Submitted by:

Arshdeep Kaur (A01)

200465607

Iteration 3 Plan

**Phase: Construction Iteration:3 Start Date: Feb 24th , 2020 End Date: March 2nd , 2020**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Core Workflows/ Disciplines** | **Use Case/ Requirements** | **Activity/ Task** | **Time**  **(hours)** | **Actual Time**  **(hours)** | **Resources(role)** |
| Requirement | ViewsPlayerHandicap  RecordsPlayerScores | ViewsPlayerHandicap  -Primary scenario(Player)  Write Scenario  ViewsPlayerHandicap  -SecondaryScenario (Clerk)  Write Scenario  RecordsPlayerScores  -Primary scenario(Player)  Write Scenario | 4  4  3 | 5  4  8 | 1 Use case Specifier |
| Analysis | ViewsPlayerHandicap  RecordsPlayerScores | ViewsPlayerHandicap  -Primary scenario(Player)  Domain Model  ViewsPlayerHandicap  -Secondary Scenario (Clerk)  Domain Model  RecordsPlayerScores  -Primary scenario(Player)  Domain Model | 2  2  2 | 4  3  5 | 1 System analyst |
| Design | ViewsPlayerHandicap  RecordsPlayerScores | ViewsPlayerHandicap  -Primary scenario(Player)  Interaction Diagram, Design Class diagram  ViewsPlayerHandicap  -Secondary scenario (Clerk)  Interaction Diagram, Design Class diagram  RecordsPlayerScores  -Primary scenario(Player)  Interaction Diagram, Design Class diagram | 4  4  4 | 5  4  6 | 1 Use case Engineer |
| Implement | ViewsPlayerHandicap  RecordsPlayerScores | ViewsPlayerHandicap  -Primary scenario(Member)  Implementing Object Oriented Design  ViewsPlayerHandicap  -Primary scenario(MembershipCommittee)  Implementing Object Oriented Design  RecordsPlayerScores  -Primary scenarioPlayer)  Implementing Object Oriented Design | 6  6  6 | 9  6  12 | 1 Component Engineer |
| Test | ViewsPlayerHandicap  RecordsPlayerScores | ViewsPlayerHandicap  -Primary scenario(Player)   * Testcase * Test Procedure * Execution of test   ViewsPlayerHandicap  -Primary scenario (Clerk)   * Testcase * Test Procedure * Execution of test   RecordsPlayerScores  -Primary scenario (Player)   * Testcase * Test Procedure * Execution of test | 3  3  2 | 3  3  3 | 1 Test Engineer |
|  |  | **TOTAL:-** | 55 | 80 | 1Use Case Specifiers  1 System Analysts  1 Use Case Engineers  1Component Engineers  1 Test Engineers |

* Project Schedule

Phase: Construction

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Iteration | Minor Milestone | Start Date | End Date | Time (hours) | Resources (Roles) |
| 1 | Tee Times | 1/13/2020 | 1/24/2020 | 90 | 1 Use Case Specifiers  1 System Analysts  1 Use Case Engineers  1Component Engineers  1TestEngineers |
| 2 | Memberships | 1/25/2020 | 02/13/2020 | 70 | 1 Use Case Specifiers  1 System Analysts  1 Use Case Engineers  1Component Engineers  1TestEngineers |
| 3 | Player Scores | 02/14/2020 | 02/28/2020 | 80 | 1 Use Case Specifiers  1 System Analysts  1 Use Case Engineers  1Component Engineers  1TestEngineers |
|  |  |  | Total | 240 | 1 Use Case Specifiers  1 System Analysts  1 Use Case Engineers  1 Component Engineers  1 TestEngineers |

* Phase Plan

Project Schedule

Phase: Transition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Iteration | Minor Milestone | Start Date | End Date | Time (hours) | Resources (Roles) |
| 1 | Beta Release | 03/09/2020 | 1/24/2020 | 25 | 1 Componenet Enginner  1 Test Engineers  1 Integrators  1 System Tester |
| 2 | Final Release | 1/25/2020 | 04/03/2020 | 30 | 1 Componenet Enginner  1 Test Engineers  1 Integrators  1 System Tester |
|  |  |  | Total | 55 | 1 Componenet Enginner  1 Test Engineers  1 Integrators  1 System Tester |

All Code:

Database

create database ClubBaistGCMS

use ClubBaistGCMS

CREATE TABLE DailyTeeSheet(

Date date NOT NULL PRIMARY KEY)

create table Player(

PlayerFirstName varchar(10) Not Null ,

PlayerLastName varchar(10) Not null ,

PhoneNumber varchar(10) Not Null,

primary key(PlayerFirstName, PlayerLastName))

cREATE TABLE TeeTime(

Date date Not Null ,

Time time Not Null,

NumberOfCarts int ,

NumberOfPlayers int Not Null,

PlayerFirstName varchar(10) Not Null,

PlayerLastName varchar(10) Not Null,

Phone varchar(15),

Primary key(Date,Time)

)

alter table TeeTime add constraint FK\_Date

Foreign Key(Date)

references DailyTeeSheet(Date)

alter table TeeTime add constraint FK1

Foreign Key(PlayerFirstName,PlayerLastName)

references Player(PlayerFirstName,PlayerLastName)

create table StandingTeeTimeRequest(

MemberNumber int Not Null,

MemberFirstName varchar(10) Not Null,

MemberLastName varchar(10) Not Null,

DayOfWeek int check (DayOfWeek > 0 and DayOfWeek <8),

RequestedTeeTime time NOT NULL,

RequestedStartDate date NOT NULL,

RequestedEndDate date NOT NULL)

Alter table StandingTeeTimeRequest Add Constraint pk\_Standing Primary Key (MemberNumber)

create procedure AddTeeTime

(@Date date = NULL,

@Time time = NULL,

@NumberOfCarts int = NULL,

@NumberOfPlayers int = NULL,

@PlayerFirstName varchar(10) = NULL,

@PlayerLastName varchar(10) = NULL,

@Phone varchar(10) = NULL

)

AS

DECLARE @ReturnCode iNT

SET @ReturnCode = 1

If @Date = NULL

RAISERROR('Procedure AddTeeTime: Add Date Parameter',16,1)

ELSE IF @Time = NULL

RAISERROR('Procedure AddTeeTime: Add Time Parameter',16,1)

ELSE IF @NumberOfPlayers = NULL

RAISERROR('Procedure AddTeeTime: Add NumberOfPlayers Parameter',16,1)

ELSE IF @PlayerFirstName = NULL

RAISERROR('Procedure AddTeeTime: Add PlayerFirstName Parameter',16,1)

ELSE IF @PlayerLastName = NULL

RAISERROR('Procedure AddTeeTime: Add PlayerLastName Parameter',16,1)

Else

BEGIN

INSERT INTO TeeTime

(Date,Time,NumberOfCarts,NumberOfPlayers,PlayerFirstName,PlayerLastName,Phone)

VALUES(@Date,@Time,@NumberOfCarts,@NumberOfPlayers,@PlayerFirstName,@PlayerLastName,@Phone)

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN INSERT',16,1)

END

RETURN @ReturnCode

create procedure GetTeeSheet(@Date date = NULL)

AS

DECLARE @ReturnCode Int

set @ReturnCode = 1

IF @Date = NULL

RAISERROR('Procedure: GetTeeSheet- Add Date Parameter',16,1)

else

BEGIN

SELECT \* FROM TeeTime WHERE Date = @Date

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN INSERT',16,1)

END

RETURN @ReturnCode

create procedure GetTeeTime(

@Date date = NULL,

@Time time = NULL)

AS

DECLARE @ReturnCode Int

set @ReturnCode = 1

IF @Date = NULL

RAISERROR('Procedure: GetTeeTime:- Add Date Parameter',16,1)

ELSE IF @Time = NULL

RAISERROR('Procedure GetTeeTime: Add Time Parameter',16,1)

else

BEGIN

SELECT \* FROM TeeTime WHERE Date = @Date and Time=@Time

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN INSERT',16,1)

END

RETURN @ReturnCode

create procedure UpdateTeeTime

(@Date date = NULL,

@Time time = NULL,

@PlayerFirstName varchar(10) = NULL,

@PlayerLastName varchar(10) = NULL,

@Phone varchar(10) = NULL

)

AS

DECLARE @ReturnCode iNT

SET @ReturnCode = 1

If @Date = NULL

RAISERROR('Procedure ModifyTeeTime: Add Date Parameter',16,1)

ELSE IF @Time = NULL

RAISERROR('Procedure ModifyTeeTime: Add Time Parameter',16,1)

ELSE IF @PlayerFirstName = NULL

RAISERROR('Procedure ModifyTeeTime: Add PlayerFirstName Parameter',16,1)

ELSE IF @PlayerLastName= NULL

RAISERROR('Procedure ModifyTeeTime: Add Time Parameter',16,1)

ELSE IF @Phone = NULL

RAISERROR('Procedure ModifyTeeTime:Add Phone Parameter',16,1)

Else

BEGIN

Update TeeTime set

PlayerFirstName=@PlayerFirstName,

PlayerLastName=@PlayerLastName,

Phone=@Phone

where Date=@Date and

Time=@Time

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN INSERT',16,1)

END

RETURN @ReturnCode

create procedure AddStandingTeeTime

(@MemberNumber int = NULL,@MemberFirstName varchar(10) = NULL, @MemberLastName varchar(10) = NULL,@DayOfWeek int = NULL,

@RequestedTeeTime time = NULL, @RequestedStartDate date = NULL, @RequestedEndDate date = NULL)

AS

DECLARE @ReturnCode int

SET @ReturnCode = 1

IF @MemberNumber = null

RAISERROR('Procedure: AddStandingTeeTime- Add Parameter MemberNumber',16,1)

ELSE IF @MemberFirstName = null

RAISERROR('Procedure: AddStandingTeeTime- Add Parameter MemberFirstName',16,1)

ELSE IF @MemberLastName = null

RAISERROR('Procedure: AddStandingTeeTime- Add Parameter MemberLastName',16,1)

ELSE IF @RequestedTeeTime = null

RAISERROR('Procedure: AddStandingTeeTime- Add Parameter RequestedTeeTime',16,1)

ELSE IF @RequestedStartDate = null

RAISERROR('Procedure: AddStandingTeeTime- Add Parameter RequestedStartDate',16,1)

ELSE IF @RequestedEndDate = null

RAISERROR('Procedure: AddStandingTeeTime- Add Parameter RequestedEndDate',16,1)

ELSE

BEGIN

INSERT INTO StandingTeeTimeRequest

(MemberNumber,MemberFirstName,MemberLastName,DayOfWeek,RequestedTeeTime,RequestedStartDate,RequestedEndDate)

VALUES

(@MemberNumber,@MemberFirstName,@MemberLastName,@DayOfWeek,@RequestedTeeTime,@RequestedStartDate,@RequestedEndDate)

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN INSERT',16,1)

END

RETURN @ReturnCode

create procedure GetStandingTeeTime(@RequestedStartDate date = NULL)

AS

DECLARE @ReturnCode Int

set @ReturnCode = 1

IF @RequestedStartDate = NULL

RAISERROR('Procedure: GetStandingTeeTime- Add RequestedStartDate Parameter',16,1)

else

BEGIN

SELECT \* FROM StandingTeeTimeRequest WHERE RequestedStartDate = @RequestedStartDate

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN SELECT',16,1)

END

RETURN @ReturnCode

create procedure SearchStandingTeeTime(@MemberNumber int = NULL)

AS

DECLARE @ReturnCode Int

set @ReturnCode = 1

IF @MemberNumber = NULL

RAISERROR('Procedure: SearchStandingTeeTime- Add MemberNumber Parameter',16,1)

else

BEGIN

SELECT \* FROM StandingTeeTimeRequest WHERE MemberNumber = @MemberNumber

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN SELECT',16,1)

END

RETURN @ReturnCode

create procedure DeleteStandingTeeTimeRequest(@MemberNumber int = NULL)

AS

DECLARE @ReturnCode Int

set @ReturnCode = 1

IF @MemberNumber = NULL

RAISERROR('Procedure: DeleteStandingTeeTimeRequest- Add MemberNumber Parameter',16,1)

else

BEGIN

Delete FROM StandingTeeTimeRequest WHERE MemberNumber = @MemberNumber

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN SELECT',16,1)

END

RETURN @ReturnCode

create procedure InsertPlayer(@PlayerFirstName varchar(10) = Null,

@PlayerLastName varchar(10)= Null,

@PhoneNumber varchar(10) = Null)

as

DECLARE @ReturnCode Int

SET @ReturnCode = 1

IF @PlayerFirstName = null

RAISERROR('Procedure: Insert Player, Parameter PlayerFirstName',16,1)

ELSE IF @PlayerLastName = null

RAISERROR('Procedure: Insert Player, Parameter PlayerFirstName',16,1)

ELSE IF @PhoneNumber = NULL

RAISERROR('Procedure: Insert Player, Parameter Phone',16,1)

else

BEGIN

INSERT INTO Player(PlayerFirstName,PlayerLastName,PhoneNumber) values(@PlayerFirstName,@PlayerLastName,@PhoneNumber)

IF @@ERROR = 0

SET @ReturnCode = 0

else

RAISERROR('Error in Insert',16,1)

END

Return @ReturnCode

create procedure InsertDate(@Date date = NULL)

AS

DECLARE @ReturnCode Int

SET @ReturnCode = 1

if @Date = Null

RAISERROR('Procedure:Insert Date, Parameter: Date Required',16,1)

ELSE

BEGIN

Insert into DailyTeeSheet(Date) values(@Date)

IF @@Error = 0

SET @ReturnCode = 0

ELSE

RAISERROR('Error in Insert, Procedure:Insert Date',16,1)

end

Return @ReturnCode

Create table MembershipApplication(

MemberApplicationNumber int IDENTITY(1,1) Primary key,

FirstName varchar(15),

LastName varchar(15),

Address varchar(20),

PostalCode varchar(6),

Phone varchar(10),

Email varchar(20),

DateOfBirth date,

Occupation varchar(15),

CompanyName varchar(15),

CompanyAddress varchar(15),

CompanyPostalCode varchar(6),

CompanyPhone varchar(10),

ShareholderName varchar(30),

Date date,

Status varchar(15) CHECK(Status='Accepted' or Status='Denied' or Status='On-hold' or Status='Waitlisted'))

create procedure GetMembershipApplications

(@Status varchar(15)=NULL)

AS

DECLARE @ReturnCode Int

SET @ReturnCode = 1

IF @STATUS =NULL

RAISERROR('Procedure GetMembershipApplications: Parameter: Status is missing',16,1)

else

BEGIN

SELECT \* FROM MembershipApplication where Status=@Status

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN SELECT',16,1)

END

RETURN @ReturnCode

create procedure RecordsMembershipApplication

(@FirstName varchar(15) = NULL,

@LastName varchar(15) = NULL,

@Address varchar(20) = NULL,

@PostalCode varchar(6) = NULL,

@Phone varchar(10) = NULL,

@Email varchar(20) = NULL,

@DateOfBirth date = NULL,

@Occupation varchar(15) = NULL,

@CompanyName varchar(10) = NULL,

@CompanyAddress varchar(20) = NULL,

@CompanyPostalCode varchar(6) = NULL,

@CompanyPhone varchar(10) = NULL,

@ShareholderName varchar(30) = NULL,

@Date date = NULL,

@Status varchar(15) = NULL)

AS

DECLARE @ReturnCode int

SET @ReturnCode = 1

IF @FirstName = null

RAISERROR('Procedure: RecordsMembershipApplication- Add Parameter MemberFirstName',16,1)

ELSE IF @LastName = null

RAISERROR('Procedure: RecordsMembershipApplication- Add Parameter MemberLastName',16,1)

ELSE IF @Status = null

RAISERROR('Procedure: RecordsMembershipApplication- Add Parameter Status',16,1)

ELSE

BEGIN

INSERT INTO MembershipApplication

(FirstName,LastName,Address, PostalCode, Phone, Email, DateOfBirth,Occupation,CompanyName,CompanyAddress,CompanyPostalCode,CompanyPhone,ShareholderName,Date,Status)

VALUES

(@FirstName,@LastName,@Address, @PostalCode, @Phone, @Email, @DateOfBirth,@Occupation,@CompanyName,@CompanyAddress,@CompanyPostalCode,@CompanyPhone,@ShareholderName,@Date,@Status)

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN INSERT',16,1)

END

RETURN @ReturnCode

create procedure UpdateMembershipApplication

(@MemberApplicationNumber int =null,

@FirstName varchar(15)=NULL,

@LastName varchar(15)=NULL,

@Address varchar(20)=NULL,

@PostalCode varchar(6)=NULL,

@Phone varchar(10)=NULL,

@Email varchar(20)=null,

@Status varchar(15) = NULL)

AS

DECLARE @ReturnCode int

SET @ReturnCode = 1

BEGIN

Update MembershipApplication

set

FirstName=@FirstName,

LastName=@LastName,

Address=@Address,

PostalCode=@PostalCode,

Phone=@Phone,

Email=@Email,

Status=@Status

where MemberApplicationNumber=@MemberApplicationNumber

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN Update',16,1)

END

RETURN @ReturnCode

Create table Member(

MemberNumber int NOT NULL Primary Key,

FirstName varchar(15),

LastName varchar(15),

Address varchar(20),

PostalCode varchar(6),

Phone varchar(10),

Email varchar(20),

MemberType varchar(15) CHECK(MemberType='Shareholder' or MemberType='Non-Shareholder'))

Create procedure AddMember(

@MemberNumber int = NULL,

@FirstName varchar(15) = NULL,

@LastName varchar(15) =NULL,

@Address varchar(20) = NULL,

@PostalCode varchar(6) = NULL,

@Phone varchar(10) = NULL,

@Email varchar(20) = NULL,

@MemberType varchar(15) = NULL)

AS

DECLARE @ReturnCode Int

SET @ReturnCode=1

IF @FirstName = null

RAISERROR('Procedure: AddMember- Add Parameter FirstName',16,1)

ELSE IF @LastName = null

RAISERROR('Procedure: AddMember- Add Parameter LastName',16,1)

ELSE IF @MemberType = null

RAISERROR('Procedure: AddMember- Add Parameter MemberType',16,1)

else

Begin

Insert into Member(MemberNumber,FirstName,LastName,Address,PostalCode,Phone,Email,MemberType)

VALUES(@MemberNumber,@FirstName,@LastName,@Address,@PostalCode,@Phone,@Email,@MemberType)

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN INSERT',16,1)

END

RETURN @ReturnCode

create table MemberAccount(

MemberAccountNumber int NOT NULL Primary Key,

Balance money )

Create procedure AddMemberAccount(

@MemberAccountNumber int =NULL,

@Balance Money = NULL,

@Amount money = null,

@ActivityDate date=NULL,

@PostedDate date = NULL,

@Description varchar(20) =NULL

)

AS

DECLARE @ReturnCode Int

SET @ReturnCode=1

Begin

Insert into MemberAccount(MemberAccountNumber,Balance)

VALUES(@MemberAccountNumber,@Balance)

Insert into MemberAccountEntry(MemberAccountNumber,Amount,ActivityDate,PostedDate,Description)

Values(@MemberAccountNumber,@Amount,@ActivityDate,@PostedDate,@Description)

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN INSERT',16,1)

END

RETURN @ReturnCode

Create table MemberAccountEntry(

MemberAccountNumber int NOT NULL,

Amount money,

ActivityDate datetime,

PostedDate datetime,

Description varchar(20))

Create procedure GetMemberAccount(

@MemberAccountNumber int =NULL

)

AS

DECLARE @ReturnCode Int

SET @ReturnCode=1

Begin

Select Balance from MemberAccount where MemberAccountNumber=@MemberAccountNumber

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN INSERT',16,1)

END

RETURN @ReturnCode

Create procedure GetMemberAccountEntry(

@MemberAccountNumber int =NULL

)

AS

DECLARE @ReturnCode Int

SET @ReturnCode=1

Begin

Select Amount,ActivityDate,PostedDate,Description from MemberAccountEntry where MemberAccountNumber=@MemberAccountNumber

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN INSERT',16,1)

END

RETURN @ReturnCode

create table PlayerScores(

MemberNumber int,

Course varchar(15) ,

Rating decimal,

Slope int,

Date date,

Score int,

Hole1 int,Hole2 int,Hole3 int,

Hole4 int,Hole5 int,Hole6 int,

Hole7 int,Hole8 int,Hole9 int,

Hole10 int,Hole11 int,Hole12 int,

Hole13 int,Hole14 int,Hole15 int,

Hole16 int,Hole17 int,Hole18 int,

HandicapDifferential decimal)

create procedure AddPlayerScore(

@MemberNumber int=NULL,

@Course varchar(15) = NULL,

@Rating decimal =NULL,

@Slope int =NULL,

@Date date =NULL,

@Score int=NULL,

@Hole1 int=NULL,@Hole2 int=NULL,@Hole3 int=NULL,

@Hole4 int=NULL,@Hole5 int=NULL,@Hole6 int=NULL,

@Hole7 int=NULL,@Hole8 int=NULL,@Hole9 int=NULL,

@Hole10 int=NULL,@Hole11 int=NULL,@Hole12 int=NULL,

@Hole13 int=NULL,@Hole14 int=NULL,@Hole15 int=NULL,

@Hole16 int=NULL,@Hole17 int=NULL,@Hole18 int=NULL,

@HandicapDifferential decimal=NULL)

AS

DECLARE @ReturnCode Int

SET @ReturnCode=1

Begin

insert into PlayerScores(

MemberNumber,

Course, Rating, Slope, Date,

Score, Hole1,Hole2,Hole3,

Hole4,Hole5,Hole6,

Hole7,Hole8,Hole9,

Hole10,Hole11,Hole12,

Hole13,Hole14,Hole15,

Hole16,Hole17,Hole18,HandicapDifferential)

values(@MemberNumber,@Course, @Rating, @Slope, @Date, @Score,

@Hole1,@Hole2,@Hole3,

@Hole4,@Hole5,@Hole6,

@Hole7,@Hole8,@Hole9,

@Hole10,@Hole11,@Hole12,

@Hole13,@Hole14,@Hole15,

@Hole16,@Hole17,@Hole18,@HandicapDifferential)

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN INSERT',16,1)

END

RETURN @ReturnCode

Create procedure GetHandicapFactor(@MemberNumber int= null)

AS

DECLARE @ReturnCode Int

SET @ReturnCode=1

Begin

select 0.96\*(SELECT AVG(HandicapDifferential) from( select TOP 3 HandicapDifferential FROM(select PlayerScores.\*

from (select PlayerScores.\*,

row\_number() over (partition by HandicapDifferential order by date desc) as seqnum

from PlayerScores where MemberNumber=@MemberNumber

) PlayerScores where MemberNumber=@MemberNumber) PlayerScores order by HandicapDifferential desc ) AS Average) AS HandicapFactor

IF @@ERROR = 0

SET @ReturnCode = 0

ELSE

RAISERROR('ERROR IN INSERT',16,1)

END

RETURN @ReturnCode

Visual Studio

\_Layout.cshtml

<!DOCTYPE html>

<html lang="eng">

<head>

<title>@ViewData["Title"]</title>

<meta charset="utf-8" />

<link rel="stylesheet" href="~/css/AStyleSheet.css" />

<script>

//script from layout

</script>

@RenderSection("scripts", required: false)

</head>

<body>

<header>

Header from layout

</header>

<nav>

<a href="BooksTeeTimeGold">BooksTeeTimeGold</a><br />

<a href="BooksTeeTimeSilver">BooksTeeTimeSilver</a><br />

<a href="BooksTeeTimeBronze">BooksTeeTimeBronze</a><br />

<a href="ModifyTeeTime">ModifyTeeTime</a>

<a href="SubmitStandingTeeTimeRequest">SubmitStandingTeeTimeRequest</a><br />

<a href="CancelsStandingTeeTimeRequest">CancelsStandingTeeTimeRequest</a><br />

<a href="RecordsMembershipApplication">RecordsMembershipApplication</a><br />

<a href="ReviewsMembershipApplication">ReviewsMembershipApplication</a><br />

<a href="ViewsMemberAccount">ViewsMemberAccount</a><br />

<a href="RecordsPlayerScores">RecordsPlayerScores</a><br />

<a href="ViewsPlayerScores">ViewsPlayerScores</a><br />

</nav>

<main>

Main from layout

@RenderBody()

</main>

<footer>

@if (IsSectionDefined("footer"))

{

@RenderSection("footer")

}

else

{

<table>

<tr>

<td>Default footer from layout</td>

</tr>

</table>

}

</footer>

</body>

</html>

AStyleSheet.css

body

{

}

header {

background-color: peachpuff;

width: 100%;

height: 10%;

position: fixed;

top: 0%;

bottom: 10%;

}

nav {

background-color: paleturquoise;

width: 100%;

height: 20%;

position: fixed;

top: 10%;

bottom: 30%;

}

main {

background-color: pink;

width: 100%;

height: 60%;

position: fixed;

top: 30%;

bottom: 90%;

}

main input {

background-color: whitesmoke;

}

footer {

background-color: dodgerblue;

width: 100%;

height: 10%;

position: fixed;

top: 90%;

bottom: 100%;

}

Domain

CBGS.cs

using System;

using System.Collections.Generic;

using System.Text;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using ClubBaistGolfSystem.TechnicalServices;

using ClubBaistGolfSystem.Domain;

namespace ClubBaistGolfSystem.Domain

{

class CBGS

{

public List<TeeTime> ViewTeeSheet(DateTime date)

{

List<TeeTime> RequestedTeeSheet;

TeeSheet TeeSheetManager = new TeeSheet();

RequestedTeeSheet = TeeSheetManager.GetTeeSheet(date);

return RequestedTeeSheet;

}

public bool BookTeeTime(TeeTime selectedTeeTime)

{

bool Confirmation;

TeeSheet TeeSheetManager = new TeeSheet();

Confirmation = TeeSheetManager.AddTeeTime(selectedTeeTime);

return Confirmation;

}

public List<StandingTeeTime> ViewStandingTeeTime(DateTime date1)

{

List<StandingTeeTime> StandingTeeTimes;

StandingTeeTimeRequests StandingTeeTimeManager = new StandingTeeTimeRequests();

StandingTeeTimes = StandingTeeTimeManager.GetStandingTeeTime(date1);

return StandingTeeTimes;

}

public bool SubmitStandingTeeTime(StandingTeeTime RequestedStandingTeeTime)

{

bool Confirmation;

StandingTeeTimeRequests StandingTeeTimeManager = new StandingTeeTimeRequests();

Confirmation = StandingTeeTimeManager.AddStandingTeeTime(RequestedStandingTeeTime);

return Confirmation;

}

public StandingTeeTime FindStandingTeeTime(string MemberNumber)

{

StandingTeeTime SearchedStandingTeeTime;

StandingTeeTimeRequests StandingTeeTimeManager = new StandingTeeTimeRequests();

SearchedStandingTeeTime = StandingTeeTimeManager.SearchStandingTeeTime(MemberNumber);

return SearchedStandingTeeTime;

}

public bool RemoveStandingTeeTimeRequest(string MemberNumber)

{

bool Confirmation2;

StandingTeeTimeRequests StandingTeeTimeManager = new StandingTeeTimeRequests();

Confirmation2 = StandingTeeTimeManager.DeleteStandingTeeTimeRequest(MemberNumber);

return Confirmation2;

}

public TeeTime ViewTeeTime(DateTime date3, DateTime time)

{

TeeTime SelectedTime = new TeeTime();

TeeSheet TeeSheetManager = new TeeSheet();

SelectedTime = TeeSheetManager.GetTeeTime(date3, time);

return SelectedTime;

}

public bool ModifyTeeTime(DateTime date3, DateTime time, TeeTime SelectedTime)

{

bool Confirmation;

TeeSheet TeeSheetManager = new TeeSheet();

Confirmation = TeeSheetManager.UpdateTeeTime(date3, time, SelectedTime);

return Confirmation;

}

public bool AddsMembershipApplication(MembershipApplication ApplicationInformation)

{

bool Confirmation;

MembershipApplications MembershipApplicationManager = new MembershipApplications();

Confirmation = MembershipApplicationManager.RecordsMembershipApplication(ApplicationInformation);

return Confirmation;

}

public List<MembershipApplication> ViewApplications(string Status)

{

List<MembershipApplication> RequestedApplications;

MembershipApplications MembershipApplicationManager = new MembershipApplications();

RequestedApplications = MembershipApplicationManager.GetMembershipAplications(Status);

return RequestedApplications;

}

public bool ModifyApplication(string MemberApplicationNumber, MembershipApplication newMembershipApplication)

{

bool Confirmation1;

MembershipApplications MembershipApplicationManager = new MembershipApplications();

Confirmation1 = MembershipApplicationManager.UpdateMembershipApplication(MemberApplicationNumber, newMembershipApplication);

return Confirmation1;

}

public bool InsertMember(Member MemberDetails)

{

bool Confirmation;

Members MemberManager = new Members();

Confirmation = MemberManager.AddMember(MemberDetails);

return Confirmation;

}

public bool InsertMemberAccount(MemberAccount AccountDetails)

{

bool Confirmation;

MemberAccounts MemberAccountManager = new MemberAccounts();

Confirmation = MemberAccountManager.AddMemberAccount(AccountDetails);

return Confirmation;

}

public MemberAccount FindAccount(string MemberAccountNumber)

{

MemberAccount RequestedAccount;

MemberAccounts MemberAccountManager = new MemberAccounts();

RequestedAccount = MemberAccountManager.GetMemberAccount(MemberAccountNumber);

return RequestedAccount;

}

public bool InsertPlayerScore(PlayerScore NewPlayerScore)

{

bool Confirmation;

PlayerScores PlayerScoreManager = new PlayerScores();

Confirmation = PlayerScoreManager.AddPlayerScore(NewPlayerScore);

return Confirmation;

}

}

}

Member.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace ClubBaistGolfSystem.Domain

{

public class Member

{

public string MemberNumber { get; set; }

public string LastName { get; set; }

public string FirstName { get; set; }

public string Address { get; set; }

public string PostalCode { get; set; }

public string Phone { get; set; }

public string Email { get; set; }

public string MemberType { get; set; }

}

}

MemberAccount.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace ClubBaistGolfSystem.Domain

{

public class MemberAccount

{

public string MemberNumber { get; set; }

public decimal Balance { get; set; }

public List<MemberAccountEntry> newMemberAccountEntries { get; set; }

}

}

MemberAccountEntry.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace ClubBaistGolfSystem.Domain

{

public class MemberAccountEntry

{

public string MemberNumber { get; set; }

public decimal Amount { get; set; }

public DateTime ActivityDate { get; set; }

public DateTime PostedDate { get; set; }

public string Description { get; set; }

}

}

MembershipApplication.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace ClubBaistGolfSystem.Domain

{

public class MembershipApplication

{

public string MemberApplicationNumber { get; set; }

public string LastName { get; set; }

public string FirstName { get; set; }

public string Address { get; set; }

public string PostalCode { get; set; }

public string Phone { get; set; }

public string Email{ get; set; }

public string DateOfBirth { get; set; }

public string Occupation { get; set; }

public string CompanyName { get; set; }

public string CompanyAddress { get; set; }

public string CompanyPostalCode { get; set; }

public string CompanyPhone { get; set; }

public string ShareholderName { get; set; }

public string Date{ get; set; }

public string Status { get; set; }

public override string ToString()

{

return String.Format("{0} {1} {2} {3} ", MemberApplicationNumber, FirstName, LastName,

Status);

}

}

}

PlayerScore.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace ClubBaistGolfSystem.Domain

{

public class PlayerScore

{

public string Course { get; set; }

public string MemberNumber { get; set; }

public decimal Rating { get; set; }

public int Slope { get; set; }

public string Date { get; set; }

public int Score { get; set; }

public int Hole1 { get; set; }

public int Hole2 { get; set; }

public int Hole3 { get; set; }

public int Hole4 { get; set; }

public int Hole5 { get; set; }

public int Hole6 { get; set; }

public int Hole7 { get; set; }

public int Hole8 { get; set; }

public int Hole9 { get; set; }

public int Hole10 { get; set; }

public int Hole11 { get; set; }

public int Hole12 { get; set; }

public int Hole13 { get; set; }

public int Hole14 { get; set; }

public int Hole15 { get; set; }

public int Hole16 { get; set; }

public int Hole17 { get; set; }

public int Hole18 { get; set; }

public decimal HandicapDifferential { get; set; }

}

}

StandingTeeTime.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace ClubBaistGolfSystem.Domain

{

public class StandingTeeTime

{

public string MemberNumber { get; set; }

public string MemberFirstName { get; set; }

public string MemberLastName { get; set; }

public string DayOfWeek { get; set; }

public string RequestedTeeTime { get; set; }

public string RequestedStartDate { get; set; }

public string RequestedEndDate { get; set; }

}

}

TeeTime.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace ClubBaistGolfSystem.Domain

{

public class TeeTime

{

public string Date { get; set; }

public string Time { get; set; }

public string NumberOfCarts { get; set; }

public string NumberOfPlayers { get; set; }

public string PlayerFirstName { get; set; }

public string PlayerLastName { get; set; }

public string Phone { get; set; }

}

}

Pages

BooksTeeTimeGold.cshtml

@page

@model BooksTeeTimeGoldModel

@{

}

<!DOCTYPE html>

<html lang="en">

<head>

<title>BooksTeeTime</title>

<meta charset="utf-8" />

<script type="text/javascript">

</script>

</head>

<body>

<header>Book Tee Time Gold</header>

<main>

<div>

<form **asp-page-handler**="FirstAction" name="ViewTeeSheet" method="post" id="BooksTeeTime">

<table>

<tr>

<td>Date:</td>

<td>

<input type="text" name="Date" value="@Model.Date" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="Submit" value="Submit" />

</td>

</tr>

</table>

@Model.Message

</form>

@if (Model.RequestedTeeSheet.Count > 0)

{

<ol>

@foreach (var TeeTime in Model.RequestedTeeSheet)

{

<li>@TeeTime</li>

}

</ol>

}

</div>

<div>

<form **asp-page-handler**="SecondAction" name="BooksTeeTime" onsubmit="return isValid(window.document.BooksTeeTime);" method="post" id="BooksTeeTime">

<table>

<tr>

<td>Date:</td>

<td>

<input type="date" name="Date" value="@Model.Date" />

</td>

</tr>

<tr>

<td>Time:</td>

<td>

<input type="Time" name="Time" value="@Model.Time" />

</td>

</tr>

<tr>

<td>Number Of Carts:</td>

<td>

<input type="text" name="Number Of Carts" value="@Model.NumberOfCarts" />

</td>

</tr>

<tr>

<td>Number Of Players:</td>

<td>

<input type="text" name="NumberOfPlayers" value="@Model.NumberOfPlayers" />

</td>

</tr>

<tr>

<td>PlayerFirstName:</td>

<td>

<input type="text" name="PlayerFirstName" value="@Model.PlayerFirstName" />

</td>

</tr>

<tr>

<td>PlayerLastName:</td>

<td>

<input type="text" name="PlayerLastName" value="@Model.PlayerLastName" />

</td>

</tr>

<tr>

<td>Phone:</td>

<td>

<input type="text" name="Phone" value="@Model.Phone" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="Submit" value="Submit" />

</td>

</tr>

</table>

</form>

</div>

@Model.Message

</main>

<footer>

</footer>

</body>

</html>

BooksTeeTimeGold.cshtml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using System.ComponentModel.DataAnnotations;

using ClubBaistGolfSystem.Domain;

namespace ClubBaistGolfSystem.Pages

{

public class BooksTeeTimeGoldModel : PageModel

{

public string Message { get; set; }

public List<TeeTime> RequestedTeeSheet { get; } = new List<TeeTime>();

[BindProperty]

[Required]

[Range(typeof(DateTime), "4/1/2020", "9/30/2020")]

public string Date { get; set; }

[BindProperty]

[Required]

public string Time { get; set; }

[BindProperty]

public string NumberOfCarts { get; set; }

[BindProperty]

[Required]

public string NumberOfPlayers { get; set; }

[BindProperty]

[Required]

public string PlayerFirstName { get; set; }

[BindProperty]

[Required]

public string PlayerLastName { get; set; }

[BindProperty]

[Required]

public string Phone { get; set; }

public void OnGet()

{

}

public void OnPostFirstAction()

{

CBGS RequestDirector = new CBGS();

DateTime date = DateTime.Parse(Date);

RequestedTeeSheet.AddRange(RequestDirector.ViewTeeSheet(date));

}

public void OnPostSecondAction()

{

TeeTime selectedTeeTime = new TeeTime();

selectedTeeTime.Date = Date;

selectedTeeTime.Time = Time;

selectedTeeTime.NumberOfCarts = NumberOfCarts;

selectedTeeTime.NumberOfPlayers = NumberOfPlayers;

selectedTeeTime.PlayerFirstName = PlayerFirstName;

selectedTeeTime.PlayerLastName = PlayerLastName;

selectedTeeTime.Phone = Phone;

CBGS RequestDirector = new CBGS();

bool Confirmation;

if (ModelState.IsValid)

{

Confirmation = RequestDirector.BookTeeTime(selectedTeeTime);

if(Confirmation)

Message = "Tee Time For Gold Level Player Booked";

}

else

{

Message = "Tee Time For Gold Level Player Not Booked";

}

}

}

}

BooksTeeTimeSilver.cshtml

@page

@model BooksTeeTimeSilverModel

@{

}

<!DOCTYPE html>

<html lang="en">

<head>

<title>BooksTeeTimeSilver</title>

<meta charset="utf-8" />

<script type="text/javascript">

</script>

</head>

<body>

<header>Book Tee Time Silver</header>

<main>

<div>

<form **asp-page-handler**="FirstAction" name="View Tee Sheet" method="post" id="BooksTeeTime">

<table>

<tr>

<td>Date:</td>

<td>

<input type="date" name="Date" value="@Model.Date" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="Submit" value="Submit" />

</td>

</tr>

</table>

@Model.Message

</form>

@if (Model.RequestedTeeSheet.Count > 0)

{

<ol>

@foreach (var TeeTime in Model.RequestedTeeSheet)

{

<li>@TeeTime</li>

}

</ol>

}

</div>

<div>

<form **asp-page-handler**="SecondAction" name="BookTeeTime" method="post" id="BooksTeeTime">

<table>

<tr>

<td>Date:</td>

<td>

<input type="date" name="Date" value="@Model.Date" />

</td>

</tr>

<tr>

<td>Time:</td>

<td>

<input type="Time" name="Time" value="@Model.Time" />

</td>

</tr>

<tr>

<td>Number Of Carts:</td>

<td>

<input type="text" name="Number Of Carts" value="@Model.NumberOfCarts" />

</td>

</tr>

<tr>

<td>Number Of Players:</td>

<td>

<input type="text" name="NumberOfPlayers" value="@Model.NumberOfPlayers" />

</td>

</tr>

<tr>

<td>PlayerFirstName:</td>

<td>

<input type="text" name="PlayerFirstName" value="@Model.PlayerFirstName" />

</td>

</tr>

<tr>

<td>PlayerLastName:</td>

<td>

<input type="text" name="PlayerLastName" value="@Model.PlayerLastName" />

</td>

</tr>

<tr>

<td>Phone:</td>

<td>

<input type="text" name="Phone" value="@Model.Phone" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="Submit" value="Submit" />

</td>

</tr>

</table>

</form>

</div>

@Model.Message

</main>

<footer>

</footer>

</body>

</html>

BooksTeeTimeSilver.cshtml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using System.ComponentModel.DataAnnotations;

using ClubBaistGolfSystem.Domain;

namespace ClubBaistGolfSystem.Pages

{

public class BooksTeeTimeSilverModel : PageModel

{

private bool validTime;

public string Message { get; set; }

public List<TeeTime> RequestedTeeSheet { get; } = new List<TeeTime>();

[BindProperty]

[Required]

[Range(typeof(DateTime), "4/1/2020", "9/30/2020")]

public string Date { get; set; }

[BindProperty]

[Required]

public string Time

{

get

{

return Time;

}

set

{

int hours = int.Parse(value.Substring(0, 2));

int mins = int.Parse(value.Substring(3));

validTime = hours > 17 && mins > 30 || hours < 15 && mins < 00;

if (validTime)

Time = value;

}

}

[BindProperty]

public string NumberOfCarts { get; set; }

[BindProperty]

[Required]

public string NumberOfPlayers { get; set; }

[BindProperty]

[Required]

public string PlayerFirstName { get; set; }

[BindProperty]

[Required]

public string PlayerLastName { get; set; }

[BindProperty]

[Required]

public string Phone{ get; set; }

public void OnGet()

{

}

public void OnPostFirstAction()

{

CBGS RequestDirector = new CBGS();

DateTime date = DateTime.Parse(Date);

RequestedTeeSheet.AddRange(RequestDirector.ViewTeeSheet(date));

}

public void OnPostSecondAction()

{

TeeTime selectedTeeTime = new TeeTime();

selectedTeeTime.Date = Date;

selectedTeeTime.Time = Time;

selectedTeeTime.NumberOfCarts = NumberOfCarts;

selectedTeeTime.NumberOfPlayers = NumberOfPlayers;

selectedTeeTime.PlayerFirstName = PlayerFirstName;

selectedTeeTime.PlayerLastName = PlayerLastName;

selectedTeeTime.Phone = Phone;

CBGS RequestDirector = new CBGS();

bool Confirmation;

if (ModelState.IsValid)

{

Confirmation = RequestDirector.BookTeeTime(selectedTeeTime);

if (validTime)

Message = "Tee Time For Silver Level Player Booked";

else

Message = "Not Valid Time for this membership level";

}

else

{

Message = "Tee Time For Silver Level Player Not Booked";

}

}

}

}

BooksTeeTimeBronze.cshtml

@page

@model BooksTeeTimeBronzeModel

@{

}

<!DOCTYPE html>

<html lang="en">

<head>

<title>BooksTeeTimeBronze</title>

<meta charset="utf-8" />

<script type="text/javascript">

</script>

</head>

<body>

<header>Book Tee Time Bronze</header>

<main>

<div>

<form **asp-page-handler**="FirstAction" name="View Tee Sheet" method="post" id="BooksTeeTime">

<table>

<tr>

<td>Date:</td>

<td>

<input type="date" name="Date" value="@Model.Date" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="Submit" value="Submit" />

</td>

</tr>

</table>

@Model.Message

</form>

@if (Model.RequestedTeeSheet.Count > 0)

{

<ol>

@foreach (var TeeTime in Model.RequestedTeeSheet)

{

<li>@TeeTime</li>

}

</ol>

}

</div>

<div>

<form **asp-page-handler**="SecondAction" name="BooksTeeTime" method="post" id="BooksTeeTime">

<table>

<tr>

<td>Date:</td>

<td>

<input type="date" name="Date" value="@Model.Date" />

</td>

</tr>

<tr>

<td>Time:</td>

<td>

<input type="Time" name="Time" value="@Model.Time" />

</td>

</tr>

<tr>

<td>Number Of Carts:</td>

<td>

<input type="text" name="Number Of Carts" value="@Model.NumberOfCarts" />

</td>

</tr>

<tr>

<td>Number Of Players:</td>

<td>

<input type="text" name="NumberOfPlayers" value="@Model.NumberOfPlayers" />

</td>

</tr>

<tr>

<td>PlayerFirstName:</td>

<td>

<input type="text" name="PlayerFirstName" value="@Model.PlayerFirstName" />

</td>

</tr>

<tr>

<td>PlayerLastName:</td>

<td>

<input type="text" name="PlayerLastName" value="@Model.PlayerLastName" />

</td>

</tr>

<tr>

<td>Phone:</td>

<td>

<input type="text" name="Phone" value="@Model.Phone" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="Submit" value="Submit" />

</td>

</tr>

</table>

</form>

</div>

@Model.Message

</main>

<footer>

</footer>

</body>

</html>

BooksTeeTimeBronze.cshtml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using System.ComponentModel.DataAnnotations;

using ClubBaistGolfSystem.Domain;

namespace ClubBaistGolfSystem.Pages

{

public class BooksTeeTimeBronzeModel : PageModel

{

public string Message { get; set; }

private bool validTime;

public List<TeeTime> RequestedTeeSheet { get; } = new List<TeeTime>();

[BindProperty]

[Required]

[Range(typeof(DateTime), "4/1/2020", "9/30/2020")]

public string Date { get; set; }

[BindProperty]

[Required]

public string Time

{

get

{

return Time;

}

set

{

int hours = int.Parse(value.Substring(0, 2));

int mins = int.Parse(value.Substring(3));

validTime = hours > 18 && mins > 00 || hours < 15 && mins < 00;

if (validTime)

Time = value;

}

}

[BindProperty]

public string NumberOfCarts { get; set; }

[BindProperty]

[Required]

public string NumberOfPlayers { get; set; }

[BindProperty]

[Required]

public string PlayerFirstName { get; set; }

[BindProperty]

[Required]

public string PlayerLastName { get; set; }

[BindProperty]

[Required]

public string Phone { get; set; }

public void OnGet()

{

}

public void OnPostFirstAction()

{

CBGS RequestDirector = new CBGS();

DateTime date = DateTime.Parse(Date);

RequestedTeeSheet.AddRange(RequestDirector.ViewTeeSheet(date));

}

public void OnPostSecondAction()

{

TeeTime selectedTeeTime = new TeeTime();

selectedTeeTime.Date = Date;

selectedTeeTime.Time = Time;

selectedTeeTime.NumberOfCarts = NumberOfCarts;

selectedTeeTime.NumberOfPlayers = NumberOfPlayers;

selectedTeeTime.PlayerFirstName = PlayerFirstName;

selectedTeeTime.Phone = Phone;

CBGS RequestDirector = new CBGS();

bool Confirmation;

if (ModelState.IsValid)

{

Confirmation = RequestDirector.BookTeeTime(selectedTeeTime);

if (validTime)

Message = "Tee Time For Bronze Level Player Booked";

else

Message = "Not Valid Time for this membership level";

}

else

{

Message = "Tee Time For Bronze Level Player Not Booked";

}

}

}

}

ModifiesTeeTime.cshtml

@page

@model ClubBaistGolfSystem.Pages.ModifiesTeeTimeModel

@{

}

<!DOCTYPE html>

<html lang="en">

<head>

<title>BooksTeeTimeSilver</title>

<meta charset="utf-8" />

<script type="text/javascript">

</script>

</head>

<body>

<header>ModifiesTeeTime</header>

<main>

<div>

<form **asp-page-handler**="FirstAction" name="View Tee Sheet" method="post" id="BooksTeeTime">

<table>

<tr>

<td>Date:</td>

<td>

<input type="date" name="Date" value="@Model.Date" />

</td>

</tr>

<tr>

<td>Time:</td>

<td>

<input type="time" name="Time" value="@Model.Time" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="Submit" value="Submit" />

</td>

</tr>

</table>

@Model.Message1

</form>

</div>

<div>

<form **asp-page-handler**="SecondAction" name="ModifiesTeeTime" method="post" id="ModifiesTeeTime">

<table>

<tr>

<td>Date:</td>

<td>

<input type="date" name="Date" value="@Model.Date" />

</td>

</tr>

<tr>

<td>Time:</td>

<td>

<input type="time" name="Time" value="@Model.Time" />

</td>

</tr>

<tr>

<td>PlayerFirstName:</td>

<td>

<input type="text" name="PlayerFirstName" value="@Model.PlayerFirstName" />

</td>

</tr>

<tr>

<td>PlayerLastName:</td>

<td>

<input type="text" name="PlayerLastName" value="@Model.PlayerLastName" />

</td>

</tr>

<tr>

<td>Phone:</td>

<td>

<input type="text" name="Phone" value="@Model.Phone" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="Submit" value="Submit" />

</td>

</tr>

</table>

</form>

</div>

@Model.Message2

</main>

<footer>

</footer>

</body>

</html>

ModifiesTeeTime.cshtml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using System.ComponentModel.DataAnnotations;

using ClubBaistGolfSystem.Domain;

using ClubBaistGolfSystem.TechnicalServices;

namespace ClubBaistGolfSystem.Pages

{

public class ModifiesTeeTimeModel : PageModel

{

public string Message1 { get; set; }

public string Message2 { get; set; }

[BindProperty]

[Required]

public string Date { get; set; }

[BindProperty]

[Required]

public string Time { get; set; }

[BindProperty]

[Required]

public string PlayerFirstName { get; set; }

[BindProperty]

[Required]

public string PlayerLastName { get; set; }

[BindProperty]

[Required]

public string Phone { get; set; }

public void OnGet()

{

}

public void OnPostFirstAction()

{

TeeTime SelectedTime = new TeeTime();

CBGS RequestDirector = new CBGS();

DateTime date3 = DateTime.Parse(Date);

DateTime time = DateTime.Parse(Time);

//var time = Convert.ToDateTime(Time);

SelectedTime = RequestDirector.ViewTeeTime(date3, time);

Message1 = "PlayerFirstName is----" + SelectedTime.PlayerFirstName + "----PlayerLastNameis----" +

SelectedTime.PlayerLastName + "-----Phone is--" + SelectedTime.Phone;

}

public void OnPostSecondAction()

{

TeeTime selectedTime1 = new TeeTime();

DateTime date3 = DateTime.Parse(Date);

DateTime time = DateTime.Parse(Time);

selectedTime1.PlayerFirstName = PlayerFirstName;

selectedTime1.PlayerLastName = PlayerLastName;

selectedTime1.Phone = Phone;

CBGS RequestDirector = new CBGS();

bool Confirmation;

Confirmation = RequestDirector.ModifyTeeTime(date3,time,selectedTime1);

if (Confirmation)

Message2 = "Tee Time Modified";

else

Message2 = "Tee Time Not Modified";

}

}

}

MakesStandingTeeTime.cshtml

@page

@model MakesStandingTeeTimeRequestModel

@{

}

<!DOCTYPE html>

<html lang="en">

<head>

<title>MakesStandingTeeTimeRequest</title>

<meta charset="utf-8" />

<script type="text/javascript">

</script>

</head>

<body>

<header>Makes Standing Tee Time Request</header>

<main>

<div>

<form **asp-page-handler**="FirstAction" name="View StandingTeeTimes" method="post" id="StandingTeeTimeRequest">

<table>

<tr>

<td>Requested Start Date:</td>

<td>

<input **asp-for**="RequestedStartDate" **type**="date" **name**="RequestedStartDate" **value**="@Model.RequestedStartDate" />

<span **asp-validation-for**="RequestedStartDate"></span>

</td>

</tr>

<tr>

<td>

<input type="submit" name="Submit" value="Submit" />

</td>

</tr>

</table>

@Model.Message

</form>

</div>

@if (Model.StandingTeeTimes.Count > 0)

{

<ol>

@foreach (var StandingTeeTime in Model.StandingTeeTimes)

{

<li>@StandingTeeTime</li>

}

</ol>

}

<div>

<form **asp-page-handler**="SecondAction" name="Makes Standing Tee Time Request" method="post" id="MakesStandingTeeTimeRequest">

<table>

<tr>

<td>MemberNumber:</td>

<td>

<input type="text" name="MemberNumber" value="@Model.MemberNumber" />

</td>

</tr>

<tr>

<td>MemberFirstName:</td>

<td>

<input type="text" name="MemberFirstName" value="@Model.MemberFirstName" />

</td>

</tr>

<tr>

<td>MemberLastName:</td>

<td>

<input type="text" name="MemberLastName" value="@Model.MemberLastName" />

</td>

</tr>

<tr>

<td>DayOfWeek:</td>

<td>

<input type="text" name="DayOfWeek" value="@Model.DayOfWeek" />

</td>

</tr>

<tr>

<td>RequestedTeeTime:</td>

<td>

<input **asp-for**="RequestedTeeTime" **type**="time" **name**="RequestedTeeTime" **value**="@Model.RequestedTeeTime" />

<span **asp-validation-for**="RequestedTeeTime"></span>

</td>

</tr>

<tr>

<td>RequestedStartDate:</td>

<td>

<input **asp-for**="RequestedStartDate" **type**="date" **name**="RequestedStartDate" **value**="@Model.RequestedStartDate" />

<span **asp-validation-for**="RequestedStartDate"></span>

</td>

</tr>

<tr>

<td>RequestedEndDate:</td>

<td>

<input **asp-for**="RequestedEndDate" **type**="date" **name**="RequestedEndDate" **value**="@Model.RequestedEndDate" />

<span **asp-validation-for**="RequestedEndDate"></span>

</td>

</tr>

<tr>

<td>

<input type="submit" name="Submit" value="Submit" />

</td>

</tr>

</table>

</form>

</div>

@Model.Message

</main>

<footer>

</footer>

</body>

</html>

MakesStandingTeeTime.cshtml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using ClubBaistGolfSystem.Domain;

using System.ComponentModel.DataAnnotations;

using ClubBaistGolfSystem.TechnicalServices;

namespace ClubBaistGolfSystem.Pages

{

public class MakesStandingTeeTimeRequestModel : PageModel

{

public List<StandingTeeTime> StandingTeeTimes { get; } = new List<StandingTeeTime>();

public string Message { get; set; }

[BindProperty]

public string MemberNumber { get; set; }

[BindProperty]

public string MemberFirstName { get; set; }

[BindProperty]

public string MemberLastName { get; set; }

[BindProperty]

public string DayOfWeek { get; set; }

[BindProperty]

[Required]

public string RequestedTeeTime { get; set; }

[BindProperty]

[Required]

//[Range(typeof(DateTime), "4/1/2020", "9/30/2020")]

public string RequestedStartDate { get; set; }

[BindProperty]

//[Range(typeof(DateTime), "4/1/2020", "4/9/2020")]

public string RequestedEndDate { get; set; }

public void OnGet()

{

}

public void OnPostFirstAction()

{

CBGS RequestDirector = new CBGS();

DateTime date = DateTime.Parse(RequestedStartDate);

StandingTeeTimes.AddRange(RequestDirector.ViewStandingTeeTime(date));

}

public void OnPostSecondAction()

{

StandingTeeTime RequestedStandingTeeTime = new StandingTeeTime();

RequestedStandingTeeTime.MemberNumber = MemberNumber;

RequestedStandingTeeTime.MemberFirstName = MemberFirstName;

RequestedStandingTeeTime.MemberLastName = MemberLastName;

RequestedStandingTeeTime.DayOfWeek = DayOfWeek;

RequestedStandingTeeTime.RequestedTeeTime = RequestedTeeTime;

RequestedStandingTeeTime.RequestedStartDate = RequestedStartDate;

RequestedStandingTeeTime.RequestedEndDate = RequestedEndDate;

bool Confirmation;

CBGS RequestDirector = new CBGS();

if (ModelState.IsValid)

{

Confirmation = RequestDirector.SubmitStandingTeeTime(RequestedStandingTeeTime);

Message = "StandingTeeTimeRequest Submitted";

}

else

{

Message = "StandingTeeTimeRequest Not Submitted";

}

}

}

}

CancelsStandingTeeTimeRequest.cshtml

@page

@model ClubBaistGolfSystem.Pages.CancelsStandingTeeTimeRequestModel

@{

}

<!DOCTYPE html>

<html lang="en">

<head>

<title>CancelsStandingTeeTimeRequest</title>

<meta charset="utf-8" />

<script type="text/javascript">

</script>

</head>

<body>

<header>Cancel Standing Tee Time Request</header>

<main>

<div>

<form **asp-page-handler**="FirstAction" name="View StandingTeeTimes" method="post" id="StandingTeeTimeRequest">

<table>

<tr>

<td>MemberNumbere:</td>

<td>

<input **asp-for**="MemberNumber" **type**="number" **name**="MemberNumber" **value**="@Model.MemberNumber" />

<span **asp-validation-for**="MemberNumber"></span>

</td>

</tr>

<tr>

<td>

<input type="submit" name="FindStandingTeeTime" value="Submit" />

</td>

<td>

<input type="reset" name="Reset" value="Reset" />

</td>

</tr>

</table>

@Model.Message1

</form>

</div>

<div>

<form **asp-page-handler**="SecondAction" name="Cancels Standing Tee Time Request" method="post" id="CancelsStandingTeeTimeRequest">

<table>

<tr>

<td>MemberNumber:</td>

<td>

<input type="text" name="MemberNumber" value="@Model.MemberNumber" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="CancelsStandingTeeTimeRequest" value="Cancel" />

</td>

</tr>

</table>

</form>

</div>

@Model.Message2

</main>

<footer>

</footer>

</body>

</html>

CancelsStandingTeeTimeRequest.cshtml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using ClubBaistGolfSystem.Domain;

using ClubBaistGolfSystem.TechnicalServices;

namespace ClubBaistGolfSystem.Pages

{

public class CancelsStandingTeeTimeRequestModel : PageModel

{

public string Message1 { get; set; }

public string Message2 { get; set; }

[BindProperty]

public string MemberNumber { get; set; }

public void OnGet()

{

}

public void OnPostFirstAction()

{

StandingTeeTime SearchedStandingTeeTime;

CBGS RequestDirector = new CBGS();

SearchedStandingTeeTime = RequestDirector.FindStandingTeeTime(MemberNumber);

Message1 = SearchedStandingTeeTime.MemberFirstName + "---" + SearchedStandingTeeTime.MemberLastName;

}

public void OnPostSecondAction()

{

bool Confirmation;

CBGS RequestDirector = new CBGS();

Confirmation = RequestDirector.RemoveStandingTeeTimeRequest(MemberNumber);

if (Confirmation)

{

Message2 = "Standing TeeTime Request Cancelled";

}

else

{

Message2 = "Standing TeeTime Request not Cancelled";

}

}

}

}

RecordsMembershipApplication.cshtml

@page

@model ClubBaistGolfSystem.Pages.RecordsMembershipApplicationModel

@{

}

<!DOCTYPE html>

<html lang="en">

<head>

<title>Records Membership Application</title>

<meta charset="utf-8" />

<script type="text/javascript">

</script>

</head>

<body>

<header>Records Membership Application</header>

<main>

<div>

<form name="BookTeeTime" method="post" id="BooksTeeTime">

<table>

<tr>

<td>FirstName:</td>

<td>

<input type="text" name="FirstName" value="@Model.FirstName" />

</td>

</tr>

<tr>

<td>LastName:</td>

<td>

<input type="text" name="LastName" value="@Model.LastName" />

</td>

</tr>

<tr>

<td>Address:</td>

<td>

<input type="text" name="Address" value="@Model.Address" />

</td>

</tr>

<tr>

<td>PostalCode:</td>

<td>

<input type="text" name="PostalCode" value="@Model.PostalCode" />

</td>

</tr>

<tr>

<td>Phone:</td>

<td>

<input type="text" name="Phone" value="@Model.Phone" />

</td>

</tr>

<tr>

<td>Email:</td>

<td>

<input type="text" name="Email" value="@Model.Email" />

</td>

</tr>

<tr>

<td>DateOfBirth:</td>

<td>

<input type="date" name="DateOfBirth" value="@Model.DateOfBirth" />

</td>

</tr>

<tr>

<td>Occupation:</td>

<td>

<input type="text" name="Occupation" value="@Model.Occupation" />

</td>

</tr>

<tr>

<td>CompanyName:</td>

<td>

<input type="text" name="CompanyName" value="@Model.CompanyName" />

</td>

</tr>

<tr>

<td>CompanyAddress:</td>

<td>

<input type="text" name="CompanyAddress" value="@Model.CompanyAddress" />

</td>

</tr>

<tr>

<td>CompanyPostalCode:</td>

<td>

<input type="text" name="CompanyPostalCode" value="@Model.CompanyPostalCode" />

</td>

</tr>

<tr>

<td>CompanyPhone:</td>

<td>

<input type="text" name="CompanyPhone" value="@Model.CompanyPhone" />

</td>

</tr>

<tr>

<td>ShareholderName:</td>

<td>

<input type="text" name="ShareholderName" value="@Model.ShareholderName" />

</td>

</tr>

<tr>

<td>Date:</td>

<td>

<input type="date" name="Date" value="@Model.Date" />

</td>

</tr>

<tr>

<td>Status:</td>

<td>

<input type="text" name="Status" value="@Model.Status" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="Submit" value="Submit" />

</td>

</tr>

</table>

</form>

</div>

@Model.Message

</main>

<footer>

</footer>

</body>

</html>

RecordsMembershipApplication.cshtml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using ClubBaistGolfSystem.Domain;

namespace ClubBaistGolfSystem.Pages

{

public class RecordsMembershipApplicationModel : PageModel

{

[BindProperty]

public string FirstName { get; set; }

[BindProperty]

public string LastName { get; set; }

[BindProperty]

public string Address { get; set; }

[BindProperty]

public string PostalCode { get; set; }

[BindProperty]

public string Phone { get; set; }

[BindProperty]

public string Email { get; set; }

[BindProperty]

public string DateOfBirth { get; set; }

[BindProperty]

public string Occupation { get; set; }

[BindProperty]

public string CompanyName { get; set; }

[BindProperty]

public string CompanyAddress { get; set; }

[BindProperty]

public string CompanyPostalCode { get; set; }

[BindProperty]

public string CompanyPhone { get; set; }

[BindProperty]

public string ShareholderName { get; set; }

[BindProperty]

public string Date { get; set; }

[BindProperty]

public string Status { get; set; }

public string Message { get; set; }

public void OnGet()

{

}

public void OnPost()

{

MembershipApplication ApplicationInformation = new MembershipApplication();

ApplicationInformation.FirstName = FirstName;

ApplicationInformation.LastName = LastName;

ApplicationInformation.Address = Address;

ApplicationInformation.PostalCode = PostalCode;

ApplicationInformation.Phone= Phone;

ApplicationInformation.Email= Email;

ApplicationInformation.DateOfBirth= DateOfBirth;

ApplicationInformation.Occupation = Occupation;

ApplicationInformation.CompanyName = CompanyName;

ApplicationInformation.CompanyAddress = CompanyAddress;

ApplicationInformation.CompanyPostalCode= CompanyPostalCode;

ApplicationInformation.CompanyPhone= CompanyPhone;

ApplicationInformation.ShareholderName = ShareholderName;

ApplicationInformation.Date = Date;

ApplicationInformation.Status = Status;

CBGS RequestDirector = new CBGS();

bool Confirmation;

Confirmation = RequestDirector.AddsMembershipApplication(ApplicationInformation);

if(Confirmation)

Message = "Membership Application Recorded";

else

Message = "Membership Application Not Recorded";

}

}

}

ReviewsMembershipApplication.cshtml

@page

@model ClubBaistGolfSystem.Pages.ReviewsMembershipApplicationModel

@{

}

<!DOCTYPE html>

<html lang="en">

<head>

<title>ReviewsMembershipApplication</title>

<meta charset="utf-8" />

<script type="text/javascript">

</script>

</head>

<body>

<header>ReviewsMembershipApplication</header>

<main>

<div>

<form **asp-page-handler**="FirstAction" name="ViewApplications" method="post" id="ViewApplications">

<table>

<tr>

<td>Status:</td>

<td>

<input type="text" name="Status" value="@Model.Status" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="Submit" value="Submit" />

</td>

</tr>

</table>

@Model.Message1

</form>

@if (Model.RequestedApplications.Count > 0)

{

<ol>

@foreach (var OnHoldApplications in Model.RequestedApplications)

{

<li>@OnHoldApplications</li>

}

</ol>

}

</div>

<div>

<form **asp-page-handler**="SecondAction" name="ModifyMembershipApplication" method="post" id="ModifyMembershipApplication">

<table>

<tr>

<td>MemberApplicationNumber:</td>

<td>

<input type="text" name="MemberApplicationNumber" value="@Model.MemberApplicationNumber" />

</td>

</tr>

<tr>

<td>FirstName:</td>

<td>

<input type="text" name="FirstName" value="@Model.FirstName" />

</td>

</tr>

<tr>

<td>LastName:</td>

<td>

<input type="text" name="LastName" value="@Model.LastName" />

</td>

</tr>

<tr>

<td>Address:</td>

<td>

<input type="text" name="Address" value="@Model.Address" />

</td>

</tr>

<tr>

<td>PostalCode:</td>

<td>

<input type="text" name="PostalCode" value="@Model.PostalCode" />

</td>

</tr>

<tr>

<td>Phone:</td>

<td>

<input type="text" name="Phone" value="@Model.Phone" />

</td>

</tr>

<tr>

<td>Email:</td>

<td>

<input type="text" name="Email" value="@Model.Email" />

</td>

</tr>

<tr>

<td>Status:</td>

<td>

<input type="text" name="Status" value="@Model.Status" />

</td>

</tr>

<tr>

<td>MemberType:</td>

<td>

<input type="text" name="MemberType" value="@Model.MemberType" />

</td>

</tr>

<tr>

<td>Balance:</td>

<td>

<input type="text" name="Balance" value="@Model.Balance" />

</td>

</tr>

<tr>

<td>Amount:</td>

<td>

<input type="text" name="Amount" value="@Model.Amount" />

</td>

</tr>

<tr>

<td>ActivityDate:</td>

<td>

<input type="date" name="ActivityDate" value="@Model.ActivityDate" />

</td>

</tr>

<tr>

<td>PostedDate:</td>

<td>

<input type="date" name="PostedDate" value="@Model.PostedDate" />

</td>

</tr>

<tr>

<td>Description:</td>

<td>

<input type="text" name="Description" value="@Model.Description" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="Submit" value="Submit" />

</td>

</tr>

</table>

</form>

</div>

@Model.Message2

<br/>

</main>

<footer>

</footer>

</body>

</html>

ReviewsMembershipApplication.cshtml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using ClubBaistGolfSystem.Domain;

using ClubBaistGolfSystem.TechnicalServices;

namespace ClubBaistGolfSystem.Pages

{

public class ReviewsMembershipApplicationModel : PageModel

{

[BindProperty]

public string Status { get; set; }

[BindProperty]

public string MemberApplicationNumber { get; set; }

[BindProperty]

public string FirstName { get; set; }

[BindProperty]

public string LastName { get; set; }

[BindProperty]

public string Address { get; set; }

[BindProperty]

public string PostalCode { get; set; }

[BindProperty]

public string Phone { get; set; }

[BindProperty]

public string Email { get; set; }

[BindProperty]

public string MemberType { get; set; }

[BindProperty]

public double Balance { get; set; }

[BindProperty]

public double Amount { get; set; }

[BindProperty]

public DateTime ActivityDate { get; set; }

[BindProperty]

public DateTime PostedDate { get; set; }

[BindProperty]

public string Description { get; set; }

public string Message1 { get; set; }

public string Message2 { get; set; }

public List<MembershipApplication> RequestedApplications { get; set; } = new List<MembershipApplication>();

public List<MemberAccountEntry> MemberAccountEntries { get; set; }

public void OnGet()

{

}

public void OnPostFirstAction()

{

CBGS RequestDirector = new CBGS();

RequestedApplications = RequestDirector.ViewApplications(Status);

}

public void OnPostSecondAction()

{

bool Confirmation1;

MembershipApplication newMembershipApplication = new MembershipApplication();

newMembershipApplication.FirstName = FirstName;

newMembershipApplication.LastName = LastName;

newMembershipApplication.Address = Address;

newMembershipApplication.PostalCode = PostalCode;

newMembershipApplication.Phone = Phone;

newMembershipApplication.Email = Email;

newMembershipApplication.Status = Status;

Member MemberDetails = new Member();

MemberDetails.MemberNumber = MemberApplicationNumber;

MemberDetails.FirstName = FirstName;

MemberDetails.LastName = LastName;

MemberDetails.Address = Address;

MemberDetails.PostalCode = PostalCode;

MemberDetails.Phone = Phone;

MemberDetails.Email = Email;

MemberDetails.MemberType = MemberType;

CBGS RequestDirector = new CBGS();

RequestDirector.InsertMember(MemberDetails);

MemberAccount NewMemberAccount = new MemberAccount();

NewMemberAccount.MemberNumber = MemberApplicationNumber;

NewMemberAccount.Balance = Convert.ToDecimal(Balance);

List<MemberAccountEntry> MemberAccountEntries = new List<MemberAccountEntry>();

MemberAccountEntry Entries = new MemberAccountEntry();

Entries.MemberNumber = MemberApplicationNumber;

Entries.Amount = Convert.ToDecimal(Amount);

Entries.ActivityDate = ActivityDate;

Entries.PostedDate = PostedDate;

Entries.Description = Description;

MemberAccountEntries.Add(Entries);

RequestDirector.InsertMemberAccount(NewMemberAccount);

Confirmation1 = RequestDirector.ModifyApplication(MemberApplicationNumber, newMembershipApplication);

if (Confirmation1)

Message2 = "Application Modified";

}

}

}

ViewsMemberAccount.cshtml

@page

@model ClubBaistGolfSystem.Pages.ViewsMemberAccountModel

@{

}

<!DOCTYPE html>

<html lang="en">

<head>

<title>ViewsMemberAccount</title>

<meta charset="utf-8" />

<script type="text/javascript">

</script>

</head>

<body>

<header>ViewsMemberAccount</header>

<main>

<div>

<form name="ViewsMemberAccount" method="post" id="ViewsMemberAccount">

<table>

<tr>

<td>MemberAccountNumber:</td>

<td>

<input id="Text1" type="text" name="MemberAccountNumber" value="@Model.MemberAccountNumber" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="FindAccount" />

</td>

</tr>

</table>

</form>

@if (Model.newMemberAccountEntries.Count > 0)

{

<ol>

@foreach (var newMemberAccountEntry in Model.newMemberAccountEntries)

{

<li>@newMemberAccountEntry</li>

}

</ol>

}

</div>

</main>

<footer>

</footer>

</body>

</html>

ViewsMemberAccount.cshtml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using ClubBaistGolfSystem.Domain;

using ClubBaistGolfSystem.Pages;

namespace ClubBaistGolfSystem.Pages

{

public class ViewsMemberAccountModel : PageModel

{

[BindProperty]

public string MemberAccountNumber { get; set; }

public List<MemberAccountEntry> newMemberAccountEntries { get; set; } = new List<MemberAccountEntry>();

public string Message { get; set; }

public void OnGet()

{

}

public void OnPost()

{

CBGS RequestDirector = new CBGS();

newMemberAccountEntries = RequestDirector.FindAccount(MemberAccountNumber).newMemberAccountEntries;

}

}

}

RecordsPlayerScores.cshtml

@page

@model ClubBaistGolfSystem.Pages.RecordsPlayerScoresModel

@{

}

<!DOCTYPE html>

<html lang="en">

<head>

<title>RecordsPlayerScores</title>

<meta charset="utf-8" />

<script type="text/javascript">

</script>

</head>

<body>

<header>Records Player Scores</header>

<main>

<div>

<form name="RecordsPlayerScores" method="post" id="RecordsPlayerScores">

<div>

<label>Course</label>

<input type="text" name="Course" />

</div>

<div>

<label>MemberNumber</label>

<input type="text" name="MemberNumber" />

</div>

<div>

<label>Rating</label>

<input type="text" name="Rating" />

</div>

<div>

<label>Slope</label>

<input type="text" name="Slope" />

</div>

<div>

<label>Date</label>

<input type="text" name="Date" />

</div>

<div>

<label>Hole1</label>

<input type="text" name="Hole1" />

<label>Hole2</label>

<input type="text" name="Hole2" />

<label>Hole3</label>

<input type="text" name="Hole3" />

<label>Hole4</label>

<input type="text" name="Hole4" />

<label>Hole5</label>

<input type="text" name="Hole5" />

<label>Hole6</label>

<input type="text" name="Hole6" /><br />

<label>Hole7</label>

<input type="text" name="Hole7" />

<label>Hole8</label>

<input type="text" name="Hole8" />

<label>Hole9</label>

<input type="text" name="Hole9" />

<label>Hole10</label>

<input type="text" name="Hole10" />

<label>Hole11</label>

<input type="text" name="Hole11" />

<label>Hole12</label>

<input type="text" name="Hole12" /><br />

<label>Hole13</label>

<input type="text" name="Hole13" />

<label>Hole14</label>

<input type="text" name="Hole14" />

<label>Hole15</label>

<input type="text" name="Hole15" />

<label>Hole16</label>

<input type="text" name="Hole16" />

<label>Hole17</label>

<input type="text" name="Hole17" />

<label>Hole18</label>

<input type="text" name="Hole18" />

</div>

<div>

<input type="submit" name="Submit" value="Submit" />

</div>

</form>

</div>

@Model.Message

</main>

<footer>

</footer>

</body>

</html>

RecordsPlayerScores.cshtml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using ClubBaistGolfSystem.Domain;

namespace ClubBaistGolfSystem.Pages

{

public class RecordsPlayerScoresModel : PageModel

{

[BindProperty]

public string Course { get; set; }

[BindProperty]

public string MemberNumber { get; set; }

[BindProperty]

public decimal Rating { get; set; }

[BindProperty]

public int Slope { get; set; }

[BindProperty]

public string Date { get; set; }

[BindProperty]

public int Score { get; set; }

[BindProperty]

public int Hole1 { get; set; }

[BindProperty]

public int Hole2 { get; set; }

[BindProperty]

public int Hole3 { get; set; }

[BindProperty]

public int Hole4 { get; set; }

[BindProperty]

public int Hole5 { get; set; }

[BindProperty]

public int Hole6 { get; set; }

[BindProperty]

public int Hole7 { get; set; }

[BindProperty]

public int Hole8 { get; set; }

[BindProperty]

public int Hole9 { get; set; }

[BindProperty]

public int Hole10 { get; set; }

[BindProperty]

public int Hole11 { get; set; }

[BindProperty]

public int Hole12 { get; set; }

[BindProperty]

public int Hole13 { get; set; }

[BindProperty]

public int Hole14 { get; set; }

[BindProperty]

public int Hole15 { get; set; }

[BindProperty]

public int Hole16 { get; set; }

[BindProperty]

public int Hole17 { get; set; }

[BindProperty]

public int Hole18 { get; set; }

[BindProperty]

public decimal HandicapDifferential { get; set; }

public string Message { get; set; }

public void OnGet()

{

}

public void OnPost()

{

PlayerScore NewPlayerScore = new PlayerScore();

NewPlayerScore.MemberNumber = MemberNumber;

NewPlayerScore.Course = Course;

NewPlayerScore.Date = Date;

NewPlayerScore.Rating = Rating;

NewPlayerScore.Slope = Slope;

NewPlayerScore.Hole1 = Hole1;

NewPlayerScore.Hole2 = Hole2;

NewPlayerScore.Hole3 = Hole3;

NewPlayerScore.Hole4 = Hole4;

NewPlayerScore.Hole5 = Hole5;

NewPlayerScore.Hole6 = Hole6;

NewPlayerScore.Hole7 = Hole7;

NewPlayerScore.Hole8 = Hole8;

NewPlayerScore.Hole9 = Hole9;

NewPlayerScore.Hole10 = Hole10;

NewPlayerScore.Hole11 = Hole11;

NewPlayerScore.Hole12 = Hole12;

NewPlayerScore.Hole13 = Hole13;

NewPlayerScore.Hole14 = Hole14;

NewPlayerScore.Hole15 = Hole15;

NewPlayerScore.Hole16 = Hole16;

NewPlayerScore.Hole17 = Hole17;

NewPlayerScore.Hole18 = Hole18;

NewPlayerScore.Score = Hole1 + Hole2 + Hole3 + Hole4 + Hole5 + Hole6 + Hole7 + Hole8 + Hole9 + Hole10 + Hole11 +

Hole12 + Hole13 + Hole14 + Hole15 + Hole16 + Hole17 + Hole18;

NewPlayerScore.HandicapDifferential = (NewPlayerScore.Score - NewPlayerScore.Rating) \* 113 / NewPlayerScore.Slope;

CBGS RequestDirector = new CBGS();

bool Confirmation;

Confirmation = RequestDirector.InsertPlayerScore(NewPlayerScore);

if(Confirmation)

{

Message ="Score is-"+NewPlayerScore.Score+ ";HandicapDiffrential is-" + NewPlayerScore.HandicapDifferential;

}

}

}

}

ViewsPlayerHandicap.cshtml

@page

@model ClubBaistGolfSystem.Pages.ViewsPlayerHandicapModel

@{

}

<!DOCTYPE html>

<html lang="en">

<head>

<title>ViewsPlayerHandicap</title>

<meta charset="utf-8" />

<script type="text/javascript">

</script>

</head>

<body>

<header>ViewsPlayerHandicap</header>

<main>

<div>

<form name="ViewsPlayerHandicap" method="post" id="ViewsPlayerHandicap">

<table>

<tr>

<td>MemberNumber:</td>

<td>

<input id="Text1" type="text" name="MemberNumber" value="@Model.MemberNumber" />

</td>

</tr>

<tr>

<td>

<input type="submit" name="ViewsPlayerHandicap" />

</td>

</tr>

</table>

</form>

@Model.Message

</div>

</main>

<footer>

</footer>

</body>

</html>

ViewsPlayerHandicap.cshtml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using ClubBaistGolfSystem.Domain;

namespace ClubBaistGolfSystem.Pages

{

public class ViewsPlayerHandicapModel : PageModel

{

[BindProperty]

public string MemberNumber { get; set; }

public string Message { get; set; }

public decimal HandicapFactor { get; set; }

public void OnGet()

{

CBGS RequestDirector = new CBGS();

HandicapFactor = RequestDirector.ViewHandicapFactor(MemberNumber);

}

}

}

TechnicalServices:

MemberAccountEntries.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using ClubBaistGolfSystem.Domain;

namespace ClubBaistGolfSystem.TechnicalServices

{

public class MemberAccountEntries

{

public List<MemberAccountEntry> GetAccountEntry(string MemberAccountNumber)

{

SqlConnection Connection = new SqlConnection();

Connection.ConnectionString = @"Persist Security Info=false; Integrated Security = true; Database = ClubBaistGCMS; Server=(localdb)\MSSQLLocalDB";

Connection.Open();

SqlCommand SampleCommand = new SqlCommand

{

Connection = Connection,

CommandType = CommandType.StoredProcedure,

CommandText = "GetMemberAccountEntry"

};

SqlParameter SampleCommandParameter;

SampleCommandParameter = new SqlParameter

{

ParameterName = "@MemberAccountNumber",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = MemberAccountNumber

};

SampleCommand.Parameters.Add(SampleCommandParameter);

SqlDataReader SampleDataReader;

SampleDataReader = SampleCommand.ExecuteReader();

List<MemberAccountEntry> newMemberAccountEntries = new List<MemberAccountEntry>();

if (SampleDataReader.HasRows)

{

while (SampleDataReader.Read())

{

MemberAccountEntry newMemberAccountEntry = new MemberAccountEntry();

newMemberAccountEntry.Amount = Convert.ToDecimal(SampleDataReader["Amount"]);

newMemberAccountEntry.ActivityDate = Convert.ToDateTime(SampleDataReader["ActivityDate"]);

newMemberAccountEntry.PostedDate = Convert.ToDateTime(SampleDataReader["PostedDate"]);

newMemberAccountEntry.Description = (string)SampleDataReader["Description"];

newMemberAccountEntries.Add(newMemberAccountEntry);

}

}

SampleDataReader.Close();

Connection.Close();

return newMemberAccountEntries;

}

}

}

MemberAccounts.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using ClubBaistGolfSystem.Domain;

namespace ClubBaistGolfSystem.TechnicalServices

{

public class MemberAccounts

{

public bool AddMemberAccount(MemberAccount accountDetails)

{

List<MemberAccountEntry> memberAccountEntries = new List<MemberAccountEntry>();

MemberAccountEntry memberAccountEntry = new MemberAccountEntry();

bool Success = false;

SqlConnection connection1 = new SqlConnection();

connection1.ConnectionString =

@"Persist Security Info=False;Integrated Security=True;Database=ClubBaistGCMS;server=(localdb)\MSSQLLocalDB";

connection1.Open();

SqlCommand SampleCommand1 = new SqlCommand();

SampleCommand1.Connection = connection1;

SampleCommand1.CommandType = CommandType.StoredProcedure;

SampleCommand1.CommandText = "AddMemberAccount";

SqlParameter SampleCommandParameter1;

var MemberAccountNumber=1;

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Balance",

SqlDbType = SqlDbType.Decimal,

Direction = ParameterDirection.Input,

SqlValue = accountDetails.Balance

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@MemberAccountNumber",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue =MemberAccountNumber

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Amount",

SqlDbType = SqlDbType.Decimal,

Direction = ParameterDirection.Input,

SqlValue = memberAccountEntry.Amount

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@ActivityDate",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = memberAccountEntry.ActivityDate

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@PostedDate",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = memberAccountEntry.PostedDate

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Description",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = memberAccountEntry.Description

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

memberAccountEntries.Add(memberAccountEntry);

SampleCommand1.ExecuteNonQuery();

connection1.Close();

Success = true;

return Success;

}

public MemberAccount GetMemberAccount(string MemberAccountNumber)

{

List<MemberAccountEntry> newMemberAccountEntries;

MemberAccountEntries MemberAccountEntryManager = new MemberAccountEntries();

newMemberAccountEntries = MemberAccountEntryManager.GetAccountEntry(MemberAccountNumber);

SqlConnection Connection = new SqlConnection();

Connection.ConnectionString = @"Persist Security Info=false; Integrated Security = true; Database = ClubBaistGCMS; Server=(localdb)\MSSQLLocalDB";

Connection.Open();

SqlCommand SampleCommand = new SqlCommand

{

Connection = Connection,

CommandType = CommandType.StoredProcedure,

CommandText = "GetMemberAccount"

};

SqlParameter SampleCommandParameter;

SampleCommandParameter = new SqlParameter

{

ParameterName = "@MemberAccountNumber",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = MemberAccountNumber

};

SampleCommand.Parameters.Add(SampleCommandParameter);

SqlDataReader SampleDataReader;

SampleDataReader = SampleCommand.ExecuteReader();

MemberAccount SelectedAccount = new MemberAccount();

if (SampleDataReader.HasRows)

{

SampleDataReader.Read();

SelectedAccount.Balance = Convert.ToDecimal(SampleDataReader["Balance"]);

}

SelectedAccount.newMemberAccountEntries = newMemberAccountEntries;

SampleDataReader.Close();

Connection.Close();

return SelectedAccount;

}

public List<MemberAccountEntry> GetAccountEntry(string MemberAccountNumber)

{

List<MemberAccountEntry> newMemberAccountEntries;

MemberAccountEntries MemberAccountEntryManager = new MemberAccountEntries();

newMemberAccountEntries = MemberAccountEntryManager.GetAccountEntry(MemberAccountNumber);

return newMemberAccountEntries;

}

}

}

Members.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using ClubBaistGolfSystem.Domain;

namespace ClubBaistGolfSystem.TechnicalServices

{

public class Members

{

public bool AddMember(Member memberDetails)

{

bool Success = false;

SqlConnection connection1 = new SqlConnection();

connection1.ConnectionString =

@"Persist Security Info=False;Integrated Security=True;Database=ClubBaistGCMS;server=(localdb)\MSSQLLocalDB";

connection1.Open();

SqlCommand SampleCommand1 = new SqlCommand();

SampleCommand1.Connection = connection1;

SampleCommand1.CommandType = CommandType.StoredProcedure;

SampleCommand1.CommandText = "AddMember";

SqlParameter SampleCommandParameter1;

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@MemberNumber",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = memberDetails.MemberNumber

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@FirstName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = memberDetails.FirstName

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@LastName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = memberDetails.LastName

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Address",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = memberDetails.Address

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@PostalCode",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = memberDetails.PostalCode

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Phone",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = memberDetails.Phone

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Email",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = memberDetails.Email

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@MemberType",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = memberDetails.MemberType

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommand1.ExecuteNonQuery();

Console.WriteLine("Success excellentquery");

connection1.Close();

Success = true;

return Success;

}

}

}

MembershipApplications.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using ClubBaistGolfSystem.Domain;

using System.Data;

using System.Data.SqlClient;

namespace ClubBaistGolfSystem.TechnicalServices

{

public class MembershipApplications

{

public bool RecordsMembershipApplication(MembershipApplication applicationInformation)

{

bool Success = false;

SqlConnection connection1 = new SqlConnection();

connection1.ConnectionString =

@"Persist Security Info=False;Integrated Security=True;Database=ClubBaistGCMS;server=(localdb)\MSSQLLocalDB";

connection1.Open();

SqlCommand SampleCommand1 = new SqlCommand();

SampleCommand1.Connection = connection1;

SampleCommand1.CommandType = CommandType.StoredProcedure;

SampleCommand1.CommandText = "RecordsMembershipApplication";

SqlParameter SampleCommandParameter1;

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@FirstName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.FirstName

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@LastName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.LastName

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Address",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.Address

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@PostalCode",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.PostalCode

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Phone",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.Phone

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Email",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.Email

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@DateOfBirth",

SqlDbType = SqlDbType.Date,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.DateOfBirth

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Occupation",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.Occupation

};

SampleCommand1.Parameters.Add(SampleCommandParameter1); SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@CompanyName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.CompanyName

};

SampleCommand1.Parameters.Add(SampleCommandParameter1); SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@CompanyAddress",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.CompanyAddress

};

SampleCommand1.Parameters.Add(SampleCommandParameter1); SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@CompanyPostalCode",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.CompanyPostalCode

};

SampleCommand1.Parameters.Add(SampleCommandParameter1); SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@CompanyPhone",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.CompanyPhone

};

SampleCommand1.Parameters.Add(SampleCommandParameter1); SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@ShareholderName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.ShareholderName

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Date",

SqlDbType = SqlDbType.Date,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.Date

};

SampleCommand1.Parameters.Add(SampleCommandParameter1); SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Status",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = applicationInformation.Status

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommand1.ExecuteNonQuery();

Console.WriteLine("Success excellentquery");

connection1.Close();

Success = true;

return Success;

}

public List<MembershipApplication> GetMembershipAplications(string Status)

{

SqlConnection connection = new SqlConnection();

connection.ConnectionString =

@"Persist Security Info=False;Integrated Security=True;Database=ClubBaistGCMS;server=(localdb)\MSSQLLocalDB";

connection.Open();

SqlCommand SampleCommand = new SqlCommand();

SampleCommand.Connection = connection;

SampleCommand.CommandType = CommandType.StoredProcedure;

SampleCommand.CommandText = "GetMembershipApplications";

SqlParameter SampleCommandParameter;

SampleCommandParameter = new SqlParameter

{

ParameterName = "@Status",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

Value = Status

};

SampleCommand.Parameters.Add(SampleCommandParameter);

SqlDataReader SampleDataReader;

SampleDataReader = SampleCommand.ExecuteReader();

List<MembershipApplication> RequestedApplications = new List<MembershipApplication>();

if (SampleDataReader.HasRows)

{

while (SampleDataReader.Read())

{

MembershipApplication OnHoldApplication = new MembershipApplication();

OnHoldApplication.MemberApplicationNumber = SampleDataReader["MemberApplicationNumber"].ToString();

OnHoldApplication.FirstName = SampleDataReader["FirstName"].ToString();

OnHoldApplication.LastName = SampleDataReader["LastName"].ToString();

OnHoldApplication.Status = SampleDataReader["Status"].ToString();

RequestedApplications.Add(OnHoldApplication);

}

}

SampleDataReader.Close();

connection.Close();

return RequestedApplications;

}

public bool UpdateMembershipApplication(string MemberApplicationNumber, MembershipApplication newMembershipApplication)

{

bool Success;

SqlConnection connection1 = new SqlConnection();

connection1.ConnectionString =

@"Persist Security Info=False;Integrated Security=True;Database=ClubBaistGCMS;server=(localdb)\MSSQLLocalDB";

connection1.Open();

SqlCommand SampleCommand1 = new SqlCommand();

SampleCommand1.Connection = connection1;

SampleCommand1.CommandType = CommandType.StoredProcedure;

SampleCommand1.CommandText = "UpdateMembershipApplication";

SqlParameter SampleCommandParameter1;

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@MemberApplicationNumber",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = MemberApplicationNumber

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@FirstName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = newMembershipApplication.FirstName

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@LastName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = newMembershipApplication.LastName

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Address",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = newMembershipApplication.Address

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@PostalCode",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = newMembershipApplication.PostalCode

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Phone",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = newMembershipApplication.Phone

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Email",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = newMembershipApplication.Email

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Status",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = newMembershipApplication.Status

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommand1.ExecuteNonQuery();

Console.WriteLine("Success excellentquery");

connection1.Close();

Success = true;

return Success;

}

}

}

PlayerScores.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using ClubBaistGolfSystem.Domain;

namespace ClubBaistGolfSystem.TechnicalServices

{

public class PlayerScores

{

public bool AddPlayerScore(PlayerScore NewPlayerScore)

{

bool Success = false;

SqlConnection connection1 = new SqlConnection();

connection1.ConnectionString =

@"Persist Security Info=False;Integrated Security=True;Database=ClubBaistGCMS;server=(localdb)\MSSQLLocalDB";

connection1.Open();

SqlCommand SampleCommand1 = new SqlCommand();

SampleCommand1.Connection = connection1;

SampleCommand1.CommandType = CommandType.StoredProcedure;

SampleCommand1.CommandText = "AddPlayerScore";

SqlParameter SampleCommandParameter1;

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@MemberNumber",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.MemberNumber

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Course",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Course

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Date",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Date

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Rating",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Rating

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Slope",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Slope

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole1",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole1

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole2",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole2

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole3",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole3

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole4",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole4

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole5",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole5

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole6",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole6

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole7",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole7

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole8",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole8

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole9",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole9

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole10",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole10

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole11",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole11

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole12",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole12

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole13",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole13

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole14",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole14

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole15",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole15

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole16",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole16

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole17",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole17

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Hole18",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Hole18

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Score",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.Score

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@HandicapDifferential",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = NewPlayerScore.HandicapDifferential

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommand1.ExecuteNonQuery();

Console.WriteLine("Success excellentquery");

connection1.Close();

Success = true;

return Success;

public double GetHandicapFactor(string MemberNumber)

{

SqlConnection Connection = new SqlConnection();

Connection.ConnectionString = @"Persist Security Info=False; Integrated Security = True; Database = ClubBaistGCMS; Server = (localdb)\MSSQLLocalDB";

Connection.Open();

var HandicapFactor=1.0;

SqlCommand SampleCommand = new SqlCommand

{

Connection = Connection,

CommandType = CommandType.StoredProcedure,

CommandText = "GetHandicapFactor"

};

SqlParameter SampleCommandParameter = new SqlParameter

{

ParameterName = "@MemberNumber",

SqlDbType = SqlDbType.Int,

Direction = ParameterDirection.Input,

SqlValue = MemberNumber

};

SampleCommand.Parameters.Add(SampleCommandParameter);

SqlDataReader SampleDataReader;

SampleDataReader = SampleCommand.ExecuteReader();

if (SampleDataReader.HasRows)

{

SampleDataReader.Read();

HandicapFactor= Convert.ToDouble(SampleDataReader["HandicapFacor"]);

}

SampleDataReader.Close();

Connection.Close();

return HandicapFactor;

} }

}

StandingTeeTimeRequest.cs

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

using ClubBaistGolfSystem.Domain;

namespace ClubBaistGolfSystem.TechnicalServices

{

class StandingTeeTimeRequests

{

public List<StandingTeeTime> GetStandingTeeTime(DateTime RequestedStartDate)

{

SqlConnection connection2 = new SqlConnection();

connection2.ConnectionString =

@"Persist Security Info=False;Integrated Security=True;Database=ClubBaistGCMS;server=(localdb)\MSSQLLocalDB";

connection2.Open();

SqlCommand SampleCommand2 = new SqlCommand();

SampleCommand2.Connection = connection2;

SampleCommand2.CommandType = CommandType.StoredProcedure;

SampleCommand2.CommandText = "GetStandingTeeTime";

SqlParameter SampleCommandParameter2;

SampleCommandParameter2 = new SqlParameter

{

ParameterName = "@RequestedStartDate",

SqlDbType = SqlDbType.Date,

Direction = ParameterDirection.Input,

Value = RequestedStartDate

};

SampleCommand2.Parameters.Add(SampleCommandParameter2);

SqlDataReader SampleDataReader;

SampleDataReader = SampleCommand2.ExecuteReader();

List<StandingTeeTime> StandingTeeTimes = new List<StandingTeeTime>();

if (SampleDataReader.HasRows)

{

while (SampleDataReader.Read())

{

StandingTeeTime RequestedStandingTeeTime = new StandingTeeTime();

RequestedStandingTeeTime.RequestedStartDate = Convert.ToDateTime(SampleDataReader["RequestedStartDate"].ToString()).ToShortDateString();

RequestedStandingTeeTime.MemberNumber = SampleDataReader["MemberNumber"].ToString();

RequestedStandingTeeTime.MemberFirstName = SampleDataReader["MemberFirstName"].ToString();

RequestedStandingTeeTime.MemberLastName = SampleDataReader["MemberLastName"].ToString();

RequestedStandingTeeTime.DayOfWeek = SampleDataReader["DayOfWeek"].ToString();

RequestedStandingTeeTime.RequestedTeeTime = SampleDataReader["RequestedTeeTime"].ToString();

RequestedStandingTeeTime.RequestedStartDate = SampleDataReader["RequestedStartDate"].ToString();

RequestedStandingTeeTime.RequestedEndDate = SampleDataReader["RequestedEndDate"].ToString();

StandingTeeTimes.Add(RequestedStandingTeeTime);

}

}

SampleDataReader.Close();

connection2.Close();

return StandingTeeTimes;

}

public bool AddStandingTeeTime(StandingTeeTime RequestedStandingTeeTime)

{

bool Success = false;

SqlConnection connection3 = new SqlConnection();

connection3.ConnectionString =

@"Persist Security Info = false; Integrated Security = true; Database=ClubBaistGCMS;server=(localdb)\MSSQLLocalDB";

connection3.Open();

SqlCommand SampleCommand3 = new SqlCommand();

SampleCommand3.Connection = connection3;

SampleCommand3.CommandType = CommandType.StoredProcedure;

SampleCommand3.CommandText = "AddStandingTeeTime";

SqlParameter SampleCommandParameter3;

SampleCommandParameter3 = new SqlParameter()

{

ParameterName = "@MemberNumber",

SqlDbType = SqlDbType.Int,

Direction = ParameterDirection.Input,

Value = RequestedStandingTeeTime.MemberNumber

};

SampleCommand3.Parameters.Add(SampleCommandParameter3);

SampleCommandParameter3 = new SqlParameter()

{

ParameterName = "@MemberFirstName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

Value = RequestedStandingTeeTime.MemberFirstName

};

SampleCommand3.Parameters.Add(SampleCommandParameter3);

SampleCommandParameter3 = new SqlParameter()

{

ParameterName = "@MemberLastName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

Value = RequestedStandingTeeTime.MemberLastName

};

SampleCommand3.Parameters.Add(SampleCommandParameter3);

SampleCommandParameter3 = new SqlParameter()

{

ParameterName = "@DayOfWeek",

SqlDbType = SqlDbType.Int,

Direction = ParameterDirection.Input,

Value = RequestedStandingTeeTime.DayOfWeek

};

SampleCommand3.Parameters.Add(SampleCommandParameter3);

SampleCommandParameter3 = new SqlParameter()

{

ParameterName = "@RequestedTeeTime",

SqlDbType = SqlDbType.DateTime,

Direction = ParameterDirection.Input,

Value = RequestedStandingTeeTime.RequestedTeeTime

};

SampleCommand3.Parameters.Add(SampleCommandParameter3);

SampleCommandParameter3 = new SqlParameter()

{

ParameterName = "@RequestedStartDate",

SqlDbType = SqlDbType.Date,

Direction = ParameterDirection.Input,

Value = RequestedStandingTeeTime.RequestedStartDate

};

SampleCommand3.Parameters.Add(SampleCommandParameter3);

SampleCommandParameter3 = new SqlParameter()

{

ParameterName = "@RequestedEndDate",

SqlDbType = SqlDbType.Date,

Direction = ParameterDirection.Input,

Value = RequestedStandingTeeTime.RequestedEndDate

};

SampleCommand3.Parameters.Add(SampleCommandParameter3);

SampleCommand3.ExecuteNonQuery();

Console.WriteLine("Excellent Non Query");

connection3.Close();

Success = true;

return Success;

}

public StandingTeeTime SearchStandingTeeTime(string MemberNumber)

{

SqlConnection Connection = new SqlConnection();

Connection.ConnectionString = @"Persist Security Info=False; Integrated Security = True; Database = ClubBaistGCMS; Server = (localdb)\MSSQLLocalDB";

Connection.Open();

SqlCommand SampleCommand = new SqlCommand

{

Connection = Connection,

CommandType = CommandType.StoredProcedure,

CommandText = "SearchStandingTeeTime"

};

SqlParameter SampleCommandParameter = new SqlParameter

{

ParameterName = "@MemberNumber",

SqlDbType = SqlDbType.Int,

Direction = ParameterDirection.Input,

SqlValue = MemberNumber

};

SampleCommand.Parameters.Add(SampleCommandParameter);

SqlDataReader SampleDataReader;

SampleDataReader = SampleCommand.ExecuteReader();

StandingTeeTime SearchedStandingTeeTime= new StandingTeeTime();

if (SampleDataReader.HasRows)

{

SampleDataReader.Read();

SearchedStandingTeeTime.MemberFirstName =(string)SampleDataReader["MemberFirstName"];

SearchedStandingTeeTime.MemberLastName = (string)SampleDataReader["MemberLastName"];

}

SampleDataReader.Close();

Connection.Close();

return SearchedStandingTeeTime;

}

public bool DeleteStandingTeeTimeRequest(string MemberNumber)

{

bool Success = false;

SqlConnection SampleConnection = new SqlConnection();

SampleConnection.ConnectionString = @"Persist Security Info=false;Integrated Security=true;Database=ClubBaistGCMS;Server=(localdb)\MSSqlLocalDb";

SampleConnection.Open();

SqlCommand SampleCommand = new SqlCommand

{

Connection = SampleConnection,

CommandType = CommandType.StoredProcedure,

CommandText = "DeleteStandingTeeTimeRequest"

};

SqlParameter SampleCommandParameter = new SqlParameter

{

ParameterName = "@MemberNumber",

SqlDbType = SqlDbType.Int,

Direction = ParameterDirection.Input,

SqlValue = MemberNumber

};

SampleCommand.Parameters.Add(SampleCommandParameter);

SampleCommand.ExecuteNonQuery();

SampleConnection.Close();

Success = true;

return Success;

}

}

}

TeeSheets.cs

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

using ClubBaistGolfSystem.Domain;

namespace ClubBaistGolfSystem.TechnicalServices

{

class TeeSheet

{

public List<TeeTime> GetTeeSheet(DateTime Date)

{

SqlConnection connection = new SqlConnection();

connection.ConnectionString =

@"Persist Security Info=False;Integrated Security=True;Database=ClubBaistGCMS;server=(localdb)\MSSQLLocalDB";

connection.Open();

SqlCommand SampleCommand = new SqlCommand();

SampleCommand.Connection = connection;

SampleCommand.CommandType = CommandType.StoredProcedure;

SampleCommand.CommandText = "GetTeeSheet";

SqlParameter SampleCommandParameter;

SampleCommandParameter = new SqlParameter

{

ParameterName = "@Date",

SqlDbType = SqlDbType.Date,

Direction = ParameterDirection.Input,

Value = Date

};

SampleCommand.Parameters.Add(SampleCommandParameter);

SqlDataReader SampleDataReader;

SampleDataReader = SampleCommand.ExecuteReader();

List<TeeTime> RequestedTeeSheet = new List<TeeTime>();

if (SampleDataReader.HasRows)

{

while (SampleDataReader.Read())

{

TeeTime RequestedTeeTime = new TeeTime();

RequestedTeeTime.Date = Convert.ToDateTime(SampleDataReader["Date"].ToString()).ToShortDateString();

RequestedTeeTime.Time = SampleDataReader["Time"].ToString();

RequestedTeeTime.NumberOfCarts = SampleDataReader["NumberOfCarts"].ToString();

RequestedTeeTime.NumberOfPlayers = SampleDataReader["NumberOfPlayers"].ToString();

RequestedTeeTime.PlayerFirstName = SampleDataReader["PlayerFirstName"].ToString();

RequestedTeeTime.PlayerLastName = SampleDataReader["PlayerLastName"].ToString();

RequestedTeeTime.Phone= SampleDataReader["Phone"].ToString();

RequestedTeeSheet.Add(RequestedTeeTime);

}

}

SampleDataReader.Close();

connection.Close();

return RequestedTeeSheet;

}

public bool AddTeeTime(TeeTime selectedTeeTime)

{

bool Success = false;

SqlConnection connection1 = new SqlConnection();

connection1.ConnectionString =

@"Persist Security Info=False;Integrated Security=True;Database=ClubBaistGCMS;server=(localdb)\MSSQLLocalDB";

connection1.Open();

SqlCommand SampleCommand1 = new SqlCommand();

SampleCommand1.Connection = connection1;

SampleCommand1.CommandType = CommandType.StoredProcedure;

SampleCommand1.CommandText = "AddTeeTime";

SqlParameter SampleCommandParameter1;

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Date",

SqlDbType = SqlDbType.DateTime,

Direction = ParameterDirection.Input,

SqlValue = selectedTeeTime.Date

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Time",

SqlDbType = SqlDbType.DateTime,

Direction = ParameterDirection.Input,

SqlValue = selectedTeeTime.Time

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@NumberOfCarts",

SqlDbType = SqlDbType.Int,

Direction = ParameterDirection.Input,

SqlValue = selectedTeeTime.NumberOfCarts

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@NumberOfPlayers",

SqlDbType = SqlDbType.Int,

Direction = ParameterDirection.Input,

SqlValue = selectedTeeTime.NumberOfPlayers

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@PlayerFirstName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = selectedTeeTime.PlayerFirstName

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@PlayerLastName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = selectedTeeTime.PlayerLastName

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Phone",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = selectedTeeTime.Phone

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommand1.ExecuteNonQuery();

Console.WriteLine("Success excellentquery");

connection1.Close();

Success = true;

return Success;

}

public TeeTime GetTeeTime(DateTime Date, DateTime Time)

{

SqlConnection Connection = new SqlConnection();

Connection.ConnectionString = @"Persist Security Info=false; Integrated Security = true; Database = ClubBaistGCMS; Server=(localdb)\MSSQLLocalDB";

Connection.Open();

SqlCommand SampleCommand = new SqlCommand

{

Connection = Connection,

CommandType = CommandType.StoredProcedure,

CommandText = "GetTeeTime"

};

SqlParameter SampleCommandParameter;

SampleCommandParameter = new SqlParameter

{

ParameterName = "@Date",

SqlDbType = SqlDbType.DateTime,

Direction = ParameterDirection.Input,

SqlValue = Date

};

SampleCommand.Parameters.Add(SampleCommandParameter);

SampleCommandParameter = new SqlParameter

{

ParameterName = "@Time",

SqlDbType = SqlDbType.DateTime,

Direction = ParameterDirection.Input,

SqlValue = Time

};

SampleCommand.Parameters.Add(SampleCommandParameter);

SqlDataReader SampleDataReader;

SampleDataReader = SampleCommand.ExecuteReader();

TeeTime SelectedTime = new TeeTime();

if (SampleDataReader.HasRows)

{

SampleDataReader.Read();

SelectedTime.PlayerFirstName = (string)SampleDataReader["PlayerFirstName"];

SelectedTime.PlayerLastName = (string)SampleDataReader["PlayerLastName"];

SelectedTime.Phone = (string)SampleDataReader["Phone"];

}

SampleDataReader.Close();

Connection.Close();

return SelectedTime;

}

public bool UpdateTeeTime(DateTime Date,DateTime Time, TeeTime selectedTime)

{

bool Success;

SqlConnection connection1 = new SqlConnection();

connection1.ConnectionString =

@"Persist Security Info=False;Integrated Security=True;Database=ClubBaistGCMS;server=(localdb)\MSSQLLocalDB";

connection1.Open();

SqlCommand SampleCommand1 = new SqlCommand();

SampleCommand1.Connection = connection1;

SampleCommand1.CommandType = CommandType.StoredProcedure;

SampleCommand1.CommandText = "UpdateTeeTime";

SqlParameter SampleCommandParameter1;

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Date",

SqlDbType = SqlDbType.DateTime,

Direction = ParameterDirection.Input,

SqlValue = Date

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Time",

SqlDbType = SqlDbType.DateTime,

Direction = ParameterDirection.Input,

SqlValue = Time

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@PlayerFirstName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = selectedTime.PlayerFirstName

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@PlayerLastName",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = selectedTime.PlayerLastName

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommandParameter1 = new SqlParameter

{

ParameterName = "@Phone",

SqlDbType = SqlDbType.VarChar,

Direction = ParameterDirection.Input,

SqlValue = selectedTime.Phone

};

SampleCommand1.Parameters.Add(SampleCommandParameter1);

SampleCommand1.ExecuteNonQuery();

Console.WriteLine("Success excellentquery");

connection1.Close();

Success = true;

return Success;

}

}

}