```csharp
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Windows;
7  using System.Windows.Controls;
8  using System.Windows.Data;
9  using System.Windows.Documents;
10 using System.Windows.Input;
11 using System.Windows.Media;
12 using System.Windows.Media.Imaging;
13 using System.Windows.Navigation;
14 using System.Windows.Shapes;
15
16 namespace QuiddichV2._0
17 {
18     /// <summary>
19     /// Interaction logic for AddPlayer.xaml
20     /// </summary>
21     public partial class AddPlayer : Page
22     {
23         public Player player;
24         public bool addPlayer;
25         private int teamID;
26         private string firstName;
27         private string lastName;
28
29         public AddPlayer()
30         {
31             InitializeComponent();
32         }
33
34         public AddPlayer(int teamID)
35         {
36             InitializeComponent();
37             this.teamID = teamID;
38         }
39
40         public AddPlayer(int teamID, string firstName, string lastName)
41         {
42             InitializeComponent();
43             this.teamID = teamID;
44             this.firstName = firstName;
45             this.lastName = lastName;
46         }
47
48         private void formLoaded(object sender, RoutedEventArgs e)
49         {
50             if(addPlayer)
51             {
52                 txtTeamID.IsReadOnly = true;
53                 txtTeamID.Text = teamID.ToString();
54             }
55             else
56             {
57                 this.GetPlayer(teamID, firstName, lastName);
58             }
59         }
60
61         private void btnSave_Click(object sender, RoutedEventArgs e)
62         {
63             if (IsValidData())
64             {
65                 if (addPlayer)
66                 {
```

```
67
68                              player = new Player();
69                              this.PutPlayerData(player);
70                              try
71                              {
72                                  QuiddichDB.AddPlayer(player);
73                              }
74                              catch (Exception ex)
75                              {
76                                  MessageBox.Show(ex.Message, ex.GetType().ToString());
77                              }
78                          }
79                          else
80                          {
81                              this.GetPlayer(teamID, firstName, lastName);
82                              Player newPlayer = new Player();
83                              newPlayer.TeamID = player.TeamID;
84                              newPlayer.FirstName = player.FirstName;
85                              newPlayer.LastName = player.LastName;
86                              newPlayer.UniformNumber = player.UniformNumber;
87                              newPlayer.Position = player.Position;
88                              this.PutPlayerData(newPlayer);
89                              try
90                              {
91                                  if (!QuiddichDB.UpdatePlayer(player, newPlayer))
92                                  {
93                                      MessageBox.Show("Another user has updated or " +
94                                          "deleted that team.", "Database Error");
95                                  }
96                                  else
97                                  {
98                                      player = newPlayer;
99                                  }
100                             }
101                             catch (Exception ex)
102                             {
103                                 MessageBox.Show(ex.Message, ex.GetType().ToString());
104                             }
105                         }
106
107                         PlayerRoster roster = new PlayerRoster(teamID);
108                         NavigationService.Navigate(roster);
109
110                 }
111         }
112
113         private void PutPlayerData(Player player)
114         {
115
116             player.TeamID = Convert.ToInt32(txtTeamID.Text);
117             player.FirstName = txtFirstName.Text;
118             player.LastName = txtLastName.Text;
119             player.UniformNumber = Convert.ToInt32(txtUniformNumber.Text);
120             player.Position = txtPosition.Text;
121
122         }
123
124         private void GetPlayer(int teamID)
125         {
126             player = QuiddichDB.GetPlayer(teamID);
127             txtTeamID.Text = player.TeamID.ToString();
128             txtFirstName.Text = player.FirstName;
129             txtLastName.Text = player.LastName;
130             txtUniformNumber.Text = player.UniformNumber.ToString();
131             txtPosition.Text = player.Position;
132
```

```
133            }
134
135        private void GetPlayer(int teamID, string firstName, string lastName)
136        {
137            player = QuiddichDB.GetPlayer(teamID, firstName, lastName);
138            txtTeamID.Text = player.TeamID.ToString();
139            txtFirstName.Text = player.FirstName;
140            txtLastName.Text = player.LastName;
141            txtUniformNumber.Text = player.UniformNumber.ToString();
142            txtPosition.Text = player.Position;
143
144        }
145
146        private bool IsValidData()
147        {
148            return Validator.IsPresent(txtFirstName) &&
149                    Validator.IsPresent(txtLastName) &&
150                    Validator.IsPresent(txtUniformNumber) &&
151                    Validator.IsInt32(txtUniformNumber) &&
152                    Validator.IsPresent(txtPosition);
153
154        }
155
156        private void btnCancel_Click(object sender, RoutedEventArgs e)
157        {
158            PlayerRoster roster = new PlayerRoster(teamID);
159            NavigationService.Navigate(roster);
160        }
161    }
162 }
163
```