

```

1
2 package Business;
3
4 /*****
5  Shawn Embry
6  Dentist Project
7  CIST2373
8  *****/
9 import java.sql.*;
10
11
12 /*****
13  * Appointments class communicates with Appointments
14  table from database.
15  *****/
16
17 /*****Properties*****/
18 private String apptDateTime, patId, dentId,
19 procCode;
20
21 public Procedures pr1 = new Procedures();
22 public Dentist dr1 = new Dentist();
23 AppointmentList bList = new AppointmentList();
24 /*****Constructors*****/
25 public Appointments()
26 {
27     apptDateTime = "";
28     patId = "";
29     dentId = "";
30     procCode = "";
31 }
32
33 public Appointments(String dt, String pid, String
34 did, String pc)

```

```

31     {
32         this.apptDateTime = dt;
33         this.patId = pid;
34         this.dentId = did;
35         this.procCode = pc;
36     }
37     /*****Behaviors*****/
38     public void setAptDT(String dt){apptDateTime=dt;}
39     public String getAptDT() {return apptDateTime; }
40
41     public void setPatId(String pid){patId=pid;}
42     public String getPatId() {return patId; }
43
44     public void setDentId(String did){dentId=did;}
45     public String getDentId() {return dentId; }
46
47     public void setProcCode(String pc){procCode=pc;}
48     public String getProcCode() {return procCode; }
49
50     /*****DB
Behaviors*****/
51
52
53     * Selects appointment from Appointmets table filtered
by patient ID.
54
55     public void selectDB(String i)
56     {
57         try
58         {
59             //Load Driver - Step #1
60             Class.forName("net.ucanaccess.jdbc.
UcanaccessDriver");

```

```

61          //Get Connection from the Driver - Step #2
62          Connection c1;
63          c1=DriverManager.getConnection("jdbc:
ucanaccess://C:
/Users/kerds/OneDrive/School/Java3/Dentist/DentistOfficeMDB
.mdb");
64
65          //Create a Statement - Step #3
66          Statement stmt = c1.createStatement();
67          ResultSet rs = stmt.executeQuery("Select
* from Appointments where patId = '" + i + "'");
68
69          //Process ResultSet
70          rs.next();
71          apptDateTime = rs.getString(1);
72          patId = rs.getString(3);
73          dentId = rs.getString(4);
74          procCode = rs.getString(2);
75          c1.close();
76          String pc = procCode;
77          pr1.selectDB(pc);
78      }
79          catch(Exception se)
80          {
81              System.out.println(se);
82          }
83      } //end selectDB()
84
85
86  * Inserts new Appointment into Appointments table
87
88  public void insertDB(String dt, String pid, String did,
String pc){

```

```
89
90
91     setAptDT(dt);
92     setPatId(pid);
93     setDentId(did);
94     setProcCode(pc);
95     this.apptDateTime = dt;
96     this.patId = pid;
97     this.dentId = did;
98     this.procCode = pc;
99
100    try{
101        Class.forName("net.ucanaccess.jdbc.
UcanaccessDriver");
102        Connection con = DriverManager.
getConnection("jdbc:ucanaccess://C:
/Users/kerds/OneDrive/School/Java3/Dentist/DentistOfficeMDB
.mdb");
103
104        Statement stmt = con.createStatement();
105        String sql = "Insert into Appointments
values ('"+getAptDT()+"', '"+
106
107        '"+getProcCode()+"', '"+
108
109        '"+getPatId()+"', '"+
110
111        '"+getDentId()+"') ";
112
113        System.out.println(sql);
114        int n1 = stmt.executeUpdate(sql);
115        if (n1==1)
116            System.out.println("INSERT
Successful!!!");
117        else
118            System.out.println("INSERT
FAILED*****");
```

```

116         con.close();
117     }
118     catch (Exception e1) {
119         System.out.println(e1);
120     }
121 }
122 } //end insertDB()
123
124
125 /*****
126  * Updates existing Appointment table data
127  *****/
128
129 ***
130 public void updateDB() {
131     try {
132         Class.forName("net.ucanaccess.jdbc.
133         UcanaccessDriver");
134         Connection con = DriverManager.
135         getConnection("jdbc:ucanaccess://C:
136         /Users/kerds/OneDrive/School/Java3/Dentist/DentistOfficeMDB
137         .mdb");
138         Statement stmt = con.createStatement();
139         String sql = "update Appointments set
140         apptDateTime = '"+getAptDT() + "','"+
141         " patId
142         ='"+getPatId()+"',"+
143         " dentId =
144         '"+getDentId() + "','"+
145         " procCode
146         ='"+getProcCode()+"'"+
147         " where

```

```

patId='"+getPatId()+"'";
141
142         System.out.println(sql);
143         int n = stmt.executeUpdate(sql);
144         if (n==1)
145             System.out.println("UPDATE
Successful!!!");
146         else
147             System.out.println("UPDATE
FAILED*****");
148         con.close();
149     }
150     catch (Exception e1) {
151         System.out.println(e1);
152     }
153
154 } //end updateDB()
155
156
/*****
****
157  *  Displays appointment data
158
****
***
159     public void display()
160     {
161         System.out.println("Date and Time = " +
getAptDT());
162         System.out.println("Patient ID = "
+getPatId());
163         System.out.println("Dentist ID = " +
getDentId());
164         System.out.println("Procedure Code = " +
getProcCode());
165     }
166

```

```

167
/*****
****
168  *  Deletes existing appointment data
169
****
***/
170     public void deleteDB() {
171
172         try{
173             Class.forName("net.ucanaccess.jdbc.
UcanaccessDriver");
174             Connection con = DriverManager.
getConnection("jdbc:ucanaccess://C:
/Users/kerds/OneDrive/School/Java3/Dentist/DentistOfficeMDB
.mdb");
175
176             Statement stmt = con.createStatement();
177             String sql = "Delete from Appointments
where patId='"+getPatId()+"'";
178             System.out.println(sql);
179             int n = stmt.executeUpdate(sql);
180             if (n==1)
181                 System.out.println("DELETE
Successful!!!");
182             else
183                 System.out.println("DELETE
FAILED*****");
184             con.close();
185         }
186         catch(Exception e1){
187             System.out.println(e1);
188         }
189
190     } //end deleteDB()
191     public static void main(String args[]) {
192

```

```

193          *****Select and Display
test*****
194          //Appointments a1;
195          //a1 = new Appointments();
196          //a1.selectDB("A910");
197          //a1.selectDBDentist("D203");
198          // bList.displayList();
199          //a1.selectDB("A902");
200          //a1.display();
201
202          //Dentist d1 = new Dentist();
203
204          //d1.getAppointments(x);
205          //a1.pr1.display();
206
207          *****Insert test*****
208          //Appointments a2;
209          //a2 = new Appointments();
210          // a2.insertDB("May 5, 2018, 230pm", "A912",
"D203", "P321");
211
212          *****Update test*****
213          //Appointments a3 = new Appointments();
214          //a3.selectDB("A912");
215          //a3.setAptDT("May 4, 2018, 4pm");
216          //a3.setDentId("D201");
217          //a3.updateDB();
218          //a3.display();
219
220          *****Delete test*****
221          //Appointments a4 = new Appointments();
222          //a4.selectDB("D203");
223          //a4.deleteDB();
224      }
225  } //END APPOINTMENTS
226

```