

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using System.Xml.Linq;
```

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "Service" in code, svc and config file together.

```
public class Service : IService
```

```
{
    private string xmlTee;

    public Service ()
    {
        xmlTee = ("C:/Documents and Settings/Vivika/My Documents/Visual Studio 2008/WebSites/TeenusK/sessiInfo.xml");
    }
}
```

// Kysib toodete kohta kogu info

```
public List<TooteInfo> KysiTooteInfo()
{
    string fileName = "C:/Documents and Settings/Vivika/My Documents/Visual Studio
2008/WebSites/TeenusK/menyy.xml";
    XDocument xmlDoc = XDocument.Load(fileName);
    List<TooteInfo> tootedRitta
    =
    (from toode in xmlDoc.Descendants("toode")
    select new TooteInfo()
    {
        ryhm = toode.Element("ryhm").Value,
        nimi = toode.Element("nimi").Value,
        hind = Int32.Parse(toode.Element("hind").Value),
    }).ToList<TooteInfo>();
    return tootedRitta;
}
```

// Kysib tooteryhmad

```
public List<TooteInfo> KysiRyhmad()
{
    string fileName = "C:/Documents and Settings/Vivika/My Documents/Visual Studio
2008/WebSites/TeenusK/menyy.xml";
    XDocument xmlDoc = XDocument.Load(fileName);
    List<TooteInfo> ryhmadRitta
    =
    (from toode in xmlDoc.Descendants("toode")
    select new TooteInfo()
    {
        ryhm = toode.Element("ryhm").Value
    }).ToList<TooteInfo>();
    List<TooteInfo> ryhmad = KustutaDuublid.KustutaDbl(ryhmadRitta);
    return ryhmad;
}
```

// Kysib päevamenyy tooted rühma järgi

```
public List<TooteInfo> KysiRyhmaJ(string Ryhm)
{
```

```

string fileName = "C:/Documents and Settings/Vivika/My Documents/Visual Studio
2008/WebSites/TeenusK/today.xml";
XDocument xmlDoc = XDocument.Load(fileName);
List<TooteInfo> tootedRitta
=
(from toode in xmlDoc.Descendants("toode")
where toode.Element("ryhm").Value.Equals(Ryhm)
select new TooteInfo()
{
ryhm = toode.Element("ryhm").Value,
nimi = toode.Element("nimi").Value,
hind = Int32.Parse(toode.Element("hind").Value),
}).ToList<TooteInfo>());
return tootedRitta;
}

// Loob teenusele uue kasutaja ja lisab tema andmed andmebaasi
public bool LooKasutaja(string kasutajanimi, string parool, string eesnimi, string perenimi)
{
if (kasutajanimi != String.Empty && parool != String.Empty)
{
return Kasutaja.looKasutaja(kasutajanimi, parool, eesnimi, perenimi);
}
else
{
throw Vead.RaiseException("looKasutaja", "http://tempuri.org/", "USER CREATION ERROR: Kasutajanimi või
parool puudub",
"1", "looKasutaja", Vead.FaultCode.Client);
}
}

// Loob kohvikule uue päevamenyy, nõuab kohvikupidaja rolli.
public string LooMenyy(string sessID, List<TooteInfo> tooted, DateTime kuupaev, bool aktiivne)
{
if (Kasutaja.sessionNotExpired(sessID, xmlTee))
{
if (Kasutaja.onVajalikusRollis(sessID, xmlTee, Kasutaja.Roll.kohvikupidaja))
{
string user = Kasutaja.getUserNameFromSessID(sessID, xmlTee);
string meetod = "looMenyy";
Ylevaade yv = new Ylevaade(user, meetod, "");
yv.lisaStatistikaXML();
uuendaSess(sessID);
return Haldus.muudaMenyy(tooted, aktiivne);
}
else
{
throw Vead.RaiseException("", "http://tempuri.org/", "NOT AUTHORIZED: Ei ole vajalikke õigusi", "1",
"", Vead.FaultCode.Client);
}
}
else
{
throw Vead.RaiseException("", "http://tempuri.org/", "SESSION EXPIRED: Kasutaja sessioon on aegunud", "1",
"", Vead.FaultCode.Client);
}
}
}

```

```

// Lisab kasutaja rollidesse. Nõuab admini rolli
public string LisaKasutajaRolli(string sessID, string rolliUsername, Kasutaja.Roll roll, string kommentaar, DateTime
alates, DateTime kuni)
{
    if (Kasutaja.sessionNotExpired(sessID, xmlTee))
    {
        if (Kasutaja.onVajalikusRollis(sessID, xmlTee, Kasutaja.Roll.admin))
        {
            string user = Kasutaja.getUserNameFromSessID(sessID, xmlTee);
            string meetod = "lisaKasutajaRolli";
            Ylevaade stat = new Ylevaade(user, meetod, "");
            stat.lisaStatistikaXML();
            // Lisab kasutaja rolli
            Kasutaja.lisaKasutajaRolli(rolliUsername, roll, kommentaar, alates, kuni);
            // Uuendab sessiooni
            uuendaSess(sessID);
            return "Kasutaja ID: " + rolliUsername + " on lisatud rolli ID: " + roll.ToString();
        }
        else
        {
            throw Vead.RaiseException("lisaKasutajaRolli", "http://tempuri.org/", "NOT AUTHORIZED: Ei ole vajalikke
6igusi", "1",
                "lisaKasutajaRolli", Vead.FaultCode.Client);
        }
    }
    else
    {
        throw Vead.RaiseException("lisaKasutajaRolli", "http://tempuri.org/", "SESSION EXPIRED: Kasutaja sessioon on
aegunud", "1",
                "lisaKasutajaRolli", Vead.FaultCode.Client);
    }
}

// Kustutab kasutaja, nõuab admini rolli
public bool KustutaKasutaja(string sessID, string username)
{
    if (Kasutaja.sessionNotExpired(sessID, xmlTee))
    {
        if (Kasutaja.onVajalikusRollis(sessID, xmlTee, Kasutaja.Roll.admin))
        {
            string user = Kasutaja.getUserNameFromSessID(sessID, xmlTee);
            string meetod = "adminKustutaKasutaja";
            Ylevaade yv = new Ylevaade(user, meetod, "");
            yv.lisaStatistikaXML();
            uuendaSess(sessID);
            return Kasutaja.kustutaKasutaja(username);
        }
        else
        {
            throw Vead.RaiseException("", "http://tempuri.org/", "NOT AUTHORIZED: Ei ole vajalikke 6igusi",
                "1", "", Vead.FaultCode.Client);
        }
    }
    else
    {
        throw Vead.RaiseException("", "http://tempuri.org/", "SESSION EXPIRED: Kasutaja sessioon on aegunud", "1",
                "", Vead.FaultCode.Client);
    }
}

```

```
}

// logib sisse
public string logIn(string username, string password)
{
    Guid sessID;
    if (Kasutaja.authUser(username, password))
    {
        sessID = Kasutaja.generateGuid(username, xmlTee);
        return sessID.ToString();
    }
    else
        return Guid.Empty.ToString();
}

// logib välja
public void logOut(string sessID)
{
    Kasutaja.updateSession(sessID, DateTime.Now, xmlTee);
}

// uuendab sessiooni
public void uuendaSess(string sessID)
{
    DateTime aeg = DateTime.Now.AddMinutes(15);

    Kasutaja.updateSession(sessID, aeg, xmlTee);
}
}
```