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

import Business.Patient;
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 = "newAppointmentServlet", urlPatterns =
{
    "/newAppointmentServlet"
})
/*************************************************************
 *  Directs from displayPatient new button. Retreieves patient in session info and
 *  checks if there is an entry for appointment date/time. If blank it can proceed to
 *  create a new appointment(createAppointment.jsp).
 ************************************************************/
public class newAppointmentServlet extends HttpServlet
{

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

         * 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 patSQL = ("select * from Patients");
                    System.out.println("The SQL statement is: " + patSQL + "\n"); // Echo For debugging
                    ResultSet rs = stmt.executeQuery(patSQL);
                //Process through the Data - Step #5
                    response.setContentType("text/html;charset=UTF-8");
        //END DRIVER STATMENTS
        response.setContentType("text/html;charset=UTF-8");
        //Put Object in session
        HttpSession session;
        session = request.getSession();
        Object pid = session.getAttribute("patID");
        String patId = (String) pid;
        Patient p1;
        p1 = new Patient();
        p1.selectDB(patId);
        p1.display();

        String pataSQL = ("select * from Appointments where patId = '" + pid + "'");
        ResultSet rt = stmt.executeQuery(pataSQL);
        //p1.app.display();
```

```
        //Create new appointment
        String x = p1.app.getAptDT();
        if (x == ""){
            RequestDispatcher rd = request.getRequestDispatcher("createAppointment.jsp");
            rd.forward(request, response);

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

try (PrintWriter out = response.getWriter())
        {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet newAppointmentServlet</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet newAppointmentServlet at " + request.getContextPath() + "</h1>");
            out.println("</body>");
            out.println("</html>");
        }

    }

    // <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 (ClassNotFoundException ex)
        {
            Logger.getLogger(newAppointmentServlet.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex)
        {
            Logger.getLogger(newAppointmentServlet.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 (ClassNotFoundException ex)
        {
            Logger.getLogger(newAppointmentServlet.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex)
        {
            Logger.getLogger(newAppointmentServlet.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>

}
```