```
1
2  import com.cete.dynamicpdf.*;
3  import com.cete.dynamicpdf.pageelements.*;
4  import java.io.FileNotFoundException;
5  import java.io.IOException;
6  import java.sql.*;
7  import java.text.SimpleDateFormat;
8  import java.util.Date;
9  import java.util.GregorianCalendar;
10 import javax.servlet.ServletConfig;
11 import javax.servlet.ServletException;
12 import javax.servlet.ServletOutputStream;
13 import javax.servlet.http.HttpServlet;
14 import javax.servlet.http.HttpServletRequest;
15 import javax.servlet.http.HttpServletResponse;
16
17
18
19 public class ContactList extends HttpServlet {
20
21     // Top, bottom, left and right margins of report
22     private float margin;
23     // Height of the header
24     private float headerHeight;
25     // Height of the footer
26     private float footerHeight;
27     // Size of paper to use
28     private PageDimensions pageSize;
29     // Bottom Y coordinate for the body of the report
30     private float bodyBottom;
31
32     // Current page that elements are being added to
33     private Page currentPage;
34     // Template for header and footer elements
35     private Template template;
36     // Current Y coordinate where elements are being added
37     private float currentY;
38     // Used to control the alternating background
39     private boolean alternateBG;
40     // Used to test for grouping
41     private String tmpChar = "";
42
43     ServletOutputStream sOut;
44     Connection connection;
45
46     public void init(ServletConfig servletConfig) throws ServletException {
47         super.init(servletConfig);
48         margin = 36;
49         headerHeight = 74;
```

```
50          footerHeight = 14;
51          pageSize = new PageDimensions(PageSize.LETTER,
52                                        PageOrientation.LANDSCAPE, margin);
53          bodyBottom = pageSize.getHeight() - (margin * 2) - footerHeight;
54
55
56      }
57      public void doGet(HttpServletRequest req, HttpServletResponse res)
58                      throws IOException,ServletException {
59          CeTeConnection ceTe = (CeTeConnection)getServletContext().getAttribute("cetecon");
60          connection = ceTe.getConnection();
61          sOut = res.getOutputStream();
62          currentY = 0;
63          alternateBG = false;
64          template = new Template();
65
66          // Create a document and set it's properties
67      Document objDocument = new Document();
68      objDocument.setCreator("ContactList.java");
69      objDocument.setAuthor("ceTe Software");
70      objDocument.setTitle("Contact List");
71          objDocument.setTemplate(template);
72
73      // Adds elements to the template
74      setHeaderTemplate();
75      setFooterTemplate();
76          ResultSet data = getContactListData();
77
78      // Builds the report
79      buildDocument(objDocument, data);
80
81          // Outputs the ContactList to the current web page
82          objDocument.drawToWeb(req, res, sOut, "ContactList.pdf");
83          ceTe.close();
84          sOut.close();
85      }
86
87      private void setHeaderTemplate() {
88      // Adds header elements to the template
89          try {
90          template.getElements().add(new Image(getServletContext()
91                      .getRealPath("/images/DPDFLogo.png"), 0, 0, 0.21f));
92          } catch (FileNotFoundException ex) {
93              ex.printStackTrace(System.err);
94          }
95      template.getElements().add(new Label("Northwind Traders", 0, 0, 720, 18,
96                                  Font.getHelveticaBold(), 18, TextAlign.CENTER));
97      template.getElements().add(new Label("Contact List", 0, 21, 720, 12,
98                                  Font.getHelvetica(), 12, TextAlign.CENTER));
```

```
        GregorianCalendar gc = new GregorianCalendar();
        Date date = gc.getTime();
        SimpleDateFormat sd1 = new SimpleDateFormat("dd MMM yyyy, H:mm:ss E");
    template.getElements().add(new Label(sd1.format(date), 0, 36, 720, 12,
                                Font.getHelvetica(), 12, TextAlign.CENTER));
    template.getElements().add(new Rectangle(0, 56, 720, 16,
                                new WebColor("0000A0"),new WebColor("0000A0")));
    template.getElements().add(new Label("Cust ID", 2, 57, 58, 12,
                                Font.getHelveticaBold(), 12, TextAlign.LEFT,
                                Grayscale.getWhite()));
    template.getElements().add(new Label("Company", 62, 57, 156, 12,
                                Font.getHelveticaBold(), 12, TextAlign.LEFT,
                                Grayscale.getWhite()));
    template.getElements().add(new Label("Contact", 222, 57, 156, 12,
                                Font.getHelveticaBold(), 12, TextAlign.LEFT,
                                Grayscale.getWhite()));
    template.getElements().add(new Label("Title", 362, 57, 156, 12,
                                Font.getHelveticaBold(), 12, TextAlign.LEFT,
                                Grayscale.getWhite()));
    template.getElements().add(new Label("Phone", 522, 57, 86, 12,
                                Font.getHelveticaBold(), 12, TextAlign.LEFT,
                                Grayscale.getWhite()));
    template.getElements().add(new Label("Fax", 622, 57, 86, 12,
                                Font.getHelveticaBold(), 12, TextAlign.LEFT,
                                Grayscale.getWhite()));
    }

    private void setFooterTemplate() {
    // Adds footer elements to the template
    PageNumberingLabel pageNumLabel = new PageNumberingLabel("Page " +
            "%%CP(i)%% of %%TP(i)%%", 0, bodyBottom + 5, 720, 10,
            Font.getHelvetica(), 10, TextAlign.CENTER);
    template.getElements().add(pageNumLabel);
    }

    private void buildDocument(Document document, ResultSet data) {
    // Builds the PDF document with data from the ResultSet
    addNewPage(document);
        try {
        while (data.next()) {
            // Add current record to the document
           addRecord(document, data);
            }
        } catch (SQLException ex) {
            ex.printStackTrace(System.err);
        }
    }

    private void addRecord(Document document, ResultSet data) {
```

```
    // Creates TextAreas that are expandable
        try {

            String tmpCustomerId = data.getString(1);
        String tmpStr = data.getString(2);

            TextArea companyName = new TextArea(data.getString(3), 62,
                                        currentY + 3, 156, 11,
                                        Font.getTimesRoman(), 11);
        TextArea contactName = new TextArea(data.getString(4), 222,
                                        currentY + 3, 136, 11,
                                        Font.getTimesRoman(), 11);
            TextArea contactTitle = new TextArea(data.getString(5), 362,
                                        currentY + 3, 156, 11,
                                        Font.getTimesRoman(), 11);




            float requiredHeight = setExpandableRecords(document,
                                                    companyName,contactName,
                                                    contactTitle,tmpStr);


            // Creates non expandable Labels
        Label customerID = new Label(tmpCustomerId, 2, currentY + 3,
                                     58, 11,Font.getTimesRoman(), 11);


        Label phone = new Label(data.getString(6), 522, currentY + 3, 96, 11,
                                Font.getTimesRoman(), 11);
            String str = data.getString(7);
        Label fax = new Label(data.wasNull()?"":str, 622, currentY + 3, 96,
                              11, Font.getTimesRoman(), 11);

        // Adds alternating background if required
        if (alternateBG) {
            currentPage.getElements().add(new Rectangle(0, currentY, 720,
                                    requiredHeight + 6, new WebColor("E0E0FF"),
                                    new WebColor("E0E0FF")));
        }
        // Toggles alternating background
        alternateBG = !alternateBG;

        // Adds elements to the current page
        currentPage.getElements().add(customerID);
        currentPage.getElements().add(companyName);
        currentPage.getElements().add(contactName);
        currentPage.getElements().add(contactTitle);
```

```
        currentPage.getElements().add(phone);
        currentPage.getElements().add(fax);

        // increments the current Y position on the page
        currentY += requiredHeight + 6;
        } catch (SQLException ex) {
            ex.printStackTrace(System.err);
        }
    }

    private float setExpandableRecords(Document document,
                                       TextArea companyName, TextArea contactName,
                                       TextArea contactTitle, String tmpStr) {
    // Gets the maximum height requred of the three TextAreas
    float requiredHeight = getMaxRecordHeight(companyName, contactName,
                                              contactTitle);

    // Add space for the section header if required
        float sectionHeaderHeight = 0;
        if (!tmpChar.equals(tmpStr.substring(0,1))) {
            sectionHeaderHeight = 26;
        }
    // Add a new page if needed
    if (bodyBottom < currentY + requiredHeight + sectionHeaderHeight + 4) {
        addNewPage(document);
        if (sectionHeaderHeight == 0) {
        // Update Y coordinate of TextArea when placed on the new page
        companyName.setY(currentY + 1);
        contactName.setY(currentY + 1);
        contactTitle.setY(currentY + 1);
        }
    }

        if (sectionHeaderHeight > 0) {
        addSectionHeader(tmpStr);
        companyName.setY(currentY + 3);
        contactName.setY(currentY + 3);
        contactTitle.setY(currentY + 3);
    }

    return requiredHeight;
    }

    private void addSectionHeader(String tmpStr) {
        tmpChar = tmpStr.substring(0, 1);
        currentPage.getElements().add(new Label("- " + tmpChar + " -", 0,
                                      currentY + 6, 720, 18, Font.getHelveticaBold(),
                                      18, TextAlign.CENTER));

```

```
    currentY += 26;
    alternateBG = false;
    }

    private float getMaxRecordHeight(TextArea companyName, TextArea contactName,
                                     TextArea contactTitle) {
    // Returns the maximum required height of the three TextAreas
    float requiredHeight = 11;
    float requiredHeightB = 0;

    requiredHeight = companyName.getRequiredHeight();
    requiredHeightB = contactName.getRequiredHeight();
    if (requiredHeightB > requiredHeight) {
            requiredHeight = requiredHeightB;
        }
    requiredHeightB = contactTitle.getRequiredHeight();
    if (requiredHeightB > requiredHeight) {
            requiredHeight = requiredHeightB;
        }

    if (requiredHeight > 11) {
        companyName.setHeight(requiredHeight);
        contactName.setHeight(requiredHeight);
        contactTitle.setHeight(requiredHeight);
    }

    return requiredHeight;
    }


    private void addNewPage(Document document) {
    // Adds a new page to the document
    currentPage = new Page(pageSize);

        currentY = headerHeight;
    alternateBG = false;

    document.getPages().add(currentPage);
    }

    private ResultSet getContactListData() {
    ResultSet data = null;
        // Creates a ResultSet for the report
    try {
            PreparedStatement ps = connection.prepareStatement("SELECT Customer"+
                                   "ID, FirstL = Left(CompanyName,1), Company"+
                                   "Name, ContactName, ContactTitle, Phone, "+
                                   "Fax FROM Customers ORDER BY CompanyName");

```

```
295         data = ps.executeQuery();
296         } catch (SQLException ex) {
297             ex.printStackTrace(System.err);
298         }
299     return data;
300     }
301
302
303
304 }
```