

ItemDAO.java

```

1 package model;
2
3 import java.sql.Connection;
4
5
6
7
8
9
10 class ItemDAO
11 {
12     private static final String DB_URL = "jdbc:derby://localhost:64413/
EECS;user=student;password=secret";
13     private static ItemDAO instance = null;
14
15     private ItemDAO() throws Exception
16     {
17         Class.forName("org.apache.derby.jdbc.ClientDriver").newInstance();
18     }
19
20     static synchronized ItemDAO getInstance() throws Exception
21     {
22         if(instance == null)
23             instance = new ItemDAO();
24         return instance;
25     }
26
27     private class GeneratePreparedStatement
28     {
29         private Connection conn;
30         PreparedStatement ps;
31
32         GeneratePreparedStatement(String query) throws Exception
33         {
34             conn = DriverManager.getConnection(DB_URL);
35             ps = conn.prepareStatement("SET SCHEMA ROUMANI");
36             ps.executeUpdate();
37             ps.close();
38             ps = conn.prepareStatement(query);
39         }
40
41         PreparedStatement getPs()
42         {
43             return ps;
44         }
45
46         void closeConnection() throws Exception
47         {
48             if (!ps.isClosed()) this.ps.close();
49             if (!conn.isClosed()) this.conn.close();
50         }
51
52

```

ItemDAO.java

```

53     }
54     List<ItemBean> getItems(String catalog, String orderBy) throws Exception
55     {
56         GeneratePreparedStatement psGenerator = null;
57         ResultSet rs = null;
58         try
59         {
60             //building sql
61             String sql = "SELECT * FROM ITEM";
62             sql = sql + " WHERE CATID=" + Integer.parseInt
(catalog);
63             sql = orderBy == null? sql : sql + " ORDER BY " + orderBy;
64             System.out.println("sql=" + sql);
65             psGenerator = new GeneratePreparedStatement(sql);
66             PreparedStatement ps = psGenerator.getPs();
67             rs = ps.executeQuery();
68             List<ItemBean> items = new ArrayList<ItemBean>();
69             while(rs.next())
70             {
71                 //ItemBean(String name, String number, double price, int
catalog, String unit)
72                 String name = rs.getString("NAME");
73                 String number = rs.getString("NUMBER");
74                 double price = rs.getDouble("PRICE");
75                 int catalogID = rs.getInt("CATID");
76                 String unit = rs.getString("UNIT");
77
78                 items.add(new ItemBean(name, number, price, catalogID,
unit));
79             }
80             if (items.size() < 1)
81                 throw new Exception("No Items Found");
82             return items;
83         }
84         catch (Exception e)
85         {
86             throw e;
87         }
88         finally
89         {
90             if(rs != null && !rs.isClosed()) rs.close();
91             if(psGenerator != null ) psGenerator.closeConnection();
92         }
93     }
94
95
96     ItemBean getItem(String number) throws Exception
97     {

```

ItemDAO.java

```

98      GeneratePreparedStatement psGenerator = null;
99      ResultSet rs = null;
100     try
101     {
102         String sql = "SELECT * FROM ITEM WHERE NUMBER='" + number + "'";
103
104         ps.setString(1, number);
105         System.out.println(ps.toString());
106         rs = ps.executeQuery();
107
108         psGenerator = new GeneratePreparedStatement(sql);
109         PreparedStatement ps = psGenerator.getPs();
110         rs = ps.executeQuery();
111         if (rs.getRow() < 0)
112             throw new Exception ("No item with number " + number);
113         ItemBean result = null;
114         if(rs.next())
115         {
116             String name = rs.getString("NAME");
117             String itemNumber = rs.getString("NUMBER");
118             double price = rs.getDouble("PRICE");
119             int catalogID = rs.getInt("CATID");
120             String unit = rs.getString("UNIT");
121
122             result = new ItemBean(name, itemNumber, price, catalogID,
unit);
123         }
124         return result;
125     }
126     catch (Exception e)
127     {
128         throw e;
129     }
130     finally
131     {
132         if(rs != null && !rs.isClosed()) rs.close();
133         if(psGenerator != null ) psGenerator.closeConnection();
134     }
135 }
136
137 List<CatalogBean> getCatalogs() throws Exception
138 {
139     GeneratePreparedStatement psGenerator = null;
140     ResultSet rs = null;
141     try
142     {
143
144         String sql = "SELECT * FROM CATEGORY";

```

ItemDAO.java

```
145
146     psGenerator = new GeneratePreparedStatement(sql);
147     PreparedStatement ps = psGenerator.getPs();
148     rs = ps.executeQuery();
149     List<CatalogBean> result = new ArrayList<CatalogBean>();
150     while(rs.next())
151     {
152         String name = rs.getString("NAME");
153         String description = rs.getString("DESCRIPTION");
154         int id = rs.getInt("ID");
155         result.add(new CatalogBean(name, description, id));
156     }
157     return result;
158 }
159 catch (Exception e)
160 {
161     throw e;
162 }
163 finally
164 {
165     if(rs != null && !rs.isClosed()) rs.close();
166     if(psGenerator != null ) psGenerator.closeConnection();
167 }
168 }
169
170 }
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
```

ItemDAO.java

193
194
195
196
197
198
199
200
201
202