

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Data;
5  using System.Web;
6  using System.Web.Services;
7  using System.Data.Sql;
8  using System.Data.SqlClient;
9  using System.Collections.ObjectModel;
10 using System.Collections;
11 using System.Web.Configuration;
12 using System.Web.Services.Protocols;
13
14 namespace imeny
15 {
16     /// <summary>
17     /// Summary description for WebService1
18     /// </summary>
19     [WebService(Namespace = "http://tempuri.org/")]
20     [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
21     [System.ComponentModel.ToolboxItem(false)]
22     // To allow this Web Service to be called from script, using ASP.NET AJAX,
23     // uncomment the following line.
24     // [System.Web.Script.Services.ScriptService]
25     public class Kasutaja
26     {
27         public string id { get; set; }
28         public string eesnimi { get; set; }
29         public string perenimi { get; set; }
30         public string parool { get; set; }
31
32         public string kasutajagrupp_id { get; set; }
33
34         public string kasutajanimi { get; set; }
35     }
36     public class Soogikoht
37     {
38         public string soogikoht_id { get; set; }
39         public string nimi { get; set; }
40         public string aadress { get; set; }
41         public string koordinaatX { get; set; }
42         public string koordinaatY { get; set; }
43         public string toidud_id { get; set; }
44
45         public string nimetus { get; set; }
46     }
47     public class Toit
48     {
49         public string toit_id { get; set; }
50         public string nimetus { get; set; }
51         public string hind1 { get; set; }
52         public string hind2 { get; set; }
53         public string tyyp { get; set; }
54         public string kehtibkuni { get; set; }
```

```

55     public string toit { get; set; }
56     public string id { get; set; }
57
58
59
60     public string toidukoht_id { get; set; }
61
62     public string soogikoht_id { get; set; }
63
64     public string kirjeldus { get; set; }
65 }
66 public class Stat
67 {
68     public string kasutajanimi { get; set; }
69     public string algus { get; set; }
70     public string lopp { get; set; }
71 }
72 public class WebService1 : System.Web.Services.WebService
73 {
74     // SQL connection string
75     private string meCon = WebConfigurationManager.ConnectionStrings[
76         "ApplicationServices"].ConnectionString;
77
78     [WebMethod] //kasutajate näitamine
79     public System.Collections.ObjectModel.Collection<Kasutaja> naitaKasutajad()
80     {
81         System.Collections.ObjectModel.Collection<Kasutaja> kasutajad = new
82             Collection<Kasutaja>();
83
84         SqlConnection con = new SqlConnection(meCon);
85         SqlCommand com = new SqlCommand(
86             "SELECT kasutajad.kasutajad_id, kasutajad.eesnimi, kasutajad.perenimi,
87             kasutajad.kasutajagrupp_id, kasutajad.parool, kasutajad.kasutajanimi
88             FROM Kasutajad", con);
89
90         com.Connection = con;
91         com.Connection.Open();
92         SqlDataReader rdr = com.ExecuteReader();
93
94         while (rdr.Read())
95         {
96             kasutajad.Add(new Kasutaja()
97             {
98                 id = rdr["kasutajad_id"].ToString(),
99                 eesnimi = rdr["eesnimi"].ToString(),
100                 perenimi = rdr["perenimi"].ToString(),
101                 kasutajagrupp_id = rdr["kasutajagrupp_id"].ToString(),
102                 kasutajanimi = rdr["kasutajanimi"].ToString(),
103                 parool = rdr["parool"].ToString()
104             });
105         }
106         com.Connection.Close();
107         return kasutajad;
108     }
109 }

```

```

106
107 [WebMethod] //meetod kyllalise loomiseks, kasutajagrupi_id pannakse automaatselt
108 3 (kylaline)
109 public string looKylaline(string kasutajanimi, string eesnimi, string perenimi,
110 string parool)
111 {
112     if (kasutajanimi == "" || eesnimi == "" || perenimi == "" || parool == "")
113     {
114         return "Sisesta kõik andmed!";
115     }
116
117     string parooliSalastus = salastaPw(parool);
118     SqlConnection con = new SqlConnection(meCon);
119     //con.Open();
120
121     SqlCommand com =
122         new SqlCommand("INSERT INTO Kasutajad (kasutajanimi, eesnimi, perenimi,
123         parool,kasutajagrupp_id) VALUES (@kasutajanimi, @eesnimi, @perenimi,
124         @parooliSalastus,@kasutajagrupp_id )", con);
125
126     com.Parameters.AddWithValue("@kasutajanimi", kasutajanimi);
127     com.Parameters.AddWithValue("@eesnimi", eesnimi);
128     com.Parameters.AddWithValue("@perenimi", perenimi);
129     com.Parameters.AddWithValue("@parooliSalastus", parooliSalastus);
130     com.Parameters.AddWithValue("@kasutajagrupp_id", 3);
131
132     try
133     {
134         SqlCommand com1 =
135         new SqlCommand("SELECT kasutajanimi FROM Kasutajad WHERE kasutajanimi =
136         @kasutajanimi", con);
137         com1.Parameters.AddWithValue("@kasutajanimi", kasutajanimi);
138
139         com1.Connection.Open();
140
141         SqlDataReader rdr = com1.ExecuteReader();
142
143         while (rdr.Read())
144         {
145             string dbkasutajanimi = (string)rdr["kasutajanimi"];
146             com1.Connection.Close();
147
148             if (kasutajanimi == dbkasutajanimi)
149             {
150                 throw new SoapException("Selline kasutajanimi on juba olemas",
151                 null);
152             }
153         }
154     }
155     finally
156     {
157         con.Close();
158     }
159 }

```

```

155
156     try
157     {
158         con.Open();
159         com.ExecuteNonQuery();
160         con.Close();
161
162         int kasutajad_id = kysiKasutajaID(kasutajanimi);
163         DateTime kehtibkuni = DateTime.MaxValue;
164
165         SqlCommand com2 =
166         new SqlCommand("INSERT INTO KasutajaKoond (kasutajagrupp_id,
kasutajad_id, kehtibkuni) VALUES (@kasutajagrupp_id, @kasutajad_id,
@kehtibkuni)", con);
167         com2.Connection.Open();
168
169         com2.Parameters.AddWithValue("@kasutajad_id", kasutajad_id);
170         com2.Parameters.AddWithValue("@kasutajagrupp_id", 3);
171         com2.Parameters.AddWithValue("@kehtibkuni", kehtibkuni);
172         com2.ExecuteNonQuery();
173         com2.Connection.Close();
174
175
176     }
177     catch (SqlException e)
178     {
179         return e.Message.ToString();
180     }
181     finally
182     {
183         con.Close();
184     }
185     return "Kasutaja lisati!";
186
187 }
188
189
190 [WebMethod] //meetod kasutaja id küsimine tabelist Kasutajad
191 private int kysiKasutajaID(string kasutajanimi)
192 {
193
194     SqlConnection con = new SqlConnection(meCon);
195     SqlCommand com = new SqlCommand("SELECT kasutajad_id FROM Kasutajad WHERE
kasutajanimi = @kasutajanimi", con);
196     com.Connection.Open();
197     com.Parameters.AddWithValue("@kasutajanimi", kasutajanimi);
198     SqlDataReader rdr = com.ExecuteReader();
199
200     rdr.Read();
201     int kasutajad_id = (int)rdr["kasutajad_id"];
202     com.Connection.Close();
203     return kasutajad_id;
204 }
205
206 [WebMethod] //parooli salastamise funktsioon

```

```

207 private String salastaPw(string tekst)
208 {
209     string vastus = "";
210     try
211     {
212         System.Security.Cryptography.SHA1 hash = System.Security.Cryptography.
SHA1.Create();
213         System.Text.ASCIIEncoding encoder = new System.Text.ASCIIEncoding();
214         byte[] combined = encoder.GetBytes(tekst);
215         hash.ComputeHash(combined);
216         vastus = Convert.ToBase64String(hash.Hash);
217     }
218     catch (Exception exept)
219     {
220         string strerr = "Viga salastamisel: " + exept.Message;
221     }
222     return vastus;
223 }
224
225 enum Liik
226 {
227     administraator, myyja, kylaline
228 }
229
230 [WebMethod] // kasutaja loomine
231 public string kasutajaLoomine(string kasutajanimi, string eesnimi, string
perenimi, int kasutajagrupp_id, string parool, string session_key)
232 {
233
234     if (kontrolliSessiooni(session_key))
235     {
236         SessiooniUuendamine(session_key, DateTime.Now.AddMinutes(10));
237         if (kontrolliOigusi(session_key, Liik.administraator))
238         {
239
240
241             if (kasutajanimi == "" || eesnimi == "" || perenimi == "" ||
kasutajagrupp_id == null || parool == "")
242             {
243                 return "Puudulikud andmed!";
244             }
245
246             string parooliSalastus = salastaPw(parool);
247             SqlConnection con = new SqlConnection(meCon);
248             //con.Open();
249
250             SqlCommand com =
251                 new SqlCommand("INSERT INTO Kasutajad (kasutajanimi, eesnimi,
perenimi, parool, kasutajagrupp_id) VALUES (@kasutajanimi,
@eesnimi, @perenimi, @parooliSalastus, @kasutajagrupp_id )", con);
252
253             com.Parameters.AddWithValue("@kasutajanimi", kasutajanimi);
254             com.Parameters.AddWithValue("@eesnimi", eesnimi);
255             com.Parameters.AddWithValue("@perenimi", perenimi);
256             com.Parameters.AddWithValue("@parooliSalastus", parooliSalastus);

```

```

257         com.Parameters.AddWithValue("@kasutajagrupp_id", kasutajagrupp_id);
258     try
259     {
260         SqlCommand com3 =
261     new SqlCommand("SELECT kasutajanimi FROM Kasutajad WHERE kasutajanimi =
262     @kasutajanimi", con);
263         com3.Parameters.AddWithValue("@kasutajanimi", kasutajanimi);
264
265         com3.Connection.Open();
266
267         SqlDataReader rdr = com3.ExecuteReader();
268
269         while (rdr.Read())
270         {
271             string dbkasutajanimi = (string)rdr["kasutajanimi"];
272             com3.Connection.Close();
273
274             if (kasutajanimi == dbkasutajanimi)
275             {
276                 throw new SoapException("Kasutajanimi on juba olemas",
277                 null);
278             }
279         }
280     finally
281     {
282         con.Close();
283     }
284
285     try
286     {
287         con.Open();
288         com.ExecuteNonQuery();
289         con.Close();
290
291         int kasutaja_id = kysiKasutajaID(kasutajanimi);
292         DateTime kuni = DateTime.MaxValue;
293
294         SqlCommand com2 =
295     new SqlCommand("INSERT INTO KasutajaKoond (kasutajagrupp_id,
296     kasutaja_id, kehtibkuni) VALUES (@kasutajagrupp_id,
297     @kasutaja_id, @kehtibkuni)", con);
298         com2.Connection.Open();
299
300         com2.Parameters.AddWithValue("@kasutaja_id", kasutaja_id);
301         com2.Parameters.AddWithValue("@kasutajagrupp_id",
302         kasutajagrupp_id);
303         com2.Parameters.AddWithValue("@kehtibkuni", kehtibkuni);
304         com2.ExecuteNonQuery();
305         com2.Connection.Close();
306
307     }
308     catch (SqlException e)
309     {

```

```

307         return e.Message.ToString();
308     }
309     finally
310     {
311         con.Close();
312     }
313     return "Kasutaja lisatud!";
314 }
315 else
316     throw new SoapException("Pole piisavalt õigusi", null);
317 }
318 else
319     throw new SoapException("Sessioon on aegunud", null);
320 }
321
322
323 [WebMethod] //parooli kontrollimine
324 public bool kontrolliParool(string parool, string kasutajanimi)
325 {
326     if (parool.Length == 0 || kasutajanimi.Length == 0)
327         return false;
328     SqlConnection con = new SqlConnection(meCon);
329     SqlCommand com = new SqlCommand("SELECT parool FROM Kasutajad WHERE
kasutajanimi = '" + kasutajanimi + "'", con);
330     com.Connection.Open();
331     try
332     {
333         SqlDataReader rdr = com.ExecuteReader();
334         rdr.Read();
335         string dbParool = rdr["parool"].ToString();
336         string test = salastaPw("asdf");
337         if (dbParool != null && dbParool != "")
338         {
339             if (dbParool == salastaPw(parool))
340                 return true;
341             else
342                 return false;
343         }
344         else
345             return false;
346     }
347     catch (Exception e)
348     {
349         return false;
350     }
351 }
352
353
354 [WebMethod] //Sessiooni uuendamise
355 public void SessiooniUuendamine(string session_key, DateTime lopp)
356 {
357
358     DateTime uuenda = DateTime.Now.AddMinutes(10);
359     SqlConnection con = new SqlConnection(meCon);
360

```

```

361 SqlCommand com = new SqlCommand("UPDATE Session SET loopp = @loopp WHERE
    session_key = @session_key", con);
362 con.Open();
363
364 com.Parameters.AddWithValue("@loopp", loopp);
365 com.Parameters.AddWithValue("@session_key", session_key);
366 com.ExecuteNonQuery();
367 con.Close();
368 }
369
370
371 [WebMethod] //logIn
372 public string logIn(string kasutajanimi, string parool)
373 {
374     if (kontrolliParool(parool, kasutajanimi))
375     {
376         Random uusSuva = new Random();
377         byte[] byteMassiiv = new byte[16];
378         uusSuva.NextBytes(byteMassiiv);
379
380         Guid session_key = new Guid(byteMassiiv);
381         sessiooniSalvestus(kysiKasutajaID(kasutajanimi), session_key.ToString());
382         return session_key.ToString();
383     }
384
385     else
386         throw new SoapException("Vale kasutajanimi või parool", null);
387
388 }
389
390 [WebMethod] // logOut
391 public void logOut(string session_key)
392 {
393     if (kontrolliSessiooni(session_key))
394     {
395         SessiooniUuendamine(session_key, DateTime.Now);
396     }
397     else
398         return;
399 }
400
401 [WebMethod] // kasutaja kustutamine
402 public string kasutajaKustutamine(int id, string session_key)
403 {
404     if (kontrolliSessiooni(session_key))
405     {
406         if (kontrolliOigusi(session_key, Liik.myyja))
407         {
408             if (id == null)
409             {
410                 return "Kasutaja ID puudu!";
411             }
412
413             SqlConnection con = new SqlConnection(meCon);
414             SqlCommand com = new SqlCommand("DELETE FROM Kasutajad WHERE id = "

```

```

        + id, con);

415
416         int affected = 0;
417
418         try
419         {
420             con.Open();
421             affected = com.ExecuteNonQuery();
422         }
423         catch (SqlException e)
424         {
425             return e.Message.ToString();
426         }
427         finally
428         {
429             con.Close();
430         }
431         if (affected > 0)
432         {
433             return "Kasutaja kustutatud!";
434         }
435         else
436             throw new SoapException("Sellise id-ga kasutajat pole", null);
437     }
438     else
439         throw new SoapException("Pole piisavalt õigusi", null);
440 }
441 else
442     throw new SoapException("Sessioon on aegunud", null);
443 }
444 // Kontrollib kas kasutajal on piisavaid õiguseid antud toiminguid teha
445 //vaatab seotud kasutaja ning kasutajagrupi id-si tabelis kasutajaRollis
446 //vaatab kas rollis olemise aeg on veel kehtiv
447
448 [WebMethod]
449 public int ShowID(string session_key) {
450     if (kontrolliOigusi(session_key, Liik.myyja)){
451         return 2;
452     }
453     if (kontrolliOigusi(session_key, Liik.administraator))
454     {
455         return 1;
456     }
457     if (kontrolliOigusi(session_key, Liik.kylaline))
458     {
459         return 3;
460     }
461
462     return 0;
463 }
464
465
466 [WebMethod]
467 private bool kontrolliOigusi(string session_key, Liik liik)
468 {

```

```

469         if (kontrolliSessiooni(session_key))
470         {
471             int kasutajagrupp_id = otsiKasutajaGrupiID(liik.ToString());
472             int kasutajad_id = otsiKasutajaIDsession(session_key);
473
474             SqlConnection con = new SqlConnection(meCon);
475             SqlCommand com = new SqlCommand("SELECT kehtibkuni FROM KasutajaKoond
WHERE kasutajagrupp_id = @kasutajagrupp_id AND " +
476                 " kasutajad_id = @kasutajad_id", con);
477             com.Connection.Open();
478             com.Parameters.AddWithValue("@kasutajagrupp_id", kasutajagrupp_id);
479             com.Parameters.AddWithValue("@kasutajad_id", kasutajad_id);
480
481             SqlDataReader rdr = com.ExecuteReader();
482             while (rdr.Read())
483             {
484                 DateTime? kuni = (DateTime?)rdr["kehtibkuni"];
485
486                 if (kuni == null)
487                 {
488                     return true;
489                 }
490                 else
491                 {
492                     if (kuni > DateTime.Now)
493                         return true;
494
495                     else
496                         return false;
497                 }
498             }
499             return false;
500         }
501         else
502             return false;
503     }
504
505
506     [WebMethod] //kontrollib kas sessioon on loodud ning aktiivne
507     public bool kontrolliSessiooni(string session_key)
508     {
509         SqlConnection con = new SqlConnection(meCon);
510         SqlCommand com = new SqlCommand("SELECT lopp FROM Session WHERE session_key
= @session_key", con);
511         com.Connection.Open();
512         com.Parameters.AddWithValue("@session_key", session_key);
513
514         SqlDataReader rdr = com.ExecuteReader();
515         rdr.Read();
516         DateTime lopp = (DateTime)rdr["lopp"];
517
518         if (lopp > DateTime.Now)
519         {
520             return true;
521         }

```

```

522         else
523             return false;
524     }
525
526     //meetod kasutaja id küsimiseks tabelist sessioon
527     private int otsiKasutajaIDsession(string session_key)
528     {
529
530         SqlConnection con = new SqlConnection(meCon);
531         SqlCommand com = new SqlCommand("SELECT kasutajad_id FROM Session WHERE
532 session_key = @session_key", con);
533         com.Connection.Open();
534         com.Parameters.AddWithValue("@session_key", session_key);
535         SqlDataReader rdr = com.ExecuteReader();
536
537         rdr.Read();
538         int kasutajad_id_session = (int)rdr["kasutajad_id"];
539         com.Connection.Close();
540         return kasutajad_id_session;
541     }
542
543     //Kategooria nimetus ID-ks @return int
544     [WebMethod]
545     private string kasutajaGrupiNimed()
546     {
547         SqlConnection con = new SqlConnection(meCon);
548         SqlCommand com = new SqlCommand("SELECT liik FROM Kasutajagrupp ", con);
549         com.Connection.Open();
550
551         SqlDataReader rdr = com.ExecuteReader();
552
553         rdr.Read();
554         string liigid = (string)rdr["liik"];
555         com.Connection.Close();
556         return liigid;
557     }
558
559     private void sessiooniSalvestus(int kasutajad_id, string session_key)
560     {
561         DateTime lopp = new DateTime();
562         DateTime algus = new DateTime();
563         algus = DateTime.Now;
564         lopp = DateTime.Now.AddMinutes(10);
565
566         SqlConnection con = new SqlConnection(meCon);
567         con.Open();
568         SqlCommand com =
569             new SqlCommand("INSERT INTO Session (algus, lopp, kasutajad_id,
570 session_key) VALUES (@algus, @lopp, @kasutajad_id, @session_key)", con);
571
572         com.Parameters.AddWithValue("@algus", algus);
573         com.Parameters.AddWithValue("@lopp", lopp);
574         com.Parameters.AddWithValue("@session_key", session_key);
575         com.Parameters.AddWithValue("@kasutajad_id", kasutajad_id);

```

```

575
576         com.ExecuteNonQuery();
577
578         con.Close();
579     }
580
581     [WebMethod]
582     public int otsiKasutajaGrupiID(string liik)
583     {
584         SqlConnection con = new SqlConnection(meCon);
585         SqlCommand com = new SqlCommand("SELECT kasutajagrupp_id FROM Kasutajagrupp
WHERE liik = @liik", con);
586         com.Connection.Open();
587         com.Parameters.AddWithValue("@liik", liik);
588         SqlDataReader rdr = com.ExecuteReader();
589
590         rdr.Read();
591         int kasutajagrupp_id = (int)rdr["kasutajagrupp_id"];
592         com.Connection.Close();
593         return kasutajagrupp_id;
594     }
595
596     [WebMethod] //näitab söögikohtasid
597     public System.Collections.ObjectModel.Collection<Soogikoht> naitaToidukohad()
598     {
599         System.Collections.ObjectModel.Collection<Soogikoht> soogikoht = new
Collection<Soogikoht>();
600
601         SqlConnection con = new SqlConnection(meCon);
602         SqlCommand com = new SqlCommand(
603             "SELECT soogikoht.soogikoht_id, soogikoht.nimi, soogikoht.nimi,
soogikoht.aadress, soogikoht.koordinaatX, soogikoht.koordinaatY FROM
Soogikoht", con);
604         com.Connection = con;
605         com.Connection.Open();
606         SqlDataReader rdr = com.ExecuteReader();
607
608         while (rdr.Read())
609         {
610             soogikoht.Add(new Soogikoht ()
611             {
612                 soogikoht_id = rdr["soogikoht_id"].ToString(),
613                 nimi = rdr["nimi"].ToString(),
614                 aadress = rdr["aadress"].ToString(),
615                 koordinaatX = rdr["koordinaatX"].ToString(),
616                 koordinaatY = rdr["koordinaatY"].ToString(),
617
618                 });
619             }
620         com.Connection.Close();
621         return soogikoht;
622     }
623
624     // Lisab toidu andmebaasi
625     // @return String

```

```

626 [WebMethod]
627 public string lisaToit(int toit_id, int soogikoht_id, string tyyp, string
nimetus, string kirjeldus, double hind1, double hind2, string kehtibkuni, string
session_key)
628 {
629     if (kontrolliSessiooni(session_key))
630     {
631         SessiooniUuendamine(session_key, DateTime.Now.AddMinutes(10));
632         if (kontrolliOigusi(session_key, Liik.myyja))
633         {
634             if (nimetus == "")
635             {
636                 return "Toidu nimetus puudub!";
637             }
638
639             SqlConnection con = new SqlConnection(meCon);
640             SqlCommand com =
641                 new SqlCommand("INSERT INTO Toit ( soogikoht_id, tyyp, nimetus,
kirjeldus, hind1, hind2, kehtibkuni ) VALUES (@soogikoht_id,
@tyyp, @nimetus, @kirjeldus, @hind1, @hind2, @kehtibkuni)", con);
642             // com.Parameters.AddWithValue("@toit_id", toit_id);
643             com.Parameters.AddWithValue("@soogikoht_id", soogikoht_id);
644             com.Parameters.AddWithValue("@tyyp", tyyp);
645             com.Parameters.AddWithValue("@nimetus", nimetus);
646             com.Parameters.AddWithValue("@kirjeldus", kirjeldus);
647             com.Parameters.AddWithValue("@hind1", hind1);
648             com.Parameters.AddWithValue("@hind2", hind2);
649             com.Parameters.AddWithValue("@kehtibkuni", kehtibkuni);
650
651             try
652             {
653                 con.Open();
654                 com.ExecuteNonQuery();
655             }
656             catch (SqlException e)
657             {
658                 return e.Message.ToString();
659             }
660             finally
661             {
662                 con.Close();
663             }
664             return "Toode lisatud!";
665         }
666         else
667             throw new SoapException("Pole piisavalt õigusi", null);
668     }
669     else
670         throw new SoapException("Sessioon on aegunud", null);
671 }
672
673 //kustutab toidu andmbebaasist
674 //@return String
675 [WebMethod]
676 public string kustutaToit(int id, string session_key)

```

```

677     {
678         if (kontrolliSessiooni(session_key))
679         {
680             SessiooniUuendamine(session_key, DateTime.Now.AddMinutes(10));
681             if (kontrolliOigusi(session_key, Liik.myyja))
682             {
683
684                 SqlConnection con = new SqlConnection(meCon);
685                 SqlCommand com = new SqlCommand("DELETE FROM Toit WHERE toit_id =' "
686                     + id + "'", con);
687
688                 int affected = 0;
689
690                 try
691                 {
692                     con.Open();
693                     affected = com.ExecuteNonQuery();
694                 }
695                 catch (SqlException e)
696                 {
697                     return e.Message.ToString();
698                 }
699                 finally
700                 {
701                     con.Close();
702                 }
703                 if (affected > 0)
704                 {
705                     return "Toit kustutatud!";
706                 }
707                 else
708                     throw new SoapException("Sellise id-ga toitu pole", null);
709             }
710             else
711                 throw new SoapException("Pole piisavalt õigusi", null);
712         }
713         else
714             throw new SoapException("Sessioon on aegunud", null);
715     }
716
717     [WebMethod] // soogikoha toitide näitamine
718     public System.Collections.ObjectModel.Collection<Toit> naitaToidud()
719     {
720         System.Collections.ObjectModel.Collection<Toit> toit = new Collection<Toit>
721             >();
722         SqlConnection con = new SqlConnection(meCon);
723         SqlCommand com = new SqlCommand(
724             "SELECT toit.soogikoht_id, toit.toit_id,
725             toit.nimetus,toit.tyyp,toit.kirjeldus, toit.hind1, toit.hind2,
726             toit.kehtibkuni FROM Toit ", con);
727         com.Connection = con;
728         com.Connection.Open();
729         SqlDataReader rdr = com.ExecuteReader();
730         while (rdr.Read())
731         {

```

```
728         toit.Add(new Toit ()
729         {
730             soogikoht_id = rdr["soogikoht_id"].ToString(),
731             toit_id = rdr["toit_id"].ToString(),
732             nimetus = rdr["nimetus"].ToString(),
733             tyypp = rdr["tyyp"].ToString(),
734             kirjeldus = rdr["kirjeldus"].ToString(),
735             hind1 = rdr["hind1"].ToString(),
736             hind2 = rdr["hind2"].ToString(),
737             kehtibkuni = rdr["kehtibkuni"].ToString(),
738
739
740             });
741     }
742     com.Connection.Close();
743     return toit;
744
745 }
746
747 public object kehtibkuni { get; set; }
748
749 public object KasutajaGrupp_id { get; set; }
750
751 public string toit_id { get; set; }
752
753 public object kasutajad_id { get; set; }
754 }
755
756
757
758
759 }
760
761
762
```