```
 1 import java.io.*;
 2 import java.sql.*;
 3 import javax.servlet.*;
 4 import javax.servlet.http.*;
 5
 6 public class LuoTunnus extends HttpServlet {
 7
 8    final String dbDriver="org.postgresql.Driver";
 9    final String dbServer ="jdbc:postgresql://localhost:10388/tsoha";
10    final String dbUser= "avtanska";        // replace with your db user account
11    final String dbPassword ="postgres"; // replace with your password
12
13   public void service(HttpServletRequest req, HttpServletResponse res)
14     throws ServletException, IOException {
15     ServletOutputStream out;
16     res.setContentType("text/html");
17     out= res.getOutputStream();
18
19     String tunnus = req.getParameter("tunnus");
20     String salasana = req.getParameter("salasana");
21     String varmistus = req.getParameter("varmistus");
22     String sukunimi = req.getParameter("sukunimi");
23     String etunimi = req.getParameter("etunimi");
24     String sposti = req.getParameter("sposti");
25     String puh = req.getParameter("puh");
26
27     out.println("<html><head><title>Database query from DB (tsoha)</title>" +
28                 "<link rel='stylesheet' style='text/css' " +
29                 "href='http://db.cs.helsinki.fi/u/avtanska/tsoha/" +
30                 "perus.css'></head><body>");
31
32     Connection con=null;
33     con= createDbConnection(dbDriver,dbServer,dbUser,dbPassword,out);
34     if (con==null) {
35        out.println("</body></html>");
36        return;
37     }
38
39     String insertHenkilö =
40         "INSERT INTO henkilö VALUES('" + tunnus + "','" + sukunimi +
41          "','" + etunimi + "','" + sposti + "','" + puh + "')";
42
43     String insertSalasana =
44         "INSERT INTO salasana VALUES('" + tunnus + "','" + salasana + "')";
45
46     Statement stmt = null;
47     ResultSet rs = null;
48     try {
49       stmt = con.createStatement();
50
51       /*
52        * Tarkistetaan pakollisten kenttien tiedot
53        */
54
55       if (tunnus.length() == 0) {
56         out.println("<p class='virhe'>Anna tunnus.</p>");
57       }
58       else if (tunnus.length() > 8) {
59         out.println("<p class='virhe'>Tunnus voi olla korkeintaan " +
60                     " <b>8</b> merkkiä pitkä.</p>");
61       }
62       else {
63         if (etunimi.length() == 0 || sukunimi.length() == 0 ||
64             salasana.length() == 0 || varmistus.length() == 0) {
65           out.println("<p><b>Täytä pakolliset kentät.</b></p>");
66         }
67         else if (etunimi.length() > 30) {
68           out.println("<span class='virhe'>Etunimi liian pitkä</span>");
69         }
```

```
70          else if (sukunimi.length() > 50) {
71            out.println("<span class='virhe'>Sukunimi liian pitkä</span>");
72          }
73          else if (sposti.length() > 60) {
74            out.println("<span class='virhe'>Sähköposti liian pitkä</span>");
75          }
76          else if (puh.length() > 20) {
77            out.println("<span class='virhe'>Puhelinnumero liian pitkä</span>");
78          }
79          else {
80
81            /*
82             * Tarkistetaan onko haluttu tunnus jo olemassa
83             */
84
85            rs = stmt.executeQuery("SELECT tunnus FROM henkilö WHERE tunnus='" +
86                                   tunnus + "'");
87
88            if (rs.next()) {
89              out.println("<p><b>Tunnus jo olemassa</b></p>");
90            }
91            else if (!salasana.equals(varmistus)) {
92              out.println("<p><b>Antamasi salasanan varmistus ei täsmää</b></p>");
93            }
94            else {
95
96              /*
97               * Kaikki kunnossa, luodaan henkilö ja salasana
98               */
99
100             int count = stmt.executeUpdate(insertHenkilö);
101             count = stmt.executeUpdate(insertSalasana);
102             out.println("<h3>Tunnuksesi luonti onnistui</h3>" +
103                         "<p>Voit kirjautua sisään tunnuksellasi.</p>");
104           }
105         }
106       }
107       while(rs.next()) {
108            out.println(rs.getString("tunnus"));
109
110       }
111
112     } catch (SQLException ee) {
113           out.println("Tietokantavirhe "+ee.getMessage());
114     } finally {
115        try {
116           if (rs!=null) rs.close();
117           if (stmt!=null) stmt.close();
118           con.close();
119        } catch(SQLException e) {
120           out.println("An SQL Exception was thrown.");
121        }
122     }
123
124     out.println("</body></html>");
125   }
126
127   private Connection createDbConnection(
128       String dbDriver, String dbServer, String dbUser, String dbPassword,
129       ServletOutputStream out) throws IOException {
130
131    // establish a database connection
132       try{
133           Class.forName(dbDriver);               // load driver
134       } catch (ClassNotFoundException e) {
135             out.println("Couldn't find driver "+dbDriver);
136             return null;
137       }
138       Connection con=null;
```

```
139         try {
140            con = DriverManager.getConnection(dbServer,dbUser,dbPassword);
141         } catch (SQLException se) {
142               out.println("Couldn\'t get connection to "+dbServer+
143                           " for "+ dbUser+"<br>");
144               out.println(se.getMessage());
145         }
146         return con;
147      }
148
149   }
```