```
1  using ASP;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Security.Policy;
6  using System.Web;
7  using System.Web.Services.Description;
8  using System.Web.UI;
9  using System.Web.UI.HtmlControls;
10 using System.Web.UI.WebControls;
11 using DAL;
12 using DAL.apiClases;
13 using gf = DAL.GlobalFunctions;
14 using System.Diagnostics;
15
16 public partial class cardDeck : System.Web.UI.Page
17 {
18     public string carddeck;
19     private List<Card> cards;
20     private Dictionary<string, clubColour> clbclr;
21     protected void Page_Load(object sender, EventArgs e)
22     {
23         //only people logged in can use this page
24         if (Session["username"] == null)
25         {
26             Response.Redirect("~/initPage.aspx");
27         }
28
29         createpacksforuser();
30         saveB.Visible = false;
31         clbclr = CardFunctions.getcolours();
32     }
33     // function creates all the pack for logged in user
34     public void createpacksforuser()
35     {
36         List<int> packsFU = PackFunctions.packsByUsername(Session
             ["username"].ToString());
37         foreach (int PackId in packsFU)
38         {
39             HtmlGenericControl cont = new HtmlGenericControl("DIV");
40             cont.Attributes.Add("class", "col");
41             PackPlaceHolder.Controls.Add(cont);
42             HtmlGenericControl css = new HtmlGenericControl("DIV");
43             string adress = "";
44             switch (PackId)
45             {
46                 case 2:
47                     adress = "images/packs/gold.png";
48                     break;
49                 case 1:
50                     adress = "images/packs/diamond.png";
51                     break;
```

```
52                  case 4:
53                      adress = "images/packs/silver.png";
54                      break;
55                  default:
56                      adress = "images/packs/silver.png";
57                      break;
58              }
59              css.Attributes.Add("class", "row");
60              css.InnerHtml += "<img src='" + adress + "'/>";
61              cont.Controls.Add(css);
62              HtmlGenericControl cont2 = new HtmlGenericControl("DIV");
63              cont2.Attributes.Add("class", "row");
64              cont.Controls.Add(cont2);
65              Button b = new Button();
66              b.Style.Add("width", "290px");
67              b.Text = PackId.ToString();
68              b.Click += new EventHandler(openPack_Click);
69              b.CssClass = "btn btn-primary";
70              b.Attributes.Add("onclick", "openpack();");
71              cont2.Controls.Add(b);
72          }
73          if (PackPlaceHolder.Controls.Count == 0)
74              carddeck = "<p>no packs available</p>";
75      }
76
77      // the function gets the rating and the card list and handesls the return if
          contains
78      private Card cardByRating(List<Card> cards, int ratinglow, int ratinghigh)
79      {
80          Card player = CardFunctions.getByRating(ratinglow,ratinghigh);
81          while (doesContain(cards, player.name))
82          {
83              player = CardFunctions.getByRating(ratinglow,ratinghigh);
84          }
85          return player;
86      }
87
88      //the function gets the current list of cards and a name of the next card and
          returns
89      //true if duplicaate and false if not.
90      private bool doesContain(List<Card> cards, string name)
91      {
92          foreach (Card player in cards)
93          {
94              if (player.name == name)
95                  return true;
96          }
97          return false;
98      }
99      //function thats called when open packs is pressed
100     protected void openPack_Click(object sender, EventArgs e)
101     {
```

```
102         int packID;
103         try
104         {
105             packID = Convert.ToInt32(((Button)sender).Text);
106             Debug.WriteLine(packID.ToString());
107         }
108         catch
109         {
110             packID = -1;
111             Debug.WriteLine("Failed to find PACKID");
112         }
113         try
114         {
115             if (packID != -1)
116             {
117                 #region chances
118                 int total = 0;
119                 string[] odds = PackFunctions.getByPackId(packID);
120                 int under80 = total + Convert.ToInt32(odds[1]);
121                 total += Convert.ToInt32(odds[1]);
122                 int under85 = total + Convert.ToInt32(odds[2]);
123                 total += Convert.ToInt32(odds[2]);
124                 int under90 = total + Convert.ToInt32(odds[3]);
125                 total += Convert.ToInt32(odds[3]);
126                 int under99 = total + Convert.ToInt32(odds[5]);
127                 total += Convert.ToInt32(odds[4]);
128                 int special = total + Convert.ToInt32(odds[4]);
129                 total += Convert.ToInt32(odds[5]);
130 
131                 Debug.WriteLine(under80);
132                 Debug.WriteLine(under85);
133                 Debug.WriteLine(under90);
134                 Debug.WriteLine(under99);
135                 Debug.WriteLine(special);
136                 #endregion
137 
138                 Random rnd = new Random();
139                 cards = new List<Card>();
140                 List<string> cid = new List<string>();
141                 for (int i = 0; i < 5; i++)
142                 {
143                     int num = rnd.Next(100);
144                     if (num <= under80)
145                     {
146                         Card card = cardByRating(cards, 0, 80);
147                         cards.Add(card);
148                         cid.Add(card.id.ToString());
149                     }
150                     else if (num > under80 && num <= under85)
151                     {
152                         Card card = cardByRating(cards, 80, 85);
153                         cards.Add(card);
```

```
154                         cid.Add(card.id.ToString());
155                     }
156                     else if (num > under85 && num <= under90)
157                     {
158                         Card card = cardByRating(cards, 85, 90);
159                         cards.Add(card);
160                         cid.Add(card.id.ToString());
161                     }
162                     else if (num > under90 && num <= under99)
163                     {
164                         Card card = cardByRating(cards, 90, 98);
165                         cards.Add(card);
166                         cid.Add(card.id.ToString());
167                     }
168                     else
169                     {
170 
171                     }
172                     if (num == 99)
173                     {
174                         int rnd2 = rnd.Next(100);
175                         if (rnd2 > 50)
176                         {
177                             Card crd = CardFunctions.getByCardId(984);
178                             cards.Add(crd);
179                             cid.Add(crd.id.ToString());
180                         }
181                     }
182                     Debug.WriteLine("num- " + num);
183                 }
184                 Debug.WriteLine("started with API");
185                 Dictionary<string, Element> dic = APICall.getListOfStats(cid);
186                 Debug.WriteLine("Done with API");
187                 foreach (Card player in cards.OrderByDescending(x =>
                      x.rating).ToList())
188                 {
189                     try
190                     {
191                         carddeck += gf.createCard(player, clbclr[player.club],
                         dic[player.id.ToString()]);
192                     }
193                     catch
194                     {
195                         carddeck += gf.createCard(player, clbclr[player.club],
                         new Element());
196                     }
197 
198                     if (!cardInv.checkDuplicate(Session["username"].ToString(),
                         player.id))
199                     {
200                         cardInv.Addplayer(Session["username"].ToString(),
                         player.id);//remooooove
```

```
201                         }
202                         Debug.WriteLine(player.id);
203                     }
204                     PackFunctions.deletePack(Session["username"].ToString(),
                          packID);//remooooove
205                     PackPlaceHolder.Controls.Remove((Button)sender);//remooooove
206                     PackPlaceHolder.Visible = false;
207                     saveB.Visible = true;
208                 }
209
210
211             }
212             catch
213             {
214
215             }
216         }
217         //function that runs when you want to save pack, saves to database.
218         protected void saveB_Click(object sender, EventArgs e)
219         {
220             PackPlaceHolder.Visible = true;
221             saveB.Visible = false;
222             carddeck = "";
223             Debug.WriteLine("saved");
224             if (PackPlaceHolder.Controls.Count == 0)
225                 carddeck = "<p>no packs available</p>";
226         }
227
228         protected void testbtn_Click(object sender, EventArgs e)
229         {
230             PackFunctions.addPack(Session["username"].ToString(), 1);
231         }
232     }
```