```
/********
Shawn Embry
Dentist Project
CIST2373
********/
package Servlets;

import Business.Patient;
import Business.Appointments;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;


@WebServlet(name = "patientUpdate", urlPatterns =
{
    "/patientUpdate"
})
/*************************************************************
 *  Directs from updatePatient.jsp. Retrieves data from entry fields and updates
 *  Patient info in database. Will also direct from createAppointment.jsp to store
 *  new appointment data into Appointments table.
 *************************************************************/
public class patientUpdate extends HttpServlet
{

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException, SQLException, ClassNotFoundException

    {
        /*****************************
         * DRIVER STEPS
         *****************************/
                    //Load Driver - Step #1

                    Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
                //Get Connection from the Driver - Step #2
                    Connection con;
                    con=DriverManager.getConnection("jdbc:ucanaccess://C:/Users/kerds/OneDrive/School/Java3/Dentist/DentistOfficeMDB.mdb");
                //Create a Statement - Step #3
                    Statement stmt = con.createStatement();
                //Execute Statement - Step #4
                    String dentSQL = ("select * from Patients");
                    System.out.println("The SQL statement is: " + dentSQL + "\n"); // Echo For debugging
                    ResultSet drs = stmt.executeQuery(dentSQL);
                //Process through the Data - Step #5
                    response.setContentType("text/html;charset=UTF-8");
                    PrintWriter out = response.getWriter();
        //END DRIVER STATMENTS

        //Put Object in session
        HttpSession session;
        session = request.getSession();             // Retrieves session
        Object pid = session.getAttribute("patID");
        String patId = (String) pid;
        Patient p1;
        p1 = new Patient();
        p1.selectDB(patId);
        p1.display();

        //Gets input from input boxes(update patient)
```

```
        String fn = request.getParameter("firstName");
        String ln = request.getParameter("lastName");
        String em = request.getParameter("email");
        String ad = request.getParameter("addr");
        //Gets input from dropdown menu (create appointment)
        String pro = request.getParameter("procedure");
        String dt = request.getParameter("dateTime");
        String di = request.getParameter("dentist");
        System.out.println(dt + " " + p1.getPatientID() + "  " + di + " " + pro);


        //Sets input as long as field is not empty
        if (fn != null && !fn.isEmpty()) {
            p1.setPatientFirstName(fn);
        }
        if (ln != null && !ln.isEmpty()) {
            p1.setPatientLastName(ln);
        }
        if (em != null && !em.isEmpty()) {
            p1.setEmail(em);
        }
        if (ad != null && !ad.isEmpty()) {
                    p1.setAddress(ad);
                }
       /* if (pro != null && !pro.isEmpty()) {
            p1.app.setProcCode(pro);
                }
        if (dt != null && !dt.isEmpty()) {
                    p1.app.setAptDT(dt);
                }
        if (di != null && !di.isEmpty()) {
                    p1.app.setDentId(di);
                }*/

        p1.updateDB();
        System.out.println(dt);
        if(dt != null)
        {
        p1.app.insertDB(dt, p1.getPatientID(), di, pro); // Updates Appointment table
        }

        System.out.println(fn);

        {
            RequestDispatcher rd = request.getRequestDispatcher("/index.jsp");
            rd.forward(request, response);
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException
    {
        try
        {
            processRequest(request, response);
        } catch (SQLException ex)
        {
            Logger.getLogger(patientUpdate.class.getName()).log(Level.SEVERE, null, ex);
        } catch (ClassNotFoundException ex)
        {
            Logger.getLogger(patientUpdate.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
```

```
    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException
    {
        try
        {
            processRequest(request, response);
        } catch (SQLException ex)
        {
            Logger.getLogger(patientUpdate.class.getName()).log(Level.SEVERE, null, ex);
        } catch (ClassNotFoundException ex)
        {
            Logger.getLogger(patientUpdate.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo()
    {
        return "Short description";
    }// </editor-fold>

}
```