|  |  |
| --- | --- |
| Name: | Lohar Hamid A. |
| Roll No: | 111 |
| Class: | SYBCA-B SEM4 |
| Subject: | .Net Programming |
| Project Title: | Cricket Score Counter |
|  |  |

**S.D. Jain International College**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Title** | **Page No.** |
| 1 | Introduction | 3 |
|  | 1.1 Project Summary | 3 |
|  | 1.2 Technical Details | 4 |
| 2 | Designing | 5 |
|  | 2.1 Database Design | 5 |
|  | 2.2 Database Relationship | 7 |
|  | 2.3 User Interface | 8 |

**1. Introduction**

Cricket Score Counter is application which can be used to count cricket score. We can record cricket match score using this application and then we can view summary of all match which we have already recorded.

**1.1Project Summary**

Cricket Score Counter is application which count cricket score and it provide live score board of running cricket match on screen. This application can be useful for scorer to record and maintain score of match. Cricket Counter records each ball and store it into database that means we can refer to any match at any time. With further enhancement this application can be used to display score card on large screen during match this score card is useful for audience and players too. Cricket Score Counter provides various kinds of reports of all match recorded. It provide following types of summary of match.

1. **Batsman Summary**: Here you can see all details about batsman’s performances in match. It shows Which Batsman made how many runs in how many balls and how many 4s and 6s of the batsman.
2. **Bowler Summary:** It shows performance of all bowlers that which bowler played how many over and gave how many runs and how many wickets taken by bowler. It also shows Economy of Bowler.
3. **Wicket Summary:** It shows details about Fall of Wickets.
4. **Over Summary:** It shows Summary of all over.

**Team Management:** Cricket Score Counter provides team management functionality. You can manage many teams in cricket counter. You can use that team further in any match. You get powerful team management where you can add players delete players and you can also set if player is ‘Batsman’ or ‘Bowler’.

**1.2 Technical Details**

|  |  |
| --- | --- |
| **Project Title** | Cricket Score Counter |
| **Project Objective** | Record Cricket Match Score |
| **Name Of College** | SDJ International College |
| **Front-End Tool** | VB.NET |
| **Back-End Tool** | Ms Access Database with using ADO.NET |
| **Operating System** | Any Windows OS with .Net Framework 3.5 or later |
| **Internal Guide** | Nehal Patel |
| **Submitted By** | Lohar Hamid A |

**Project Profile**

**2. Designing**

Designing will include database designing and User interface designing.

**2.1 Database Design**

**Teams**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Description** | **Constraints** |
| Id | AutoNumber | Unique Id for Teams | Primary Key |
| team\_name | Text | Name of Team | - |

**Players**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Description** | **Constraints** |
| Id | AutoNumber | Unique ID of Players | Primary Key |
| Player\_name | Text | Name of Player | - |
| Player\_type | Text | Type of Player Either Batsman,Bowler or AllRounder | - |
| Team\_id | Number | Reference of Team | Foreign Key references Teams(id) |

**Matchs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Description** | **Constraints** |
| Id | AutoNumber | Unique ID of Match | Primary Key |
| Team1 | Number | Team one who is playing | Foreign Key references Teams(id) |
| Team2 | Number | Team two who is playing | Foreign Key references Teams(id) |
| Match\_date | Date | Date on which match is recorded | - |
| Overs | Number | Number of Overs of match | - |
| Team\_win | Number | Team Id of winner team | - |

**Match\_log**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Description** | **Constraints** |
| Id | AutoNumber | Unique ID of Match records | Primary Key |
| Match\_id | Number | Reference to Match | Foreign Key references Match(id) |
| Ball\_no | Number | It is used to refer to ball no of match | - |
| Ball\_type | Text | Refer to if ball is normal ball or wide ball | - |
| Player\_bat1 | Number | Refer to On strike batsman | Foreign Key references Players(id) |
| Player\_bat2 | Number | Refer to off strike batsman | Foreign Key references Players(id) |
| Player\_ball | Number | Refer to Bowler | Foreign Key references Players(id) |
| Score | Number | Score Achieved on ball | - |
| Wicket | Number | To store wicket details | - |

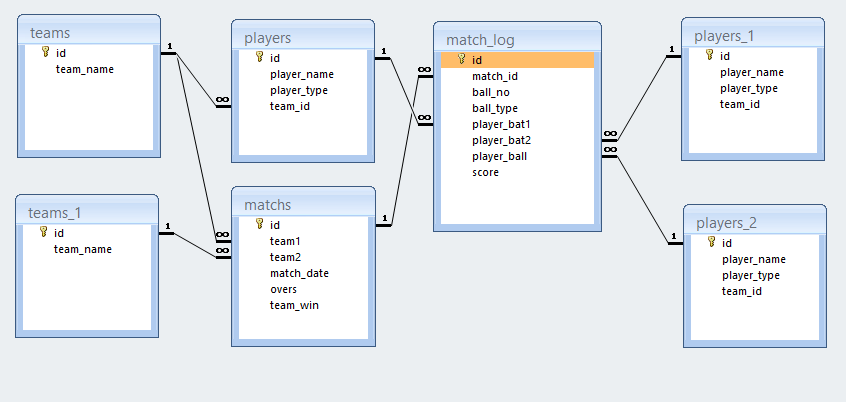
**status**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Description** | **Constraints** |
| Status\_name | Text | Variable name for status | Unique |
| Status\_value | Text | Value of status variable | - |

**Wicket\_type**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Description** | **Constraints** |
| Wicket\_num | Number | Wicket Number | - |
| Wicket\_value | Text | To indicate wicket type text | - |

**2.2 Database Relationship**

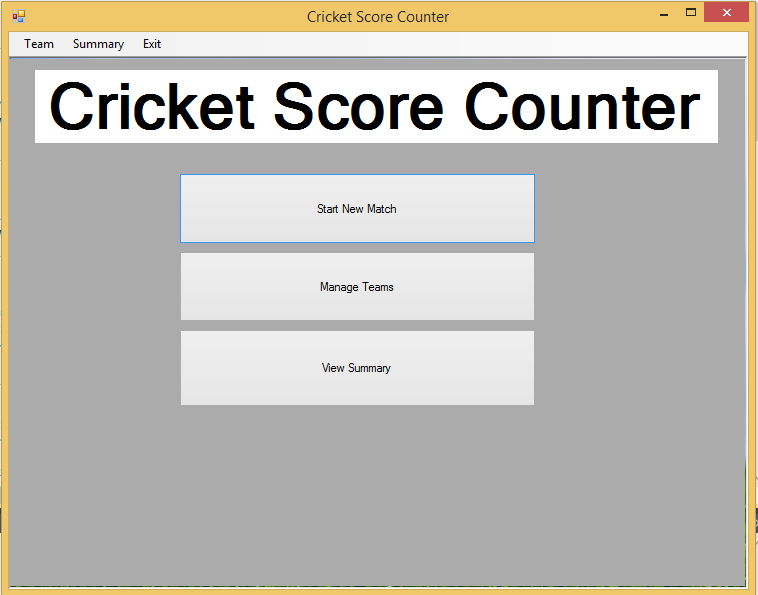


**2.3 User Interface**

Here is list of forms included in Cricket Score Counter.

1. **Main Form**

This is main form which is displayed first on opening application.



**Codes**

Public Class MainForm

Private Sub DashboardToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

End Sub

Private Sub OpenToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Dim frm As New frmDashboard()

frm.MdiParent = Me

frm.Show()

End Sub

Private Sub ManageTeamToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ManageTeamToolStripMenuItem.Click

Dim frm As New frmTeamManagement()

'frm.MdiParent = Me

frm.ShowDialog()

End Sub

Private Sub MainForm\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

MatchStatus.init()

If (MatchStatus.Status("match\_now") <> "-1") Then

btnStartNewMatch.Text = "Continue Counting"

End If

End Sub

Private Sub btnStartNewMatch\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnStartNewMatch.Click

If (MatchStatus.Status("match\_now") <> "-1") Then

panelHomeButtons.Visible = False

Dim frm2 As New frmDashboard()

frm2.MdiParent = Me

frm2.Show()

'panelHomeButtons.Visible = True

Else

Dim frm As New frmStartMatch

If (frm.ShowDialog() = Windows.Forms.DialogResult.OK) Then

panelHomeButtons.Visible = False

Dim frm2 As New frmDashboard()

frm2.MdiParent = Me

frm2.Show()

'panelHomeButtons.Visible = True

End If

End If

End Sub

Private Sub BowlerSummaryToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BowlerSummaryToolStripMenuItem.Click

Dim frm As New frmBowlerSummary()

frm.ShowDialog()

End Sub

Private Sub BatsmanSummaryToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BatsmanSummaryToolStripMenuItem.Click

Dim frm As New frmBatsmanSummary()

frm.ShowDialog()

End Sub

Private Sub WicketSummaryToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles WicketSummaryToolStripMenuItem.Click

Dim frm As New frmWicketSummary()

frm.ShowDialog()

End Sub

Private Sub btnManageTeams\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnManageTeams.Click

Dim frm As New frmTeamManagement()

frm.ShowDialog()

End Sub

Private Sub btnViewSummary\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnViewSummary.Click

Dim frm As New frmSummary()

frm.Show()

End Sub

Private Sub ExitToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ExitToolStripMenuItem.Click

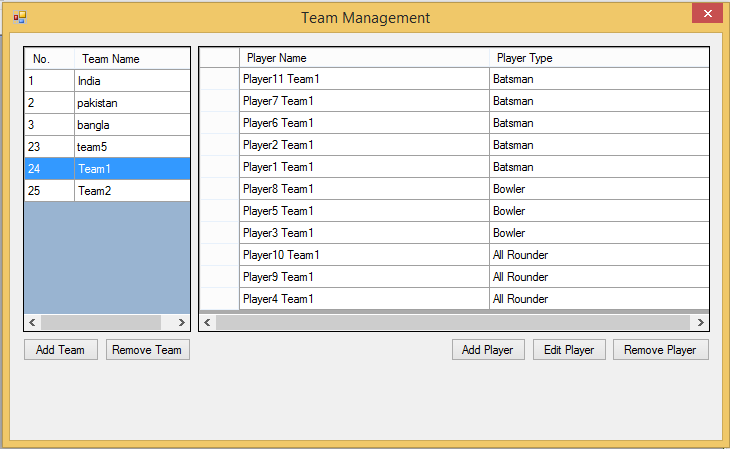
Me.Close()

End Sub

End Class

1. **Team management**

In this form we can add and remove teams. And we can view all players of team. We can add players in particular team and we can remove player from team and we can also update player’s details.



**Codes**

Imports System.Data.OleDb

Imports WindowsApplication1.DbConnection

Public Class frmTeamManagement

Dim dsTeam As New DataSet()

Dim dbc As New DbConnection()

Dim daTeams As OleDbDataAdapter

Private Sub GetTeamFromUser()

Dim myNum As String

Try

myNum = InputBox("Please Enter Team Name: ")

dbc.InsertIntoTeams(myNum)

loadTeams()

Catch e As Exception

MsgBox(e.Message())

End Try

End Sub

Private Sub loadTeams()

Try

dsTeam.Tables("teams").Clear()

Catch ex As Exception

End Try

daTeams.Fill(dsTeam, "teams")

dgvTeams.DataSource = dsTeam.Tables("teams")

dgvTeams.ClearSelection()

dgvTeams.Columns(1).MinimumWidth = 130

dgvTeams.Columns(0).MinimumWidth = 50

dgvTeams.Columns(0).Width = 50

dgvTeams.Columns(1).HeaderText = "Team Name"

dgvTeams.Columns(0).HeaderText = "No."

End Sub

Sub loadPlayers(ByVal team\_id As String)

Try

dsTeam.Tables("players").Clear()

Catch ex As Exception

End Try

dbc.FillDs("select \* from players where team\_id=" + team\_id + " order by player\_type desc", dsTeam, "players")

dgvPlayers.DataSource = dsTeam.Tables("players")

dgvPlayers.Columns(3).Visible = False

dgvPlayers.Columns(0).Visible = False

dgvPlayers.Columns(1).MinimumWidth = 250

dgvPlayers.Columns(2).MinimumWidth = 220

dgvPlayers.Columns(0).HeaderText = "ID"

dgvPlayers.Columns(1).HeaderText = "Player Name"

dgvPlayers.Columns(2).HeaderText = "Player Type"

dgvPlayers.ClearSelection()

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnInsertTeam.Click

GetTeamFromUser()

End Sub

Private Sub TeamManagement\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

daTeams = dbc.RunSql("select id,team\_name as TeamName from teams")

loadTeams()

End Sub

Private Sub dgvTeams\_CellClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles dgvTeams.CellClick

Try

dgvTeams.Rows(e.RowIndex).Selected = True

loadPlayers(dgvTeams.Rows(e.RowIndex).Cells("id").Value)

Catch ex As Exception

'MsgBox(ex.Message)

End Try

End Sub

Private Sub dgvPlayers\_CellClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles dgvPlayers.CellClick

Try

sender.Rows(e.RowIndex).Selected = True

Catch ex As Exception

'MsgBox(ex.Message)

End Try

End Sub

Private Sub TeamManagement\_Shown(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Shown

dgvTeams.ClearSelection()

dgvTeams.Columns("TeamName").Width = 200

dgvTeams.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.None

dgvTeams.AllowUserToResizeRows = False

End Sub

Private Sub btnRemovePlayer\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRemovePlayer.Click

If (dgvPlayers.SelectedRows.Count <> 1) Then

MsgBox("Please Select Player")

Return

End If

Dim rowNum = dgvPlayers.CurrentRow.Cells("id").Value

Dim teamId = dgvPlayers.CurrentRow.Cells("team\_id").Value

Dim res As Integer = dbc.RemoveFromPlayers(rowNum)

If (res = 1) Then

loadPlayers(teamId)

End If

End Sub

Private Sub btnRemoveTeam\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRemoveTeam.Click

Try

If (dgvTeams.SelectedRows.Count <= 0) Then

MsgBox("Please Select Team To Remove")

Return

End If

Dim rowNum = dgvTeams.CurrentRow.Cells("id").Value

dbc.DeleteFromTeams(rowNum)

loadTeams()

Catch ex As Exception

'MsgBox(ex.Message)

End Try

End Sub

Private Sub btnInsertPlayer\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnInsertPlayer.Click

If (dgvTeams.SelectedRows.Count <> 1) Then

MsgBox("Please Select One Team")

Return

End If

If (dgvPlayers.Rows.Count >= 11) Then

MsgBox("One Team Can only have 11 Players")

Return

End If

Dim rowNum = dgvTeams.CurrentRow.Cells("id").Value

Dim dialog As New dialogInsertPlayer(rowNum)

If (dialog.ShowDialog() = Windows.Forms.DialogResult.OK) Then

loadPlayers(rowNum)

End If

End Sub

Private Sub dgvPlayers\_RowsAdded(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewRowsAddedEventArgs) Handles dgvPlayers.RowsAdded

For i As Integer = e.RowIndex To e.RowIndex + e.RowCount - 1

dgvPlayers.Rows(i).Cells.Item("player\_type").Value = CricketCounter.getPlayerTypeString(dgvPlayers.Rows(i).Cells.Item("player\_type").Value)

Next

End Sub

Private Sub btnEditPlayer\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEditPlayer.Click

If (dgvPlayers.SelectedRows.Count <> 1) Then

MsgBox("Please Select Player")

Return

End If

Dim rowNum = dgvPlayers.CurrentRow.Cells("id").Value

Dim dialog As New dialogEditPlayer(dgvPlayers.CurrentRow.Cells("team\_id").Value, rowNum, dgvPlayers.CurrentRow.Cells("player\_name").Value, dgvPlayers.CurrentRow.Cells("player\_type").Value)

If (dialog.ShowDialog() = Windows.Forms.DialogResult.OK) Then

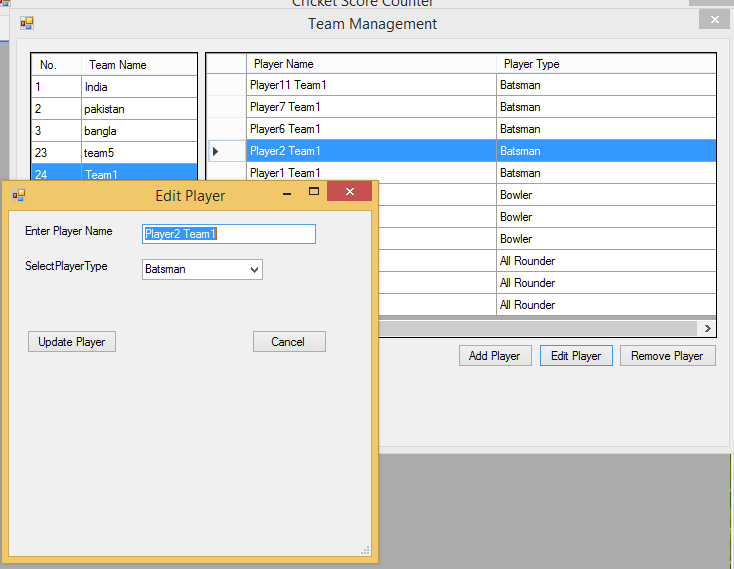
loadPlayers(dgvPlayers.CurrentRow.Cells("team\_id").Value)

End If

End Sub

End Class

**2.1 Edit Player**



**Codes**

Public Class dialogEditPlayer

Dim currentTeam As String

Dim playerName As String

Dim playerType As String

Dim playerId As String

Dim dbc As New DbConnection()

Public Sub New(ByVal teamId As String, ByVal playerId As String, ByVal playerName As String, ByVal playerType As String)

currentTeam = teamId

Me.playerName = playerName

Me.playerId = playerId

Me.playerType = playerType

' This call is required by the designer.

InitializeComponent()

' Add any initialization after the InitializeComponent() call.

End Sub

Private Sub dialogEditPlayer\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

tbPlayerName.Text = playerName

If (playerType = "Batsman") Then

comboPlayerType.SelectedIndex = 0

ElseIf (playerType = "Bowler") Then

comboPlayerType.SelectedIndex = 1

Else

comboPlayerType.SelectedIndex = 2

End If

End Sub

Private Sub btnEditPlayerUpdate\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEditPlayerUpdate.Click

Dim playerName As String = tbPlayerName.Text

Dim pt(2) As String

pt(0) = "bat"

pt(1) = "ball"

pt(2) = "all"

Dim playerType As String = pt(comboPlayerType.SelectedIndex)

'MsgBox(currentTeam + " " + playerName + " " + playerType + " " + playerId)

Dim res As Integer = dbc.UpdatePlayers(playerId, playerName, playerType, currentTeam)

If (res = 1) Then

Me.DialogResult = Windows.Forms.DialogResult.OK

End If

End Sub

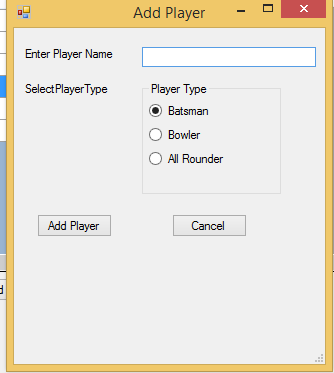
Private Sub btnEditPlayerCancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEditPlayerCancel.Click

Me.DialogResult = Windows.Forms.DialogResult.Cancel

End Sub

End Class

* 1. **Add Player**



**Codes**

Public Class dialogInsertPlayer

Dim dbc As New DbConnection()

Dim currentTeam As String

Public Sub New(ByVal teamId As String)

currentTeam = teamId

' This call is required by the designer.

InitializeComponent()

' Add any initialization after the InitializeComponent() call.

End Sub

Private Sub btnDialogInsertPlayer\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDialogInsertPlayer.Click

Dim playerName As String = tbPlayerName.Text

Dim playerType As String = "bat"

If (rbBwoler.Checked) Then

playerType = "ball"

ElseIf (rbAllRounder.Checked) Then

playerType = "all"

End If

'MsgBox(currentTeam + " " + playerName + " " + playerType)

If (String.IsNullOrEmpty(playerName)) Then

MsgBox("Enter Player Name First")

Return

End If

Dim res As Integer = dbc.InsertIntoPlayers(playerName, playerType, currentTeam)

If (res = 1) Then

Me.DialogResult = Windows.Forms.DialogResult.OK

End If

End Sub

Private Sub btnInsertPlayerCancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnInsertPlayerCancel.Click

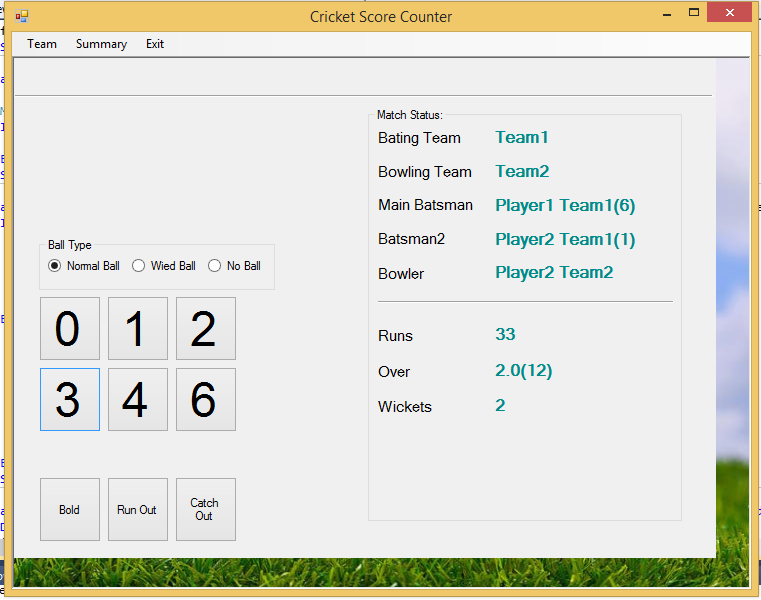
Me.DialogResult = Windows.Forms.DialogResult.Cancel

End Sub

End Class

1. **Match Dashboard**

This form contain main dashboard where score will be counted. Here buttons are given for counting score. Click on button of score and score will be added to scoreboard and scoreboard will be updated. You can select options from ball type that if ball is wide or no ball then you can select that options.



**Codes**

Public Class frmDashboard

Dim dbc As New DbConnection()

Dim ds As New DataSet

Dim isWicket As Boolean

Private Sub frmDashboard\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

MatchStatus.update()

End Sub

Private Sub frmDashboard\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

formLoad()

End Sub

Sub formLoad()

dbc.FillDs("select \* from matchs where id=" + MatchStatus.Status("match\_now"), ds, "matchs")

If (MatchStatus.Status("inning") = "2") Then

MatchStatus.Status("team1") = ds.Tables("matchs").Rows(0)("team2")

MatchStatus.Status("team2") = ds.Tables("matchs").Rows(0)("team1")

End If

dbc.FillDs("select \* from teams where id in(" + MatchStatus.Status("team1") + "," + MatchStatus.Status("team2") + ")", ds, "teams")

dbc.FillDs("select \* from players where team\_id in (" + MatchStatus.Status("team1") + "," + MatchStatus.Status("team2") + ")", ds, "players")

ReloadMatchStatus()

End Sub

Sub ReloadMatchStatus()

Dim wicketCount As String = dbc.getWicketCount(MatchStatus.Status("match\_now"), "")

Dim ballno As String = MatchStatus.Status("ball\_now")

If (Val(ballno) >= Val(MatchStatus.Status("overs\_now")) \* 6 Or Val(wicketCount) >= 9) Then

If (MatchStatus.Status("inning") = "1") Then

Dim frm As New dialogChangeInning()

If (frm.ShowDialog() = Windows.Forms.DialogResult.OK) Then

formLoad()

End If

ElseIf (MatchStatus.Status("inning") = "2") Then

Dim bno1 As String = (Val(MatchStatus.Status("inning")) \* Val(MatchStatus.Status("overs\_now")) \* 6) - (Val(MatchStatus.Status("overs\_now")) \* 6)

If (Val(ballno) >= Val(MatchStatus.Status("inning")) \* Val(MatchStatus.Status("overs\_now")) \* 6) Then

dbc.execute("update status set status\_value = '-1'")

MatchStatus.init()

MsgBox("match Finished")

Me.Close()

Return

End If

End If

End If

If (isWicket) Then

Dim frm As New dialogSelectBatsman()

If (frm.ShowDialog() = Windows.Forms.DialogResult.OK) Then

isWicket = False

End If

End If

If (Val(ballno) Mod 6 = 0 And Not (Val(ballno) = Val(MatchStatus.Status("overs\_now")) \* 6 Or Val(ballno) = 0)) Then

Dim frm As New dialogSelectBowler(MatchStatus.Status("team2"))

If (frm.ShowDialog() = Windows.Forms.DialogResult.OK) Then

Dim temp As String

temp = MatchStatus.Status("player\_bat1")

MatchStatus.Status("player\_bat1") = MatchStatus.Status("player\_bat2")

MatchStatus.Status("player\_bat2") = temp

End If

End If

Dim p1Score As String = dbc.getTotalRunsOfPlayer(MatchStatus.Status("player\_bat1"), MatchStatus.Status("match\_now"))

Dim p2Score As String = dbc.getTotalRunsOfPlayer(MatchStatus.Status("player\_bat2"), MatchStatus.Status("match\_now"))

lblBatingTeam.Text = ds.Tables("teams").Select("id = " + MatchStatus.Status("team1") + "")(0)("team\_name")

lblBowlingTeam.Text = ds.Tables("teams").Select("id = " + MatchStatus.Status("team2") + "")(0)("team\_name")

lblOnStrikeBatsman.Text = ds.Tables("players").Select("id = " + MatchStatus.Status("player\_bat1") + "")(0)("player\_name") & "(" + p1Score + ")"

lblOffStrikeBatsman.Text = ds.Tables("players").Select("id = " + MatchStatus.Status("player\_bat2") + "")(0)("player\_name") & "(" + p2Score + ")"

lblBowler.Text = ds.Tables("players").Select("id = " + MatchStatus.Status("player\_ball") + "")(0)("player\_name")

lblRuns.Text = dbc.getRuns(MatchStatus.Status("match\_now"), MatchStatus.Status("team1"))

lblWickets.Text = wicketCount

Dim bno As String = (Val(MatchStatus.Status("inning")) \* Val(MatchStatus.Status("overs\_now")) \* 6) - (Val(MatchStatus.Status("overs\_now")) \* 6)

lblOvers.Text = OverFromBall(Val(ballno) - bno) & "(" & Val(ballno) - bno & ")"

radioNormalBall.Checked = True

'.Text = dbc.GetTeamName(MatchStatus.Status("team1"))

End Sub

Private Sub btnRun0\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRun0.Click

CountRun(0)

ReloadMatchStatus()

End Sub

Private Sub btnRun1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRun1.Click

CountRun(1)

ReloadMatchStatus()

End Sub

Private Sub btnRun2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRun2.Click

CountRun(2)

ReloadMatchStatus()

End Sub

Private Sub btnRun3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRun3.Click

CountRun(3)

ReloadMatchStatus()

End Sub

Private Sub btnRun4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRun4.Click

CountRun(4)

ReloadMatchStatus()

End Sub

Private Sub btnRun6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRun6.Click

CountRun(6)

ReloadMatchStatus()

End Sub

Private Sub btnBold\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnBold.Click

CountRun(6, "1")

ReloadMatchStatus()

End Sub

Private Sub btnRunOut\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRunOut.Click

CountRun(6, "2")

ReloadMatchStatus()

End Sub

Private Sub btnCatchOut\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCatchOut.Click

CountRun(6, "3")

ReloadMatchStatus()

End Sub

Sub CountRun(ByVal score As String, Optional ByVal wicket As String = "0")

Dim ballno As String = MatchStatus.Status("ball\_now")

Dim nextBallNo As String = Val(MatchStatus.Status("ball\_now")) + 1

Dim ballType As String = getBallType()

If (wicket <> "0") Then

isWicket = True

Else

End If

If (ballType <> "normal") Then

score = Val(score) + 1

nextBallNo = ballno

End If

Dim str As String = "insert into match\_log(match\_id,ball\_no,ball\_type,player\_bat1,player\_bat2,player\_ball,score,wicket)" & \_

"VALUES (" + MatchStatus.Status("match\_now") + "," + ballno + ",'" + ballType + "'," & \_

MatchStatus.Status("player\_bat1") + "," + MatchStatus.Status("player\_bat2") + "," + MatchStatus.Status("player\_ball") & \_

"," + score + "," + wicket + ")"

dbc.CountRun(str)

If (wicket <> "0") Then

isWicket = True

Else

If (score = "1" Or score = "3") Then

Dim temp As String

temp = MatchStatus.Status("player\_bat1")

MatchStatus.Status("player\_bat1") = MatchStatus.Status("player\_bat2")

MatchStatus.Status("player\_bat2") = temp

End If

End If

MatchStatus.Status("ball\_now") = nextBallNo

'MsgBox(str)

End Sub

Function getBallType() As String

If (radioNoball.Checked) Then

Return "no"

ElseIf (radioWiedBall.Checked) Then

Return "wide"

End If

Return "normal"

End Function

Function OverFromBall(ByVal ball As String) As String

Dim dl As Integer = Val(ball)

Dim str As String = Math.Floor(dl / 6) & "." & CStr(dl Mod 6)

Return str

End Function

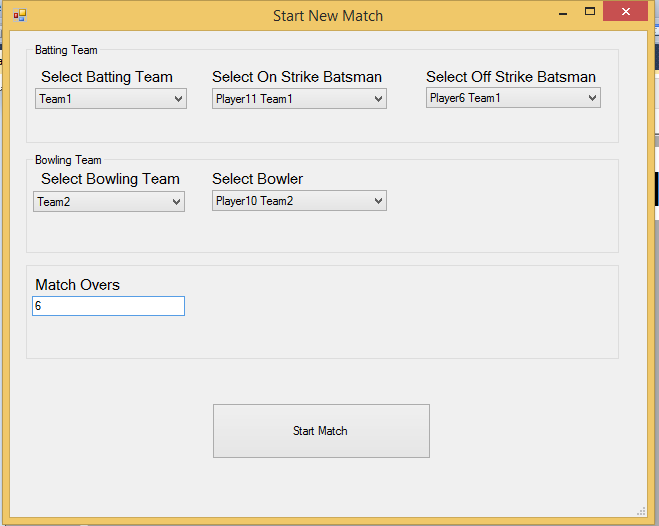
Sub CountWicket(ByVal wicket\_type As String)

End Sub

End Class

**3.1 Start New Match**

This form is used to setup new match. Here you can select batting team and bowling team and players.



**Codes**

Public Class frmStartMatch

Dim dbc As New DbConnection()

Dim ht As New Hashtable

Private Sub frmStartMatch\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dsTeams As New DataSet

Dim dsTeams2 As New DataSet

Dim teamStr As String = "select \* from teams where id in(select team\_id from players group by team\_id having count(team\_id)=11)"

dbc.FillDs(teamStr, dsTeams, "teams")

dbc.FillDs(teamStr, dsTeams2, "teams")

comboBattingTeam.DataSource = dsTeams.Tables("teams")

comboBattingTeam.DisplayMember = "team\_name"

comboBattingTeam.ValueMember = "id"

comboBattingTeam.SelectedIndex = -1

comboBowlingTeam.DataSource = dsTeams2.Tables("teams")

comboBowlingTeam.DisplayMember = "team\_name"

comboBowlingTeam.ValueMember = "id"

comboBowlingTeam.SelectedIndex = -1

End Sub

Private Sub btnStartMatchNow\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnStartMatchNow.Click

If (ValidateForm()) Then

MatchStatus.Status("player\_ball") = comboBowler.SelectedValue

MatchStatus.Status("player\_bat1") = comboOnStrikeBatsman.SelectedValue

MatchStatus.Status("player\_bat2") = comboOffStrikeBatsman.SelectedValue

MatchStatus.Status("team1") = comboBattingTeam.SelectedValue

MatchStatus.Status("team2") = comboBowlingTeam.SelectedValue

MatchStatus.Status("match\_now") = dbc.InsertNewmatch(MatchStatus.Status("team1"), MatchStatus.Status("team2"), tbMatchOvers.Text)

MatchStatus.Status("ball\_now") = "0"

MatchStatus.Status("overs\_now") = tbMatchOvers.Text

MatchStatus.Status("inning") = "1"

MatchStatus.update()

Me.DialogResult = Windows.Forms.DialogResult.OK

End If

End Sub

Private Sub comboBattingTeam\_SelectedValueChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles comboBattingTeam.SelectionChangeCommitted

LoadPlayerBatting(comboBattingTeam.SelectedValue)

End Sub

Sub LoadPlayerBatting(ByVal id As String)

Dim ds As New DataSet

Dim ds2 As New DataSet

dbc.FillDs("select \* from players where team\_id=" + id + " order by player\_type desc", ds, "players")

dbc.FillDs("select \* from players where team\_id=" + id + " order by player\_type desc", ds2, "players")

comboOnStrikeBatsman.DataSource = ds.Tables("players")

comboOnStrikeBatsman.DisplayMember = "player\_name"

comboOnStrikeBatsman.ValueMember = "id"

comboOffStrikeBatsman.DataSource = ds2.Tables("players")

comboOffStrikeBatsman.DisplayMember = "player\_name"

comboOffStrikeBatsman.ValueMember = "id"

End Sub

Sub LoadPlayerBowling(ByVal id As String)

Dim ds As New DataSet

dbc.FillDs("select \* from players where team\_id=" + id + " order by player\_type desc", ds, "players")

comboBowler.DataSource = ds.Tables("players")

comboBowler.DisplayMember = "player\_name"

comboBowler.ValueMember = "id"

End Sub

Private Sub comboBowlingTeam\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles comboBowlingTeam.SelectionChangeCommitted

LoadPlayerBowling(comboBowlingTeam.SelectedValue)

End Sub

Function ValidateForm() As Boolean

If (comboBattingTeam.SelectedIndex = -1) Then

MsgBox("Please Select Batting Team")

Return False

ElseIf (comboBowlingTeam.SelectedIndex = -1) Then

MsgBox("Please Select Bowling Team")

Return False

ElseIf (comboBattingTeam.SelectedValue = comboBowlingTeam.SelectedValue) Then

MsgBox("Please Select Tow Different Teams")

Return False

ElseIf (comboOnStrikeBatsman.SelectedValue = comboOffStrikeBatsman.SelectedValue) Then

MsgBox("Select Different On and Off strike batsman")

Return False

ElseIf (tbMatchOvers.Text = vbNullString) Then

MsgBox("Enter match Overs")

Return False

End If

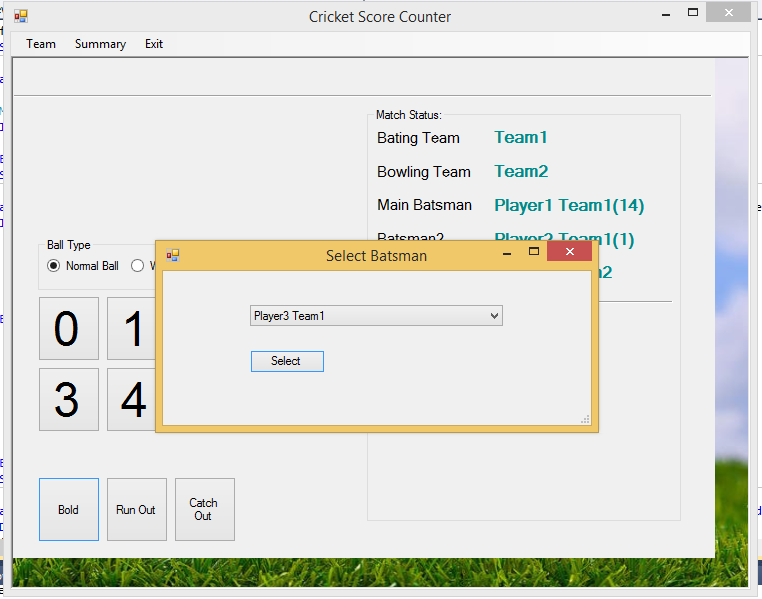
Return True

End Function

End Class

**3.2 Select New Batsman**

This is dialog box which is pop up when any wicket is gone. It helps to select new batsman in place of batsman who is gone.



**Codes**

Public Class dialogSelectBatsman

Private ds As New DataSet()

Dim dbc As New DbConnection()

Private Sub dialogSelectBatsman\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

dbc.FillDs("select \* from players where id not in(select distinct player\_bat1 from match\_log where match\_id=" + MatchStatus.Status("match\_now") + " and not wicket=0) and not id=" + MatchStatus.Status("player\_bat2") + " and team\_id=" + MatchStatus.Status("team1"), ds, "players")

comboSelectBatsman.DataSource = ds.Tables("players")

comboSelectBatsman.ValueMember = "id"

comboSelectBatsman.DisplayMember = "player\_name"

End Sub

Private Sub btnSelectBatsman\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSelectBatsman.Click

MatchStatus.Status("player\_bat1") = comboSelectBatsman.SelectedValue

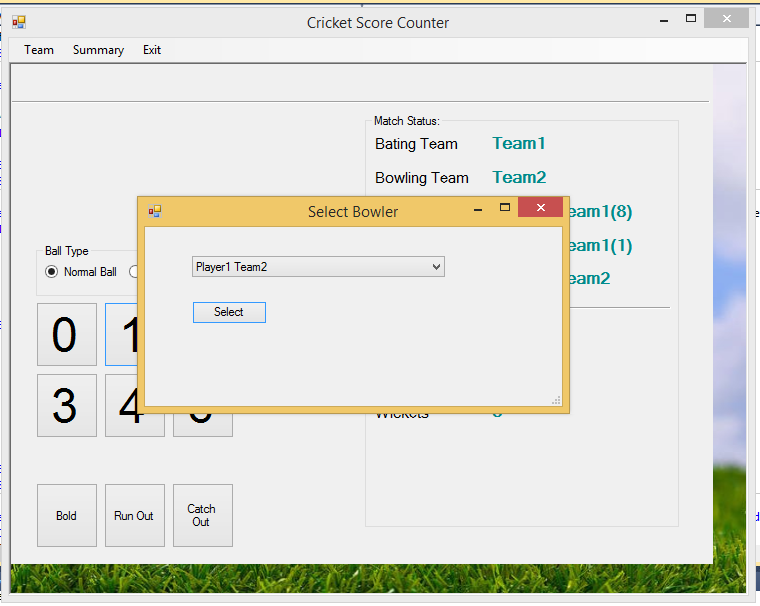
Me.DialogResult = Windows.Forms.DialogResult.OK

End Sub

End Class

**3.3 Select New Bowler**

This is dialog box which is pop up when Over is complete. It helps to select new bowler for new over.



**Codes**

Public Class dialogSelectBowler

Private ds As New DataSet()

Private team\_id As String

Dim dbc As New DbConnection()

Public Sub New(ByVal team\_id As String)

Me.team\_id = team\_id

InitializeComponent()

End Sub

Private Sub dialogSelectBowler\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

dbc.FillDs("select \* from players where team\_id=" + team\_id + " and id not in(select TOP 1 player\_ball from match\_log order by id desc)", ds, "players")

comboSelectBowler.DataSource = ds.Tables("players")

comboSelectBowler.ValueMember = "id"

comboSelectBowler.DisplayMember = "player\_name"

End Sub

Private Sub btnSelectBowler\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSelectBowler.Click

MatchStatus.Status("player\_ball") = comboSelectBowler.SelectedValue

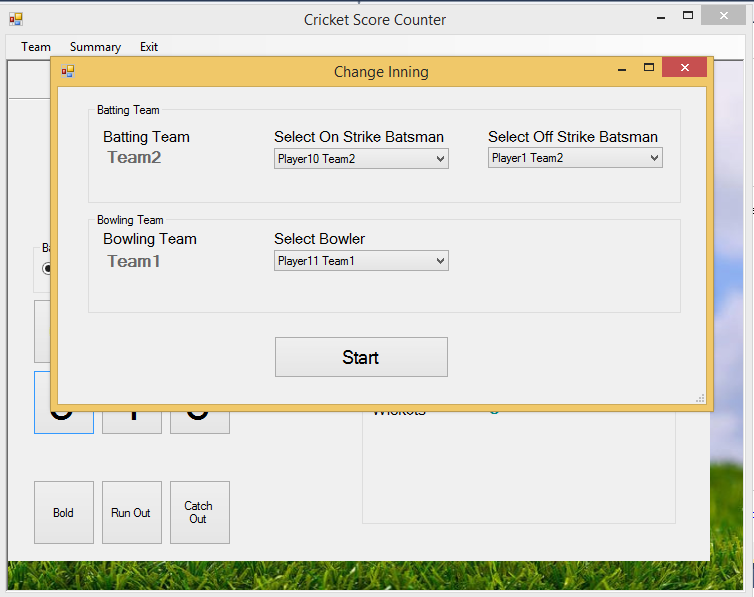
Me.DialogResult = Windows.Forms.DialogResult.OK

End Sub

End Class

**3.4 Change Inning**

When inning finish this form helps to select players for new inning.



**Codes**

Public Class dialogChangeInning

Dim dbc As New DbConnection()

Sub LoadPlayerBatting(ByVal id As String)

Dim ds As New DataSet

Dim ds2 As New DataSet

dbc.FillDs("select \* from players where team\_id=" + id + " order by player\_type desc", ds, "players")

dbc.FillDs("select \* from players where team\_id=" + id + " order by player\_type desc", ds2, "players")

comboOnStrikeBatsman.DataSource = ds.Tables("players")

comboOnStrikeBatsman.DisplayMember = "player\_name"

comboOnStrikeBatsman.ValueMember = "id"

comboOffStrikeBatsman.DataSource = ds2.Tables("players")

comboOffStrikeBatsman.DisplayMember = "player\_name"

comboOffStrikeBatsman.ValueMember = "id"

End Sub

Sub LoadPlayerBowling(ByVal id As String)

Dim ds As New DataSet

dbc.FillDs("select \* from players where team\_id=" + id + " order by player\_type desc", ds, "players")

comboBowler.DataSource = ds.Tables("players")

comboBowler.DisplayMember = "player\_name"

comboBowler.ValueMember = "id"

End Sub

Function ValidateForm() As Boolean

If (comboOnStrikeBatsman.SelectedValue = comboOffStrikeBatsman.SelectedValue) Then

MsgBox("Select Different On and Off strike batsman")

Return False

End If

Return True

End Function

Private Sub btnStartInning\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnStartInning.Click

If (ValidateForm()) Then

MatchStatus.Status("player\_ball") = comboBowler.SelectedValue

MatchStatus.Status("player\_bat1") = comboOnStrikeBatsman.SelectedValue

MatchStatus.Status("player\_bat2") = comboOffStrikeBatsman.SelectedValue

MatchStatus.Status("ball\_now") = (MatchStatus.Status("overs\_now")) \* 6

MatchStatus.Status("inning") = "2"

MatchStatus.update()

Me.DialogResult = Windows.Forms.DialogResult.OK

End If

End Sub

Private Sub dialogChangeInning\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

lblBatingTeam.Text = dbc.GetTeamName(MatchStatus.Status("team2"))

lblBowlingTeam.Text = dbc.GetTeamName(MatchStatus.Status("team1"))

LoadPlayerBatting(MatchStatus.Status("team2"))

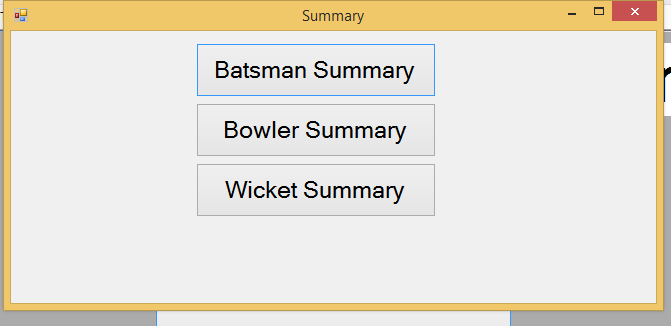
LoadPlayerBowling(MatchStatus.Status("team1"))

End Sub

End Class

1. **Summary**

This is form containing buttons for viewing different type of summaries.



**Codes**

Public Class frmSummary

Private Sub btnBatsmanSummary\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnBatsmanSummary.Click

Dim frm As New frmBatsmanSummary()

frm.ShowDialog()

End Sub

Private Sub btnBowlerSummary\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnBowlerSummary.Click

Dim frm As New frmBowlerSummary()

frm.ShowDialog()

End Sub

Private Sub btnWicketSummary\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnWicketSummary.Click

Dim frm As New frmWicketSummary()

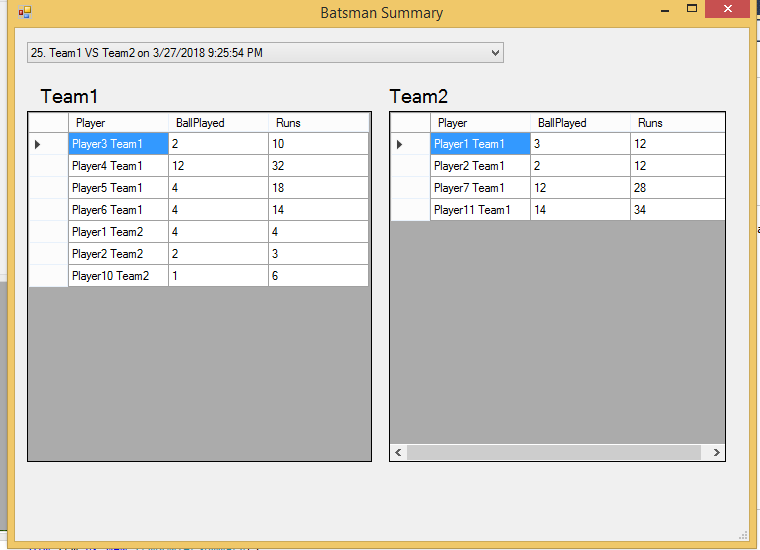
frm.ShowDialog()

End Sub

End Class

**4.1 Batsman Summary**

Batsman summary includes ball played by batsman runs made by batsman etc.



**Codes**

Public Class frmBatsmanSummary

Dim dbc As New DbConnection()

Dim ds As New DataSet()

Private Sub frmBatsmanSummary\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

dbc.FillDs("select id,team1,team2,overs,id&'. '&(select team\_name from teams where id=matchs.team1)&' VS '&(select team\_name from teams where id=matchs.team2)&' on '&match\_date as teamNames from matchs", ds, "matchs")

comboSelectMatch.DataSource = ds.Tables("matchs")

comboSelectMatch.DisplayMember = "teamNames"

comboSelectMatch.ValueMember = "id"

End Sub

Private Sub comboSelectMatch\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles comboSelectMatch.SelectionChangeCommitted

Dim dr = ds.Tables("matchs").Select("id=" + CStr(comboSelectMatch.SelectedValue))

lblTeam1.Text = dbc.GetTeamName(dr(0)("team1"))

lblTeam2.Text = dbc.GetTeamName(dr(0)("team2"))

Dim overs As Integer = dr(0)("overs")

Try

ds.Tables("players1").Clear()

ds.Tables("players2").Clear()

Catch ex As Exception

End Try

dbc.FillDs("select (select player\_name from players where id=match\_log.player\_bat1) as Player,count(player\_bat1) as BallPlayed ,sum(score) as Runs from match\_log where match\_id=" & comboSelectMatch.SelectedValue & " and ball\_no < " & (overs \* 6) & " group by player\_bat1", ds, "players1")

dgvTeam1.DataSource = ds.Tables("players1")

dbc.FillDs("select (select player\_name from players where id=match\_log.player\_bat1) as Player,count(player\_bat1) as BallPlayed ,sum(score) as Runs from match\_log where match\_id=" & comboSelectMatch.SelectedValue & " and ball\_no > " & (overs \* 6) & " group by player\_bat1", ds, "players2")

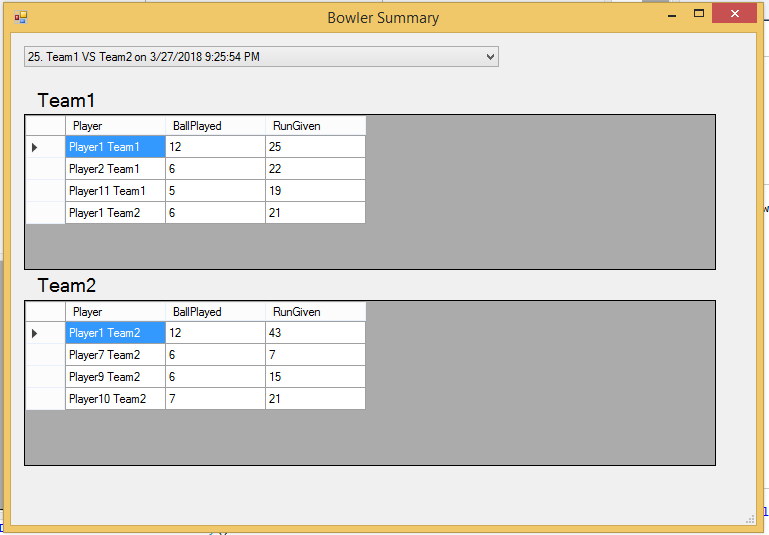
dgvTeam2.DataSource = ds.Tables("players2")

End Sub

End Class

**4.2 Bowler Summary**

Bowler summary includes balls thrown by bowler and Runs given to batsman. Etc.



**Codes**

Public Class frmBowlerSummary

Dim dbc As New DbConnection()

Dim ds As New DataSet()

Private Sub frmBowlerSummary\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

dbc.FillDs("select id,team1,team2,overs,id&'. '&(select team\_name from teams where id=matchs.team1)&' VS '&(select team\_name from teams where id=matchs.team2)&' on '&match\_date as teamNames from matchs", ds, "matchs")

comboSelectMatch.DataSource = ds.Tables("matchs")

comboSelectMatch.DisplayMember = "teamNames"

comboSelectMatch.ValueMember = "id"

End Sub

Private Sub comboSelectMatch\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles comboSelectMatch.SelectionChangeCommitted

Dim dr = ds.Tables("matchs").Select("id=" + CStr(comboSelectMatch.SelectedValue))

lblTeam1.Text = dbc.GetTeamName(dr(0)("team1"))

lblTeam2.Text = dbc.GetTeamName(dr(0)("team2"))

Dim overs As Integer = dr(0)("overs")

Try

ds.Tables("players1").Clear()

ds.Tables("players2").Clear()

Catch ex As Exception

End Try

dbc.FillDs("select (select player\_name from players where id=match\_log.player\_ball) as Player,count(player\_ball) as BallPlayed ,sum(score) as RunGiven from match\_log where match\_id=" & comboSelectMatch.SelectedValue & " and ball\_no > " & (overs \* 6) & " group by player\_ball", ds, "players1")

dgvTeam1.DataSource = ds.Tables("players1")

dbc.FillDs("select (select player\_name from players where id=match\_log.player\_ball) as Player,count(player\_ball) as BallPlayed ,sum(score) as RunGiven from match\_log where match\_id=" & comboSelectMatch.SelectedValue & " and ball\_no < " & (overs \* 6) & " group by player\_ball", ds, "players2")

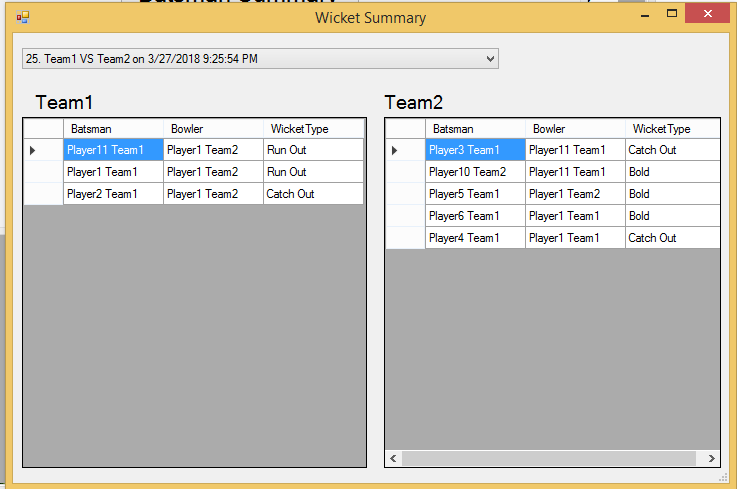
dgvTeam2.DataSource = ds.Tables("players2")

End Sub

End Class

**4.3 Wicket Summary**

Wicket Summary Provide details about Fall of wickets.



**Codes**

Public Class frmWicketSummary

Dim dbc As New DbConnection()

Dim ds As New DataSet()

Function WicketString(ByVal num As String) As String

If (num = "1") Then

Return "Bold"

ElseIf (num = "2") Then

Return "Run Now"

ElseIf (num = "3") Then

Return "Catch Out"

End If

Return ""

End Function

Private Sub frmWicketSummary\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

dbc.FillDs("select id,team1,team2,overs,id&'. '&(select team\_name from teams where id=matchs.team1)&' VS '&(select team\_name from teams where id=matchs.team2)&' on '&match\_date as teamNames from matchs", ds, "matchs")

comboSelectMatch.DataSource = ds.Tables("matchs")

comboSelectMatch.DisplayMember = "teamNames"

comboSelectMatch.ValueMember = "id"

End Sub

Private Sub comboSelectMatch\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles comboSelectMatch.SelectionChangeCommitted

Dim dr = ds.Tables("matchs").Select("id=" + CStr(comboSelectMatch.SelectedValue))

lblTeam1.Text = dbc.GetTeamName(dr(0)("team1"))

lblTeam2.Text = dbc.GetTeamName(dr(0)("team2"))

Dim overs As Integer = dr(0)("overs")

Try

ds.Tables("players1").Clear()

ds.Tables("players2").Clear()

Catch ex As Exception

End Try

dbc.FillDs("select (select player\_name from players where id=match\_log.player\_bat1) as Batsman,(select player\_name from players where id=match\_log.player\_ball)as Bowler,(select wicket\_value from wicket\_type where wicket\_num=match\_log.wicket) as WicketType from match\_log where match\_id=" & comboSelectMatch.SelectedValue & " and ball\_no > " & (overs \* 6) & " and not wicket=0", ds, "players1")

dgvTeam1.DataSource = ds.Tables("players1")

dbc.FillDs("select (select player\_name from players where id=match\_log.player\_bat1) as Batsman,(select player\_name from players where id=match\_log.player\_ball)as Bowler,(select wicket\_value from wicket\_type where wicket\_num=match\_log.wicket) as WicketType from match\_log where match\_id=" & comboSelectMatch.SelectedValue & " and ball\_no < " & (overs \* 6) & " and not wicket=0", ds, "players2")

dgvTeam2.DataSource = ds.Tables("players2")

End Sub

Private Sub dgvTeam1\_RowsAdded(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewRowsAddedEventArgs) Handles dgvTeam1.RowsAdded

For i As Integer = e.RowIndex To e.RowIndex + e.RowCount - 1

' dgvTeam1.Columns(dgvTeam1.Rows(i).Cells.Item("wicket").ColumnIndex).ValueType = GetType(String)

'dgvTeam1.Rows(i).Cells.Item("wicket").Value = WicketString(dgvTeam1.Rows(i).Cells.Item("wicket").Value)

Next

End Sub

End Class

**5.0 Helper Classes**

**5.1 DbConnection**

This is common class for connecting to database. All database related queries are fired from this class.

Imports System.Data.OleDb

Public Class DbConnection

Private conn As New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data Source=E:\extra\bca\sem4\hmdadie\ccbyhmd\ccbyhmdDb.accdb")

Private cmd As New OleDbCommand()

Private da As New OleDbDataAdapter(cmd)

Sub New()

cmd.Connection = conn

End Sub

Public Function RunSql(ByVal str As String) As OleDbDataAdapter

Dim cmd2 As New OleDbCommand()

cmd2.Connection = conn

cmd2.CommandText = str

Return New OleDbDataAdapter(cmd2)

End Function

Public Sub FillDs(ByVal str As String, ByRef ds As DataSet, ByVal tableName As String)

cmd.Connection = conn

cmd.CommandText = str

da.SelectCommand = cmd

da.Fill(ds, tableName)

End Sub

Public Sub InsertIntoTeams(ByVal teamName As String)

conn.Open()

cmd.CommandText = "insert into teams(team\_name) values ('" + teamName + "')"

cmd.ExecuteNonQuery()

conn.Close()

End Sub

Sub DeleteFromTeams(ByVal id As String)

conn.Open()

cmd.CommandText = "delete from teams where id = " + id

cmd.ExecuteNonQuery()

conn.Close()

End Sub

Function InsertIntoPlayers(ByVal pName As String, ByVal pType As String, ByVal teamId As String)

conn.Open()

cmd.CommandText = "insert into players(player\_name,player\_type,team\_id) values ('" + pName + "','" + pType + "'," + teamId + ")"

Dim cnt = cmd.ExecuteNonQuery()

conn.Close()

Return cnt

End Function

Function RemoveFromPlayers(ByVal id As String) As Integer

conn.Open()

cmd.CommandText = "delete from players where id = " + id

Dim cnt = cmd.ExecuteNonQuery()

conn.Close()

Return cnt

End Function

Function UpdatePlayers(ByVal id As String, ByVal pName As String, ByVal pType As String, ByVal teamId As String)

conn.Open()

cmd.CommandText = "update players set player\_name='" + pName + "',player\_type='" + pType + "',team\_id=" + teamId + " where id=" + id

Dim cnt = cmd.ExecuteNonQuery()

conn.Close()

Return cnt

End Function

Function InsertNewmatch(ByVal team1 As String, ByVal team2 As String, ByVal overs As String) As Integer

conn.Open()

cmd.CommandText = "insert into matchs (team1,team2,overs) values ('" + team1 + "','" + team2 + "'," + overs + ")"

Dim cnt = cmd.ExecuteNonQuery()

If (cnt = 1) Then

cmd.CommandText = "select max(id) from matchs"

Return cmd.ExecuteScalar()

End If

conn.Close()

Return 0

End Function

Function GetPlayerName(ByVal id As String) As String

conn.Open()

cmd.CommandText = "select player\_name from players where id=" + id

Dim temp = cmd.ExecuteScalar()

conn.Close()

Return temp

End Function

Function GetTeamName(ByVal id As String) As String

conn.Open()

cmd.CommandText = "select team\_name from teams where id=" + id

Dim temp = cmd.ExecuteScalar()

conn.Close()

Return temp

End Function

Function execute(ByVal str As String) As Integer

conn.Open()

cmd.CommandText = str

Dim cnt = cmd.ExecuteNonQuery()

conn.Close()

Return cnt

End Function

Function getRuns(ByVal match\_id As String, ByVal team\_id As String) As String

conn.Open()

Dim bno As String = (Val(MatchStatus.Status("inning")) \* Val(MatchStatus.Status("overs\_now")) \* 6) - (Val(MatchStatus.Status("overs\_now")) \* 6)

cmd.CommandText = "select sum(score) from match\_log where match\_id=" + match\_id + " and ball\_no > " + bno

'MsgBox("select sum(score) from match\_log where match\_id=" + match\_id + " and team1=" + team\_id)

Try

Dim temp As String = "0"

temp = cmd.ExecuteScalar()

Return temp

Catch ex As Exception

'MsgBox(ex.Message)

Finally

conn.Close()

End Try

Return "0"

End Function

Function getWicketCount(ByVal match\_id As String, ByVal team\_id As String) As String

conn.Open()

Dim bno As String = (Val(MatchStatus.Status("inning")) \* Val(MatchStatus.Status("overs\_now")) \* 6) - (Val(MatchStatus.Status("overs\_now")) \* 6)

cmd.CommandText = "select count(wicket) from match\_log where not wicket=0 and match\_id=" + match\_id + " and ball\_no > " + bno

Try

Dim temp As String = "0"

temp = cmd.ExecuteScalar()

Return temp

Catch ex As Exception

' MsgBox(ex.Message)

Finally

conn.Close()

End Try

Return "0"

End Function

Function CountRun(ByVal queryStr As String) As Integer

conn.Open()

cmd.CommandText = queryStr

Dim cnt = cmd.ExecuteNonQuery()

conn.Close()

Return cnt

End Function

Function getTotalRunsOfPlayer(ByVal player\_di As String, ByVal match\_id As String) As String

conn.Open()

cmd.CommandText = "select sum(score) from match\_log where player\_bat1=" + player\_di + " and match\_id=" + match\_id

Try

Dim temp As String = "0"

temp = cmd.ExecuteScalar()

Return temp

Catch ex As Exception

'MsgBox(ex.Message)

Finally

conn.Close()

End Try

Return "0"

End Function

End Class

**5.2 MatchStatus**

This class is used to store some status about match like. Current Match ID current player’s details etc.

Public Class MatchStatus

Private Shared st As New Hashtable

Shared fetched As Boolean

Shared Sub init()

Dim dbc As New DbConnection()

Dim ds As New DataSet

dbc.FillDs("select \* from status", ds, "status")

For i = 0 To ds.Tables("status").Rows.Count - 1

st(ds.Tables("status").Rows(i)("status\_name")) = ds.Tables("status").Rows(i)("status\_value")

Next

End Sub

Public Shared Sub update()

Dim dbc As New DbConnection()

Dim tempstr As String

For i = 0 To st.Keys.Count - 1

tempstr = "update status set status\_value='" + st.Item(st.Keys(i)) + "' where status\_name='" + st.Keys(i) + "'"

dbc.execute(tempstr)

'MsgBox("update status set status\_value='" + st.Keys(i) + "' where status\_name='" + st.Item(st.Keys(i)) + "'")

Next

End Sub

Public Shared Property Status(ByVal nm As String) As String

Get

Return st.Item(nm)

End Get

Set(ByVal value As String)

st.Item(nm) = value

End Set

End Property

Public Shared Property NewProperty() As String

Get

Return ""

End Get

Set(ByVal value As String)

End Set

End Property

End Class

**5.2 CricketCounter**

This form provide some common global functions to help calculations.

Public Class CricketCounter

Public Shared Function PlayerTypeList() As ArrayList

Dim typeList As New ArrayList()

typeList.Add(New PlayerType("bat", "Batsman"))

typeList.Add(New PlayerType("ball", "Bowler"))

typeList.Add(New PlayerType("all", "All Rounder"))

Return typeList

End Function

Public Shared Function getPlayerTypeString(ByVal str As String) As String

If (str = "bat") Then

Return "Batsman"

ElseIf (str = "ball") Then

Return "Bowler"

Else

Return "All Rounder"

End If

Return 0

End Function

Public Shared Function getPlayerStringType(ByVal str As String) As String

If (str = "Batsman") Then

Return "bat"

ElseIf (str = "Bowler") Then

Return "ball"

Else

Return "all"

End If

Return 0

End Function

End Class

Class PlayerType

Public typeText As String

Public typeLabel As String

Sub New(ByVal t As String, ByVal l As String)

typeText = t

typeLabel = l

End Sub

End Class