```java
package DBLayer;
import java.sql.*;
import java.util.ArrayList;
import ModelLayer.*;

public class DBColumn implements IFDBColumn
{
   private Connection con;

   public DBColumn()
   {
         con = DbConnection.getInstance().getDBcon();
   }

   public ArrayList<Column> getAllColumn(boolean
   retriveAssociation)
   {
         return miscWhere("", retriveAssociation);
   }

   public Column findColumn(int colNo, boolean
   retriveAssociation)
   {
         String wClause = " colNo = ' " + colNo + "'";
         return singleWhere(wClause, retriveAssociation);
   }

   public int createColumn(Column column) throws Exception
   {
         int rc = -1;
         String query = "INSERT INTO mfColumn(colNo, hallNr) VALUES
         ('"+
                     column.getColNo() + "', '" +
                     column.getHallNr() + "')";
         System.out.println("insert : " + query);

         try
         {
               Statement stmt = con.createStatement();
               stmt.setQueryTimeout(5);
               rc = stmt.executeUpdate(query);
               stmt.close();
         }
         catch(SQLException ex)
         {
               System.out.println("Column er ikke oprettet");
               throw new Exception("Column er ikke tilføjet");
         }
         return (rc);
   }

   public int deleteColumn(int colNo)
   {
         int rc =-1;

         String qry = "DELETE FROM mfColumn WHERE colNo = '" +
         colNo + "'";
         System.out.println(qry);
         try
         {
               Statement stmt = con.createStatement();
               stmt.setQueryTimeout(5);
               rc = stmt.executeUpdate(qry);
               stmt.close();
```

```
65          }
66           catch(Exception ex)
67           {
68                      System.out.println("Delete exception in Column db: "+
69                      ex);
70           }
71          return(rc);
72    }
73
74    public int updateColumn(Column column)
75    {
76          Column colkobj = column;
77          int rc= -1;
78
79          String qry = "UPDATE mfColumn SET" +
80                      "colNo = '" + colkobj.getColNo() +"', " +
81                      "hallNr ='" + colkobj.getHallNr() + "', ";
82
83
84                      System.out.println("Update query: " + qry);
85                      try
86                      {
87                            Statement stmt = con.createStatement();
88                            stmt.setQueryTimeout(5);
89                            rc = stmt.executeUpdate(qry);
90
91                            stmt.close();
92                      }
93                      catch(Exception ex)
94                      {
95                            System.out.println("Update exception in Column db:
" +
96                            ex);
97                      }
98                      return(rc);
99    }
100
101   private String buildquery(String wClause)
102   {
103         String query = "SELECT colNo, hallNr FROM mfColumn";
104         if(wClause.length()>0)
105               query=query + " WHERE " + wClause;
106         return query;
107   }
108
109   private ArrayList<Column> miscWhere(String wClause, boolean
110   retrieveAssociation)
111   {
112         ResultSet results;
113         ArrayList<Column> colums = new ArrayList<Column>();
114
115         String query = buildquery(wClause);
116
117         try
118         {
119               Statement stmt = con.createStatement();
120               stmt.setQueryTimeout(5);
121               results = stmt.executeQuery(query);
122
123         while(results.next())
124         {
125               Column colobj = new Column();
126               colobj = buildColums(results);
127               colums.add(colobj);
```

```
128         }
129         stmt.close();
130         return colums;
131         }
132         catch(Exception e)
133         {
134               System.out.println(e.getMessage());
135               return null;
136         }
137
138
139   }
140
141   private Column singleWhere(String wClause, boolean
142   retrieveAssociation)
143   {
144         ResultSet results;
145         Column colobj = new Column();
146         String query = buildquery(wClause);
147         System.out.println(query);
148
149         try
150         {
151               Statement stmt = con.createStatement();
152               stmt.setQueryTimeout(5);
153               results = stmt.executeQuery(query);
154
155               if(results.next())
156               {
157                     colobj = buildColums(results);
158               }
159               stmt.close();
160         }
161         catch(Exception e)
162         {
163               System.out.println("Query exception: " + e);
164         }
165         return colobj;
166   }
167
168   private Column buildColums(ResultSet results)
169   {
170         Column colObj = new Column();
171         try
172         {
173               colObj.setColNo(results.getInt("colNo"));
174               colObj.setHallNr(results.getInt("hallNr"));
175         }
176         catch(Exception e)
177         {
178               System.out.println("Error in building column");
179         }
180         return colObj;
181
182   }
183
184
185
186 }
```