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

public class DBCage implements IFDBCage
{
   private Connection con;
   //FISK
   public DBCage()
   {
         con = DbConnection.getInstance().getDBcon();
   }

   public ArrayList<Cage> getAllCages(boolean
   retriveAssociation)
   {
         return miscWhere("", retriveAssociation);
   }

   public Cage findCage(int cageNo, boolean
   retrieveAssociation)
   {
         String wClause = " cageNo = '" + cageNo + "'";
         return singleWhere(wClause, retrieveAssociation);
   }

   //Creates a new cage
   public int insertCage(Cage cag) throws Exception
   {
         int rc = -1;
         String query = "INSERT INTO mfCage(cageNo, colNr, CageType)
         VALUES ('"+
                     cag.getCageNo() + "', '" +
                     cag.getColNr() + "', '" +
                     cag.getCageType() + "')";
         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("Cage er ikke oprettet");
               throw new Exception("Cage er ikke tilføjet");
         }
         return (rc);

   }

   public int deleteCage(int cageNo)
   {
         int rc =-1;

         String qry = "DELETE FROM mfCage WHERE cageNo = '" +
         cageNo + "'";
         System.out.println(qry);
         try
         {
               Statement stmt = con.createStatement();
```

```
65                stmt.setQueryTimeout(5);
66                rc = stmt.executeUpdate(qry);
67                stmt.close();
68          }
69           catch(Exception ex)
70           {
71                      System.out.println("Delete exception in Cage db: "+ex)
72                      ;
73           }
74          return(rc);
75    }
76
77    public int updateCage(Cage cage)
78    {
79          Cage cagekobj = cage;
80          int rc= -1;
81
82          String qry = "UPDATE mfCage SET" +
83                      "cageNo = '" + cagekobj.getCageNo()+"', " +
84                      "colNr ='" + cagekobj.getColNr() + "', " +
85                      "CageType ='" + cagekobj.getCageType() + "', ";
86
87                      System.out.println("Update query: " + qry);
88                      try
89                      {
90                            Statement stmt = con.createStatement();
91                            stmt.setQueryTimeout(5);
92                            rc = stmt.executeUpdate(qry);
93
94                            stmt.close();
95                      }
96                      catch(Exception ex)
97                      {
98                            System.out.println("Update exception in Mink db:
" + ex)
99                            ;
100                     }
101                     return(rc);
102   }
103
104   //Method will build cage objectfg
105   private Cage buildCage(ResultSet results)
106   {
107         Cage cageObj = new Cage();
108         try
109         {
110               cageObj.setCageNo(results.getInt("cageNo"));
111               cageObj.setColNr(results.getInt("colNr"));
112               cageObj.setCageType(results.getString("CageType"));
113         }
114         catch(Exception e)
115         {
116               System.out.println("Error in building cage");
117         }
118         return cageObj;
119
120   }
121
122   private String buildquery(String wClause)
123   {
124         String query = "SELECT cageNo, colNr, CageType FROM
125         mfCage";
126         if(wClause.length()>0)
127               query=query + " WHERE " + wClause;
```

```
128         return query;
129   }
130
131   private Cage singleWhere(String wClause, boolean
132   retrieveAssociation)
133   {
134         ResultSet results;
135         Cage cageobj = new Cage();
136         String query = buildquery(wClause);
137         System.out.println(query);
138
139         try
140         {
141               Statement stmt = con.createStatement();
142               stmt.setQueryTimeout(5);
143               results = stmt.executeQuery(query);
144
145               if(results.next())
146               {
147                     cageobj = buildCage(results);
148               }
149               stmt.close();
150         }
151         catch(Exception e)
152         {
153               System.out.println("Query exception: " + e);
154         }
155         return cageobj;
156   }
157
158   private ArrayList<Cage> miscWhere(String wClause, boolean
159   retrieveAssociation)
160   {
161         ResultSet results;
162         ArrayList<Cage> cages = new ArrayList<Cage>();
163
164         String query = buildquery(wClause);
165
166         try
167         {
168               Statement stmt = con.createStatement();
169               stmt.setQueryTimeout(5);
170               results = stmt.executeQuery(query);
171
172         while(results.next())
173         {
174               Cage cageobj = new Cage();
175               cageobj = buildCage(results);
176               cages.add(cageobj);
177         }
178         stmt.close();
179         return cages;
180         }
181         catch(Exception e)
182         {
183               System.out.println(e.getMessage());
184               return null;
185         }
186
187
188   }
189
190 }
```