```java
package DBLayer;
import ModelLayer.*;

import java.sql.*;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;

public class DBDiseaseReport implements IFDBDiseaseReport {

   private Connection con;


// Creates a new instance of DBPlasmDisease
   public DBDiseaseReport()
   {
         con = DbConnection.getInstance().getDBcon();
   }

   @Override
   public ArrayList<DiseaseReport> getAllDiseaseReport(boolean
   retriveAssociation)
   {
         return miscWhere("", retriveAssociation);
   }

   @Override
   public DiseaseReport searchDiseaseReportByCageNumber(int
   cageNo, boolean retriveAssociation)
   {
         String wClause = "  cageNumber = '" + cageNo + "'";
         return singleWhere(wClause, retriveAssociation);
   }

   @Override
   public int insertDiseaseReport(DiseaseReport diseasereport)
   {
       int rc = -1;
        PreparedStatement pstmt = null;
        String insert = "INSERT INTO mfDiseaseReport(
        cageNumber, diseaseId, reportDate)"+"values(?,?,?)";
        System.out.println(insert);
        try
        {
            pstmt = con.prepareStatement(insert);
            pstmt.setInt( 1, diseasereport.getCageNo() );
            pstmt.setInt( 2, diseasereport.getDiseaseId() );
            pstmt.setString( 3, getTodaysDate() );
            rc = pstmt.executeUpdate();
        }
        catch(SQLException sqlE)
        {
            System.out.println(sqlE.getMessage());
            return rc;
        }
        catch(Exception e)
        {
         System.out.println(e.getMessage());
         return rc;
        }

        return rc;
   }

```

```
65
66    //SingleWhere is used when we only select one DiseaseReport.
67
68    private DiseaseReport singleWhere(String wClause, boolean
69    retrieveAssociation)
70    {
71          ResultSet results;
72          DiseaseReport diseasereportObj = null;
73          String query =  buildQuery(wClause);
74          System.out.println(query);
75          try
76          {
77                //read the health status from the database.
78                Statement stmt = con.createStatement();
79                stmt.setQueryTimeout(5);
80                results = stmt.executeQuery(query);
81                if( results.next() )
82                      diseasereportObj = buildDiseaseReport(results);
83                stmt.close();
84          }
85          catch(Exception ex)
86          {
87                System.out.println(ex.getMessage());
88                return null;
89          }
90
91          return diseasereportObj;
92    }
93
94
95
96 //michWere is used whenever we want to select more than one
97 row of health status of a cage.
98 private ArrayList<DiseaseReport> miscWhere(String wClause,
99 boolean retrieveAssociation)
100 {
101   ResultSet results;
102   ArrayList<DiseaseReport> list = new ArrayList <
103   DiseaseReport>();
104
105   String query = buildQuery(wClause);
106   System.out.println(query);
107   try
108   {
109         // read the health status from the database
110       Statement stmt = con.createStatement();
111       stmt.setQueryTimeout(5);
112       results = stmt.executeQuery(query);
113
114       while(results.next())
115       {
116         list.add(buildDiseaseReport(results));
117       }
118       stmt.close();
119
120       return list;
121   }
122   catch(Exception ex)
123   {
124         System.out.println(ex.getMessage());
125         return null;
126   }
127 }
128
```

```
129
130   //method to build the query
131   private String buildQuery(String wClause)
132   {
133         String query = "SELECT cageNumber, diseaseId, reportDate
134         FROM mfDiseaseReport";
135         if (wClause.length() > 0)
136               query = query + " WHERE " + wClause;
137
138         return query;
139   }
140
141
142   //method to build health status object.
143   private DiseaseReport buildDiseaseReport(ResultSet results)
144   {
145         try
146         {
147             DiseaseReport diseasereportObj = new DiseaseReport(
148                     results.getInt( "cageNumber" ),
149                     results.getInt( "diseaseId" ),
150                     results.getString( "reportDate" ));
151
152             return diseasereportObj;
153         }
154         catch (Exception ex)
155         {
156               System.out.println(ex.getMessage());
157               return null;
158         }
159
160   }
161
162
163
164   @Override
165   public int deleteDiseaseReportWithCageNo(int cageNo)
166   {
167         int rc=-1;
168         String query="delete from mfDiseaseReport where cageNumber
169         = '" + cageNo + "'";
170         System.out.println(query);
171         try
172         {
173               // delete from health status
174               Statement stmt = con.createStatement();
175               stmt.setQueryTimeout(5);
176               rc = stmt.executeUpdate(query);
177               stmt.close();
178         }//end try
179             catch(Exception ex)
180             {
181                     System.out.println("Delete exception in
182                     DiseaseReport db: "+ex.getMessage());
183             }
184
185         return(rc);
186   }
187
188     public String getTodaysDate()
189     {
190         Calendar calendar = Calendar.getInstance();
191         SimpleDateFormat dateFormat = new SimpleDateFormat(
192         "dd/MM/yyyy");
```

```
193         return dateFormat.format(calendar.getTime());
194     }
195 
196 }
```