaderColumn Title="Location" Width="80" />
<DayPilot:RowHeaderColumn Title="Status" Width="60" />
</HeaderColumns>
</DayPilot:DayPilotScheduler>

<br />

```

</asp:Content>

Default.aspx.cs

```

/*
 *
 */ using System; using
System.Configuration;
using System.Data; using
System.Data.SqlClient;
using System.Drawing;
using DayPilot.Web.Ui;
using
DayPilot.Web.Ui.Data;
using
DayPilot.Web.Ui.Enums;
using
DayPilot.Web.Ui.Enums.Sche
duler; using
DayPilot.Web.Ui.Events.Sch
eduler;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // prevent invalid ViewState errors in Firefox
    }
}

```

```

        if (Request.Browser.Browser == "Firefox")
Response.Cache.SetNoStore();
        DayPilotScheduler1.Separators.Clear();
        DayPilotScheduler1.Separators.Add(DateTime.Now, Color.Red);

        if (!IsPostBack)
        {

            //DayPilotScheduler1.StartDate = new
DateTime(DateTime.Today.Year, 1, 1);

            DayPilotScheduler1.Scale = TimeScale.Manual;
            DateTime start = new DateTime(DateTime.Today.Year, 1, 1, 12, 0,
0);
            DateTime end = start.AddYears(1);

            DayPilotScheduler1.Timeline.Clear();
            for (DateTime cell = start; cell < end; cell = cell.AddDays(1))
            {
                DayPilotScheduler1.Timeline.Add(cell, cell.AddDays(1));
            }

            LoadResourcesAndEvents();

            // scroll to this month
            DateTime firstOfMonth = new DateTime(DateTime.Today.Year,
DateTime.Today.Month, 1);
            DayPilotScheduler1.SetScrollX(DateTime.Today);
        }
    }
    protected void DayPilotScheduler1_EventMove(object sender,
DayPilot.Web.Ui.Events.EventMoveEventArgs e)
    {
        string id = e.Value;
        DateTime start = e.NewStart;
        DateTime end = e.NewEnd;          string
        resource = e.NewResource;
        string username = "guest";

        //Check weather session variable null or not
        if (Session["UserName"] != null)
        {
            //Retrieving UserName from Session
            username = (string)(Session["UserName"]);
        }
        else
        {
            //Do Something else
        }
        string message = null;

        if (!dbIsFree(id, start, end, resource))
        {

```

```

        message = "The reservation cannot overlap with an
existing reservation.";          }          else if (e.OldEnd <=
DateTime.Today)
    {
        message = "This reservation cannot be changed anymore.";
    }
    else if (e.OldStart < DateTime.Today)
    {
        if (e.OldResource != e.NewResource)
        {
            message = "The room cannot be changed anymore.";
        }
    }
else
{
    message = "The reservation start cannot be changed anymore.";
}

    else if (e.NewStart < DateTime.Today)
    {
        message = "The reservation cannot be moved to the past.";
    }
else
{
    dbUpdateEvent(id, start, end, resource, username);
    //message = "Reservation moved.";
}

    LoadResourcesAndEvents();
    DayPilotScheduler1.UpdateWithMessage(message);
}
private DataTable dbGetEvents(DateTime start, DateTime end)
{
    //SqlDataAdapter da = new SqlDataAdapter("SELECT [ReservationId],
[ReservationName], [ReservationStart], [ReservationEnd], [RoomId],
[ReservationStatus], [ReservationPaid] FROM [Reservation] WHERE NOT
((([ReservationEnd] <= @start) OR ([ReservationStart] >= @end))",
ConfigurationManager.ConnectionStrings["daypilot"].ConnectionString);
    SqlDataAdapter da = new SqlDataAdapter("SELECT [ReservationId],
[ReservationName], [ReservationStart], [ReservationEnd], [RoomId],
[ReservationStatus] FROM [Reservation] WHERE NOT (([ReservationEnd] <=
@start)
OR ([ReservationStart] >= @end))",
ConfigurationManager.ConnectionStrings["srrs"].ConnectionString);
    da.SelectCommand.Parameters.AddWithValue("start", start);
    da.SelectCommand.Parameters.AddWithValue("end", end);
    DataTable dt = new
DataTable();          da.Fill(dt);
return dt;
}
    private void dbUpdateEvent(string id, DateTime start, DateTime end,
string resource, string username)
    {
        using (SqlConnection con = new

```



```

SqlConnection(ConfigurationManager.ConnectionStrings["srrs"].ConnectionString
))
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("UPDATE [Reservation] SET
ReservationStart = @start, ReservationEnd = @end, RoomId = @resource,
UserName
= @username WHERE ReservationId = @id", con);
cmd.Parameters.AddWithValue("id", id);
cmd.Parameters.AddWithValue("start", start);
cmd.Parameters.AddWithValue("end", end);
cmd.Parameters.AddWithValue("resource", resource);
cmd.Parameters.AddWithValue("username", username);
cmd.ExecuteNonQuery();
    }
}

private bool dbIsFree(string id, DateTime start, DateTime end, string
resource)
{
    // event with the specified id will be ignored
    SqlDataAdapter da = new SqlDataAdapter("SELECT count(ReservationId)
as count FROM [Reservation] WHERE NOT (([ReservationEnd] <= @start) OR
([ReservationStart] >= @end)) AND RoomId = @resource AND ReservationId <>
@id",
ConfigurationManager.ConnectionStrings["srrs"].ConnectionString);
da.SelectCommand.Parameters.AddWithValue("id", id);
da.SelectCommand.Parameters.AddWithValue("start", start);
da.SelectCommand.Parameters.AddWithValue("end", end);
da.SelectCommand.Parameters.AddWithValue("resource", resource);
    DataTable dt = new DataTable();
da.Fill(dt);

    int count = Convert.ToInt32(dt.Rows[0]["count"]);
return count == 0;
}

private void LoadResources()
{
    DayPilotScheduler1.Resources.Clear();

    string roomFilter = "0";
    if (DayPilotScheduler1.ClientState["filter"] != null)
    {
        roomFilter = (string)
DayPilotScheduler1.ClientState["filter"]["room"];
    }

    SqlDataAdapter da = new SqlDataAdapter("SELECT [RoomId], [RoomName],
[RoomStatus], [RoomSize], [Capacity], [Location], [Projector] FROM [Room]
WHERE
RoomType = @rt or @rt = '0' ORDER BY RoomId ASC",
ConfigurationManager.ConnectionStrings["srrs"].ConnectionString);
da.SelectCommand.Parameters.AddWithValue("rt", roomFilter);

```

```

        DataTable dt = new DataTable();
da.Fill(dt);

        foreach (DataRow r in dt.Rows)
        {
            string name = (string)r["RoomName"];
string id = Convert.ToString(r["RoomId"]);
string status = (string)r["RoomStatus"];
int sqft = Convert.ToInt32(r["RoomSize"]);
            string sqftFormatted = (sqft == 1) ? "1 sqft" :
String.Format("{0} sqfts", sqft);
            int capacity = Convert.ToInt32(r["Capacity"]);
string capacityFormatted = (capacity == 1) ? "1 Person" :
String.Format("{0} Persons", capacity);
            string location
= (string)r["Location"];
            string projector =
(string)r["Projector"];

            Resource res = new Resource(name, id);
res.DataItem = r;
            res.Columns.Add(new ResourceColumn(id));
res.Columns.Add(new ResourceColumn(sqftFormatted));
res.Columns.Add(new ResourceColumn(capacityFormatted));
res.Columns.Add(new ResourceColumn(projector));
res.Columns.Add(new ResourceColumn(location));
res.Columns.Add(new ResourceColumn(status));
            DayPilotScheduler1.Resources.Add(res);
        }
    }
    protected void DayPilotScheduler1_Command(object sender,
DayPilot.Web.Ui.Events.CommandEventArgs e)
    {
        switch (e.Command)
        {
            case
"refresh":

LoadResourcesAndEvents();
break;
            case "filter":
                LoadResourcesAndEvents();
break;
        }
    }
    private void LoadResourcesAndEvents()
    {
        LoadResources();
        DayPilotScheduler1.DataSource =
dbGetEvents(DayPilotScheduler1.VisibleStart, DayPilotScheduler1.VisibleEnd);
        DayPilotScheduler1.DataBind();
        DayPilotScheduler1.Update();
    }

    protected void DayPilotScheduler1_EventResize(object sender,
DayPilot.Web.Ui.Events.EventResizeEventArgs e)

```

```

        {
            string id = e.Value;
            DateTime start = e.NewStart;
            DateTime end = e.NewEnd;
            string resource = e.Resource;

            string username = "guest";
            //Check weather session variable null or not
            if (Session["UserName"] != null)
            {
                //Retrieving UserName from Session
                username = (string)(Session["UserName"]);
            }
            else
            {
                //Do Something else
            }
            string message = null;

            if (!dbIsFree(id, start, end, resource))
            {
                message = "The reservation cannot overlap with an
existing reservation.";
            }
            else if (e.OldEnd <= DateTime.Today)
            {
                message = "This reservation cannot be changed anymore.";
            }
            else if (e.OldStart != e.NewStart)
            {
                if (e.OldStart < DateTime.Today)
                {
                    message = "The reservation start cannot be changed anymore.";
                }
                else if (e.NewStart < DateTime.Today)
                {
                    message = "The reservation cannot be moved to the past.";
                }
            }
        }
        else
        {
            dbUpdateEvent(id, start, end, resource, username);
            //message = "Reservation updated.";
        }

        LoadResourcesAndEvents();
        DayPilotScheduler1.UpdateWithMessage(message);
    }
    protected void
DayPilotScheduler1_BeforeEventRender(object sender,
DayPilot.Web.Ui.Events.Scheduler.BeforeEventRenderEventArgs e)
    {
        e.InnerHTML = String.Format("{0} ({1:d} - {2:d})", e.Text,
e.Start, e.End);
        int status =
Convert.ToInt32(e.Tag["ReservationStatus"]);
    }

```

Meeting Room Booking System

```

switch (status)
{
    case 0: // new
        if (e.Start < DateTime.Today.AddDays(2)) // must be confirmed
two day in advance
        {
            e.DurationBarColor = "red";
            e.ToolTip = "Expired (not confirmed in
time)";
        }
        else
        {
            e.DurationBarColor = "orange";
            e.ToolTip =
"New";
        }
        break;
    case 1: //
confirmed
        if (e.Start < DateTime.Today || (e.Start == DateTime.Today &&
DateTime.Now.TimeOfDay.Hours > 18)) // must arrive before 6 pm
        {
            e.DurationBarColor = "#f41616"; // red
            e.ToolTip = "Late
arrival";
        }
        else
        {
            e.DurationBarColor = "green";
            e.ToolTip =
"Confirmed";
        }
        break;
    case 2: // arrived
        if (e.End < DateTime.Today || (e.End == DateTime.Today &&
DateTime.Now.TimeOfDay.Hours > 11)) // must checkout before 10 am
        {
            e.DurationBarColor = "#f41616"; // red
            e.ToolTip = "Late
checkout";
        }
        else
        {
            e.DurationBarColor = "#1691f4"; // blue
            e.ToolTip =
"Arrived";
        }
        break;
    case 3: // checked out
        e.DurationBarColor = "gray";
        e.ToolTip = "Checked out";
        break;
    default:
        throw new
ArgumentException("Unexpected status.");
}

e.InnerHTML = e.InnerHTML + String.Format("<br /><span
style='color:gray'>{0}</span>", e.ToolTip);

int paid = 0; // Convert.ToInt32(e.DataItem["ReservationPaid"]);
string paidColor = "aaaaaa";

//e.Areas.Add(new Area().Bottom(10).Right(4).Html("<div
style='color:"

```

```

+ paidColor + "; font-size: 8pt;'">Paid: " + paid +
"%</div>").Visibility(AreaVisibility.Visible));
    //e.Areas.Add(new
Area().Left(4).Bottom(8).Right(4).Height(2).Html("<div
style='background-color:" + paidColor + "; height: 100%; width:" + paid +
"%'></div>").Visibility(AreaVisibility.Visible));
    }
    protected void DayPilotScheduler1_BeforeCellRender(object sender,
DayPilot.Web.Ui.Events.BeforeCellRenderEventArgs e)
    {
        if (e.IsBusiness)
        {
            e.BackgroundColor =
"#ffffff";        }        else
        {
            e.BackgroundColor = "#f8f8f8";
        }
    }
    protected void DayPilotScheduler1_BeforeResHeaderRender(object sender,
BeforeResHeaderRenderEventArgs e)
    {
        string status = (string)e.DataItem["RoomStatus"];
switch (status)
    {
case "Dirty":
        e.CssClass =
"status_dirty";        break;
case "Cleanup":
        e.CssClass = "status_cleanup";
break;
    }
    }
}

```

Activation.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Activation.aspx.cs"
Inherits="Activation" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:Literal ID="Literal1" runat="server"></asp:Literal>

        </div>
    </form>
</body>

```

</html>

Activation.aspx.cs

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

public partial class Activation : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            //string cs = "Data Source=.\SQLEXPRESS01;Initial
            Catalog=SRR;Integrated Security=True";
            string cs = "Data Source=dcm.uhcl.edu;Initial
            Catalog=c563318sp02g4;Persist Security
            Info=True;User ID=c563318sp02g4;Password=7850477";
            string activationCode =
            !string.IsNullOrEmpty(Request.QueryString["uid"]) ?
            Request.QueryString["uid"]
            : Guid.Empty.ToString();
            using (SqlConnection con = new SqlConnection(cs))
            {
                using (SqlCommand cmd = new SqlCommand("delete from
            UserActivation where ActivationCode=@code", con))
                {
                    cmd.Parameters.AddWithValue("@code", activationCode);
            con.Open();
                    if (cmd.ExecuteNonQuery() == 1)
                    {
                        Literal1.Text = "Activation Successfull";
            Response.Redirect("http://dcm.uhcl.edu/c563318sp02g4/SRRS/Login.aspx");
                    }
                }
            }
            else
            {
                Literal1.Text = "Invalid Activation Code";
            }
        }
    }
}
```

New.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="New.aspx.cs"
Inherits="New" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>New event</title>
    <link href='css/main.css' type="text/css" rel="stylesheet" />
</head>
<body class="dialog">
    <form id="form1" runat="server">
        <div>
            <table border="0" cellspacing="4" cellpadding="0">
                <tr>
                    <td align="right"></td>
                    <td>
                        <div class="header">New Reservation</div>
                    </td>
                </tr>
                <tr>
                    <td align="right">Start:</td>
                    <td><asp:TextBox ID="TextBoxStart"
runat="server"></asp:TextBox></td>
                </tr>
                <tr>
                    <td align="right">Start Time:</td>
                    <td><asp:TextBox ID="TextBoxStartTime"
runat="server"></asp:TextBox></td>
                </tr>
                <tr>
                    <td align="right"><!--<td><select id="StartTime" name="StartTime">
                        <option value=""></option>
                        <option value="8.00">8:00</option>
                        <option value="9.00">9:00</option>
                        <option value="10.00">10:00</option>
                        <option value="11.00">11:00</option>
                        <option value="12.00">12:00</option>
                        <option value="13.00">13:00</option>
                        <option value="14.00">14:00</option>
                        <option value="15.00">15:00</option>
                        <option value="16.00">16:00</option>
                        <option value="17.00">17:00</option>
                        <option value="18.00">18:00</option>
                        <option value="19.00">19:00</option>
                        <option value="20.00">20:00</option>
                    </select></td>-->
                </tr>
                <tr>
                    <td align="right">End:</td>
                    <td><asp:TextBox ID="TextBoxEnd"
runat="server"></asp:TextBox></td>
                </tr>
                <tr>
                    <td align="right">End Time:</td>
                    <td><asp:TextBox ID="TextBoxEndTime"
runat="server"></asp:TextBox></td>
                </tr>
                <tr>
                    <td align="right"><!--<td><select id="EndTime" name="EndTime">
                        <option value=""></option>
                        <option value="8.00">8:00</option>
                        <option value="9.00">9:00</option>
                        <option value="10.00">10:00</option>
                    </select></td>-->
                </tr>
            </table>
        </div>
    </form>
</body>
</html>

```

```

        <option value="11.00">11:00</option>
        <option value="12.00">12:00</option>
        <option value="13.00">13:00</option>
        <option value="14.00">14:00</option>
        <option value="15.00">15:00</option>
        <option value="16.00">16:00</option>
        <option value="17.00">17:00</option>
        <option value="18.00">18:00</option>
        <option value="19.00">19:00</option>
        <option value="20.00">20:00</option>
    </select></td>-->
</tr>
<tr>
    <td align="right">Room:</td>
    <td><asp:DropDownList ID="DropDownList1"
runat="server"></asp:DropDownList></td>
</tr>
<tr>
    <td align="right">Name:</td>
    <td><asp:TextBox ID="TextBoxName"
runat="server"></asp:TextBox></td>
</tr>
<tr>
    <td align="right"></td>
    <td>
        <asp:Button ID="ButtonOK" runat="server"
OnClick="ButtonOK_Click" Text="OK" />
        <asp:Button ID="ButtonCancel" runat="server"
Text="Cancel"
OnClick="ButtonCancel_Click" />
    </td>
</tr>
</table>

</div>
</form>
</body> </html>

```

New.aspx.cs

```

/*
 *
 * WDD Project
 */
using System; using
System.Configuration; using
System.Data; using
System.Data.SqlClient;
using System.Web.UI;

public partial class New : Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {

```



```

        TextBoxStart.Text =
Convert.ToDateTime(Request.QueryString["start"]).ToShortDateString();
        TextBoxEnd.Text =
Convert.ToDateTime(Request.QueryString["end"]).ToShortDateString();
        TextBoxStartTime.Text =
Convert.ToDateTime(Request.QueryString["start"]).ToShortTimeString();
        TextBoxEndTime.Text =
Convert.ToDateTime(Request.QueryString["end"]).ToShortTimeString()
;

        TextBoxName.Focus();

        DropDownList1.DataSource = dbGetResources();
        DropDownList1.DataTextField = "RoomName";
        DropDownList1.DataValueField = "RoomId";
        DropDownList1.SelectedValue = Request.QueryString["r"];
        DropDownList1.DataBind();
    }
}

protected void ButtonOK_Click(object sender, EventArgs e)
{
    DateTime start =
Convert.ToDateTime(TextBoxStart.Text).Date.AddHours(12);
    DateTime end =
Convert.ToDateTime(TextBoxEnd.Text).Date.AddHours(12);
    string sTime = TextBoxStartTime.Text;
    string eTime = TextBoxEndTime.Text;
    string name = TextBoxName.Text;
    string resource = DropDownList1.SelectedValue;

    string username = "guest";
    //Check weather session variable null or not
    if (Session["UserName"] != null)
    {
        //Retrieving UserName from Session
        username = (string)(Session["UserName"]);
    }
    else
    {
        //Do Something else
    }
    dbInsertEvent1(start, end, sTime, eTime, name, resource, username,
0);

    //dbInsertEvent(start, end, name, resource, 0);
    Modal.Close(this, "OK");
}

private DataTable dbGetResources()
{
    SqlDataAdapter da = new SqlDataAdapter("SELECT [RoomId], [RoomName]
FROM [Room]",
ConfigurationManager.ConnectionStrings["srrs"].ConnectionString);
    DataTable dt = new DataTable();
    da.Fill(dt);

    return dt;
}

```

```

        private void dbInsertEvent(DateTime start, DateTime end, string name,
string resource, int status)
        {
            using (SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["srrs"].ConnectionString
))
            {
                con.Open();
                SqlCommand cmd = new SqlCommand("INSERT INTO [Reservation]
(ReservationStart, ReservationEnd, ReservationName, RoomId,
ReservationStatus)
VALUES(@start, @end, @name, @resource, @status)", con);
                //cmd.Parameters.AddWithValue("id", id);
                cmd.Parameters.AddWithValue("start", start);
                cmd.Parameters.AddWithValue("end", end);
                cmd.Parameters.AddWithValue("name", name);
                cmd.Parameters.AddWithValue("resource", resource);
                cmd.Parameters.AddWithValue("status", status);
                cmd.ExecuteNonQuery();
            }
        }

        private void dbInsertEvent1(DateTime start, DateTime end,
string sTime, string eTime, string name, string resource, string
username, int status)
        {
            using (SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["srrs"].ConnectionString
))
            {
                con.Open();
                SqlCommand cmd = new SqlCommand("INSERT INTO [Reservation]
(ReservationStart, ReservationEnd, StartTime, EndTime, ReservationName,
RoomId,
UserName, ReservationStatus) VALUES(@start, @end, @stime, @etime, @name,
@resource, @username, @status)", con);
                //cmd.Parameters.AddWithValue("id", id);
                cmd.Parameters.AddWithValue("start", start);
                cmd.Parameters.AddWithValue("end", end);
                cmd.Parameters.AddWithValue("stime", sTime);
                cmd.Parameters.AddWithValue("etime", eTime);
                cmd.Parameters.AddWithValue("name", name);
                cmd.Parameters.AddWithValue("resource", resource);
                cmd.Parameters.AddWithValue("username", username);
                cmd.Parameters.AddWithValue("status", status);
                cmd.ExecuteNonQuery();
            }
        }

        protected void ButtonCancel_Click(object sender, EventArgs e)
        {
            Modal.Close(this);
        }
    }

```

Edit.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Edit.aspx.cs"
Inherits="Edit" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>New event</title>
    <link href='css/main.css' type="text/css" rel="stylesheet" />
</head>
<body class="dialog">
    <form id="form1" runat="server">
        <div>
            <table border="0" cellspacing="4" cellpadding="0">
                <tr>
                    <td align="right"></td>
                    <td>
                        <div class="header">Edit Reservation</div>
                        <asp:LinkButton ID="LinkButtonDelete" runat="server"
OnClick="LinkButtonDelete_Click">Delete</asp:LinkButton>
                        <br />
                        <br />
                    </td>
                </tr>
                <tr>
                    <td align="right">Start:</td>
                    <td><asp:TextBox ID="TextBoxStart"
runat="server"></asp:TextBox></td>
                    <td align="right">Start Time:</td>
                    <td><asp:TextBox ID="TextBoxStartTime"
runat="server"></asp:TextBox></td>
                </tr>
                <tr>
                    <td align="right">End:</td>
                    <td><asp:TextBox ID="TextBoxEnd"
runat="server"></asp:TextBox></td>
                    <td align="right">End Time:</td>
                    <td><asp:TextBox ID="TextBoxEndTime"
runat="server"></asp:TextBox></td>
                </tr>
                <tr>
                    <td align="right">Resource:</td>
                    <td><asp:DropDownList ID="DropDownListRoom" runat="server"
Width="200px"></asp:DropDownList></td>
                </tr>
                <tr>
                    <td align="right">Name:</td>
                    <td><asp:TextBox ID="TextBoxName" runat="server"
Width="200px"></asp:TextBox></td>
                </tr>
                <tr>
                    <td align="right">Reserved By:</td>

```

```

                <td><asp:TextBox ID="TextBoxUser" runat="server"
Width="200px"
ReadOnly="true"></asp:TextBox></td>
            </tr>
            <tr>
                <td align="right">
                    Status:</td>
                <td>
                    <asp:DropDownList ID="DropDownListStatus" runat="server">
                        <asp:ListItem Value="0">New</asp:ListItem>
                        <asp:ListItem Value="1">Confirmed</asp:ListItem>
                        <asp:ListItem Value="2">Arrived</asp:ListItem>
                        <asp:ListItem Value="3">Checked out</asp:ListItem>
                    </asp:DropDownList></td>
            </tr>
            <!--
            <tr>
                <td align="right">
                    Paid:</td>
                <td>
                    <asp:DropDownList ID="DropDownListPaid" runat="server">
                        <asp:ListItem Value="0">0%</asp:ListItem>
                        <asp:ListItem Value="50">50%</asp:ListItem>
<asp:ListItem Value="100">100%</asp:ListItem>
                    </asp:DropDownList></td>
            </tr>
            -->

            <tr>
                <td align="right"></td>
                <td>
                    <asp:Button ID="ButtonOK" runat="server"
OnClick="ButtonOK_Click" Text=" OK " />
                    <asp:Button ID="ButtonCancel" runat="server"
Text="Cancel"
OnClick="ButtonCancel_Click" />
                </td>
            </tr>
        </table>

    </div>
</form>
</body>
</html>

```

```

Edit.aspx.cs using System;
using System.Configuration;
using System.Data; using
System.Data.SqlClient;
using System.Web; using
System.Web.UI;

```

```

public partial class Edit : Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Response.Cache.SetCacheability(HttpCacheability.NoCache);

        if (!IsPostBack)
        {
            DataRow dr = dbGetEvent(Request.QueryString["id"]);

            if (dr == null)
            {
                throw new Exception("The event was not found");
            }

            TextBoxStart.Text =
Convert.ToDateTime(dr["ReservationStart"]).ToShortDateString();
            TextBoxEnd.Text =
Convert.ToDateTime(dr["ReservationEnd"]).ToShortDateString();
            TextBoxStartTime.Text = dr["StartTime"] as string;
            TextBoxEndTime.Text = dr["EndTime"] as string;
            TextBoxName.Text = dr["ReservationName"] as string;
            TextBoxUser.Text = dr["UserName"] as string;

            DropDownListRoom.DataSource = dbGetResources();
            DropDownListRoom.DataTextField = "RoomName";
            DropDownListRoom.DataValueField = "RoomId";
            DropDownListRoom.SelectedValue = Convert.ToString(dr["RoomId"]);
            DropDownListRoom.DataBind();

            DropDownListStatus.SelectedValue =
Convert.ToString(dr["ReservationStatus"]);
            DropDownListPaid.SelectedValue =
Convert.ToString(dr["ReservationPaid"]);
            TextBoxName.Focus();
        }
    }

    protected void ButtonOK_Click(object sender, EventArgs e)
    {
        DateTime start =
Convert.ToDateTime(TextBoxStart.Text).Date.AddHours(12);
        DateTime end =
Convert.ToDateTime(TextBoxEnd.Text).Date.AddHours(12);          string sTime
= TextBoxStartTime.Text;          string eTime = TextBoxEndTime.Text;
        string name = TextBoxName.Text;
        string resource = DropDownListRoom.SelectedValue;
        string id = Request.QueryString["id"];
        int status = Convert.ToInt32(DropDownListStatus.SelectedValue);
        int paid = Convert.ToInt32(DropDownListPaid.SelectedValue);

        string username = "guest";
        //Check weather session variable null or not
        if (Session["UserName"] != null)
    
```

```

        {
            //Retrieving UserName from Session
            username = (string) (Session["UserName"]);
        }
    else
    {
        //Do Something else
    }
    dbUpdateEvent1(id, start, end, sTime, eTime, name, resource, status,
    paid, username);
    //dbUpdateEvent(id, start, end, name, resource, status, paid, );
    Modal.Close(this, "OK");
}
private DataTable dbGetResources()
{
    SqlDataAdapter da = new SqlDataAdapter("SELECT [RoomId], [RoomName]
FROM [Room]",
ConfigurationManager.ConnectionStrings["srrs"].ConnectionString);
    DataTable dt = new DataTable();
    da.Fill(dt);

    return dt;
}
private DataRow dbGetEvent(string id)
{
    SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM [Reservation]
WHERE ReservationId = @id",
ConfigurationManager.ConnectionStrings["srrs"].ConnectionString);
    da.SelectCommand.Parameters.AddWithValue("id", id);
    DataTable dt = new DataTable();
    da.Fill(dt);

    if (dt.Rows.Count > 0)
    {
        return dt.Rows[0];
    }
    return null;
}
private void dbUpdateEvent(string id, DateTime start, DateTime end, string
name, string resource, int status, int paid)
{
    using (SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["srrs"].ConnectionString
))
    {
        con.Open();
        //SqlCommand cmd = new SqlCommand("UPDATE [Reservation] SET
ReservationStart = @start, ReservationEnd = @end, ReservationName = @name,
RoomId = @resource, ReservationStatus = @status, ReservationPaid = @paid
WHERE
ReservationId = @id", con);
        SqlCommand cmd = new SqlCommand("UPDATE [Reservation] SET

```

```

ReservationStart = @start, ReservationEnd = @end, ReservationName =
@name, RoomId = @resource, ReservationStatus = @status WHERE
ReservationId = @id", con);          cmd.Parameters.AddWithValue("id",
id);          cmd.Parameters.AddWithValue("start", start);
cmd.Parameters.AddWithValue("end", end);
cmd.Parameters.AddWithValue("name", name);
cmd.Parameters.AddWithValue("resource", resource);
cmd.Parameters.AddWithValue("status", status);
cmd.Parameters.AddWithValue("paid", paid);
cmd.ExecuteNonQuery();
    }
}
private void dbUpdateEvent1(string id, DateTime start, DateTime
end, string sTime, string eTime, string name, string resource, int status,
int paid, string username)
{
    using (SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["srrs"].ConnectionString
))
    {
        con.Open();
        //SqlCommand cmd = new SqlCommand("UPDATE [Reservation] SET
ReservationStart = @start, ReservationEnd = @end, ReservationName = @name,
RoomId = @resource, ReservationStatus = @status, ReservationPaid = @paid
WHERE
ReservationId = @id", con);
        SqlCommand cmd = new SqlCommand("UPDATE [Reservation] SET
ReservationStart = @start, ReservationEnd = @end, StartTime = @stime, EndTime
=
@etime, ReservationName = @name, RoomId = @resource, ReservationStatus =
@status, UserName=@username WHERE ReservationId = @id", con);
        cmd.Parameters.AddWithValue("id", id);
        cmd.Parameters.AddWithValue("start", start);
        cmd.Parameters.AddWithValue("end", end);
        cmd.Parameters.AddWithValue("name", name);
        cmd.Parameters.AddWithValue("stime", sTime);
        cmd.Parameters.AddWithValue("etime", eTime);
        cmd.Parameters.AddWithValue("resource", resource);
        cmd.Parameters.AddWithValue("status", status);
        cmd.Parameters.AddWithValue("paid", paid);
        cmd.Parameters.AddWithValue("username", username);
        cmd.ExecuteNonQuery();
    }
}

private void dbDeleteEvent(string id)
{
    using (SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["srrs"].ConnectionString
))
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("DELETE FROM [Reservation] WHERE
ReservationId = @id", con);

```

```

        cmd.Parameters.AddWithValue("id", id);
cmd.ExecuteNonQuery();
    }
}
protected void ButtonCancel_Click(object sender, EventArgs e)
{
    Modal.Close(this);
}
protected void LinkButtonDelete_Click(object sender, EventArgs e)
{
    string id = Request.QueryString["id"];
    dbDeleteEvent(id);
Modal.Close(this, "OK");
}
}

```

Site.Master

```

<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Site.master.cs"
Inherits="SiteMaster" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head runat="server">
    <title></title>
    <link href="~/Media/layout.css" rel="stylesheet" type="text/css" />
    <asp:ContentPlaceholder ID="HeadContent"
runat="server"></asp:ContentPlaceholder>
</head>
<body>
    <form runat="server">
        <div class="page">
            <div id="header">
                <div class="bg-help">
                    <div class="inBox">
                        <h1 id="logo"><a href=''>Study Room Reservation
System</a></h1>
                        <p id="claim"><a
href="http://dcm.uhcl.edu/c563318sp02g4/">WDD SP18.2 Group 04</a> | UHCL |
<b><a href="http://dcm.uhcl.edu/c563318sp02g4/SRRSMaster">Logout</a></b></p>
                        <hr class="hidden" />
                    </div>
                </div>
            </div>
            <div class="shadow"></div>
            <div class="main">
                <asp:ContentPlaceholder ID="MainContent" runat="server"/>
            </div>
            <div class="clear">
            </div>
        </div>
        <div class="footer">

```



```

    </div>
  </form>
</body>
</html>

```

Web.config

```

<?xml version="1.0"?>
<configuration>
  <connectionStrings>
    <!--<add name="daypilot" connectionString="data
source=.\\SQLEXPRESS01;Integrated
Security=SSPI;AttachDBFilename=|DataDirectory|\\daypilot.mdf;User
Instance=true" providerName="System.Data.SqlClient" />-->    <!--<add
name="daypilot" connectionString="data
Source=.\\SQLEXPRESS01;AttachDbFilename=|DataDirectory|\\daypilot.mdf;Integrate
d Security=True;User Instance=False" providerName="System.Data.SqlClient" />-
->

    <!--
    <add name="daypilot"
    providerName="System.Data.SqlClient"
    connectionString="Data Source=.\\SQLEXPRESS01;Initial
Catalog=SRR;Integrated Security=True;MultipleActiveResultSets=True"/>
    -->

    <add name="daypilot"
      connectionString="Data Source=dcm.uhcl.edu;Initial
Catalog=c563318sp02g4;Persist Security Info=True;User
ID=c563318sp02g4;Password=7850477"
      providerName="System.Data.SqlClient"
    />

    <add name="srrs"
      connectionString="Data Source=dcm.uhcl.edu;Initial
Catalog=c563318sp02g4;Persist Security Info=True;User
ID=c563318sp02g4;Password=7850477"
      providerName="System.Data.SqlClient"
    />
  </connectionStrings>

  <system.web>
    <compilation debug="true" targetFramework="4.0" />
    <globalization uiCulture="en" culture="en-us" />
    <pages clientIDMode="AutoID" />
    <customErrors mode="Off"/>
  </system.web>
</configuration>

```