Coverage Summary for Class: Product (models)

Class	Class, %	Method, %	Line, %
Product	100% (1/ 1)	89.5% (34/ 38)	82.8% (144/ 174)

```
 1  package models;
 2
 3  import java.sql.Connection;
 4  import java.sql.DriverManager;
 5  import java.sql.SQLException;
 6  import java.sql.Statement;
 7  import java.util.Date;
 8  import java.sql.ResultSet;
 9
10  /**
11   * Created by akshay on 10/29/2016.
12   */
13  public class Product {
14      private long id;
15      private String uploadedBy;
16      private String imagePath;
17      private float price;
18      private String description;
19      private Date dateUploaded;
20      private Date dateSold;
21      private float priceBought;
22      private String onlineLink;
23      private float soldPrice;
24      private int condition;
25      private int months;
26      private int category;
27      private String location;
28      private String paymentMethod;
29
30      public Product() {
31
32      }
33
34      public Product(long id, String uploadedBy, String imagePath, float price, String description,
35                     Date dateUploaded, Date dateSold, float priceBought, String onlineLink,
36                     float soldPrice, int condition, int months, int category, String location, String paymentMethod) {
37          System.out.println("Got to product constructor");
38          this.id = id;
39          this.uploadedBy = uploadedBy;
40          this.imagePath = imagePath;
41          this.price = price;
42          this.description = description;
43          this.dateUploaded = dateUploaded;
44          this.dateSold = dateSold;
45          this.priceBought = priceBought;
46          this.onlineLink = onlineLink;
47          this.soldPrice = soldPrice;
48          this.condition = condition;
49          this.months = months;
50          this.category = category;
51          this.location = location;
52          if (paymentMethod == "")
53              this.paymentMethod = "Unspecified";
54          else this.paymentMethod = paymentMethod;
55      }
56
57      public long getId() {
58
59          return id;
60      }
61
62      public void setId(long id) {
63          this.id = id;
64      }
65
66      public String getImagePath() {
67          return imagePath;
```

```
    }

    public void setImagePath(String imagePath) {
        this.imagePath = imagePath;
    }

    public float getPrice() {
        return price;
    }

    public void setPrice(float price) {
        this.price = price;
    }

    public String getPaymentMethod() {
        return paymentMethod;
    }

    public void setPaymentMethod(String paymentMethod) {
        this.paymentMethod = paymentMethod;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }


    public Date getDateUploaded() {
        return dateUploaded;
    }

    public void setDateUploaded(Date dateUploaded) {
        this.dateUploaded = dateUploaded;
    }

    public void setDateSold(Date dateSold) {
        this.dateSold = dateSold;
    }

    public String getOnlineLink() {
        return onlineLink;
    }

    public void setOnlineLink(String onlineLink) {
        this.onlineLink = onlineLink;
    }

    public float getSoldPrice() {
        return soldPrice;
    }

    public void setSoldPrice(float soldPrice) {
        this.soldPrice = soldPrice;
    }

    public int getCondition() {
        return condition;
    }

    public void setCondition(int condition) {
        this.condition = condition;
    }

    public int getMonths() {
        return months;
    }

    public String mapMonthsToString() {
        String[] months = new String[4];
        months[0] = "Less than 3 months";
        months[1] = "3-6 months";
        months[2] = "6 months - 3 years";
        months[3] = ">3 years";
        return months[this.getMonths()-1];
    }
```

```
147
148     public void setMonths(int months) {
149         this.months = months;
150     }
151     public float getPriceBought() {
152         return priceBought;
153     }
154
155     public void setPriceBought(float priceBought) {
156         this.priceBought = priceBought;
157     }
158
159     public String getUploadedBy() {
160         return uploadedBy;
161     }
162
163     public void setUploadedBy(String uploadedBy) {
164         this.uploadedBy = uploadedBy;
165     }
166     public int getCategory() {
167         return category;
168     }
169
170     public void setCategory(int category) {
171         this.category = category;
172     }
173
174     public String getLocation() {
175         return location;
176     }
177
178     public void setLocation(String location) {
179         this.location = location;
180     }
181
182     public boolean checkConditions(){
183         if(price > 999999 || price < 0 || imagePath.length()>100 || imagePath.length() == 0 || priceBought > 999999
184                 || priceBought < 0 || description.length() == 0 || description.length() > 65535 ||
185                 category > 4 || category < 1 || onlineLink.length() > 255 || condition > 5 || condition < 1 ||
186                 months > 4 || months < 1 || location.length()>255 || location.length() == 0)
187             return false;
188         return true;
189     }
190
191     public boolean addProductToDatabase() throws ClassNotFoundException {
192         return addProductToDatabase(false);
193     }
194
195     public boolean addProductToDatabase(boolean isTest) throws ClassNotFoundException {
196         String myDriver = null;
197         String myURL = null;
198         Connection conn = null;
199         boolean returnVal = true;
200         if(isTest) {
201             myDriver = "com.mysql.jdbc.Driver";
202             myURL = "jdbc:mysql://localhost:3306/mydatabase?zeroDateTimeBehavior=convertToNull";
203         }
204         else{
205             myDriver = "com.mysql.jdbc.Driver";
206             myURL = "jdbc:mysql://lionmart.cvkcqiaoutkr.us-east-1.rds.amazonaws.com:3306/lionmart";
207         }
208         try {
209             Class.forName(myDriver);
210             if(isTest)
211             {
212                 conn = DriverManager.getConnection(myURL, "root", "");
213             }
214             else
215             {
216                 conn = DriverManager.getConnection(myURL, "lionadmin", "lionlynx42");
217             }
218             Statement st = conn.createStatement();
219             st.executeUpdate("CREATE TABLE IF NOT EXISTS product (id INT PRIMARY KEY, price DECIMAL(8,2), imagepath VARCHAR(100),category INT NOT NULL,price_bought DECIMAL(8,2) NOT NULL,description TEXT NOT NULL,date_upl
220
221             java.sql.Timestamp product_timestamp = new java.sql.Timestamp(this.getDateUploaded().getTime());
222             //Check conditions before actually attempting to insert into database
223             boolean shouldInsert = checkConditions();
224             if (shouldInsert)
225                 st.executeUpdate("INSERT INTO product(id,imagepath, price, category, price_bought, description, date_upload,online_link,price_sold,product_condition,months_used,location,user_id, payment_method) VALUES ("
```

```
            else
                return false;
            //Confirm that product is, in fact, inserted into DB.
            ResultSet rs = st.executeQuery("SELECT * from product where id = "+id);
            if(!rs.next())
                returnVal = false;
            if(isTest)
                st.executeUpdate("DROP TABLE product;");
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
        return returnVal;
    }
    // the below method is made for specific test cases and is not redundant.  DO NOT DELETE.
    public boolean addProductToDatabase2(boolean isTest) throws ClassNotFoundException {
        String myDriver = null;
        String myURL = null;
        Connection conn = null;
        boolean returnVal = true;
        if(isTest) {
            myDriver = "com.mysql.jdbc.Driver";
            myURL = "jdbc:mysql://localhost:3306/mydatabase?zeroDateTimeBehavior=convertToNull";
        }
        else{
            myDriver = "com.mysql.jdbc.Driver";
            myURL = "jdbc:mysql://lionmart.cvkcqiaoutkr.us-east-1.rds.amazonaws.com:3306/lionmart";
        }
        try {
            Class.forName(myDriver);
            if(isTest)
            {
                conn = DriverManager.getConnection(myURL, "root", "");
            }
            else
            {
                conn = DriverManager.getConnection(myURL, "lionadmin", "lionlynx42");
            }
            Statement st = conn.createStatement();
            st.executeUpdate("CREATE TABLE IF NOT EXISTS product (id INT PRIMARY KEY, price DECIMAL(8,2), imagepath VARCHAR(100),category INT NOT NULL,price_bought DECIMAL(8,2) NOT NULL,description TEXT NOT NULL,date_upl
            //TODO img path
            java.sql.Timestamp product_timestamp = new java.sql.Timestamp(this.getDateUploaded().getTime());
            //Check conditions before actually attempting to insert into database
            boolean shouldInsert = checkConditions();
            if (shouldInsert)
                st.executeUpdate("INSERT INTO product(id,imagepath, price, category, price_bought, description, date_upload,online_link,price_sold,product_condition,months_used,location,user_id, payment_method) VALUES ("
            else
                return false;
            //Confirm that product is, in fact, inserted into DB.
            ResultSet rs = st.executeQuery("SELECT * from product where id = "+id);
            if(!rs.next())
                returnVal = false;
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
        return returnVal;
    }

    public boolean updateProductInDatabase() throws ClassNotFoundException {
        return updateProductInDatabase(false);
    }

    public boolean updateProductInDatabase(boolean isTest) throws ClassNotFoundException {
        String myDriver = null;
        String myURL = null;
        Connection conn = null;
        boolean returnVal = true;
        if(isTest) {
            myDriver = "com.mysql.jdbc.Driver";
            myURL = "jdbc:mysql://localhost:3306/mydatabase?zeroDateTimeBehavior=convertToNull";
        }
        else{
            myDriver = "com.mysql.jdbc.Driver";
            myURL = "jdbc:mysql://lionmart.cvkcqiaoutkr.us-east-1.rds.amazonaws.com:3306/lionmart";
        }
        try {
```

```
305             Class.forName(myDriver);
306             if(isTest)
307             {
308                 conn = DriverManager.getConnection(myURL, "root", "");
309             }
310             else
311             {
312                 conn = DriverManager.getConnection(myURL, "lionadmin", "lionlynx42");
313             }
314             Statement st = conn.createStatement();
315             st.executeUpdate("CREATE TABLE IF NOT EXISTS product (id INT PRIMARY KEY, price DECIMAL(8,2), imagepath VARCHAR(100),category INT NOT NULL,price_bought DECIMAL(8,2) NOT NULL,description TEXT NOT NULL,date_upl
316
317             java.sql.Timestamp product_timestamp = new java.sql.Timestamp(this.getDateUploaded().getTime());
318             //Check conditions before actually attempting to update into database
319             boolean shouldInsert = checkConditions();
320
321             if (shouldInsert) {
322                 System.out.println("shouldinsert is True!");
323                 st.executeUpdate("UPDATE product SET price=" + this.getPrice() + ",category='" + this.getCategory() + "',price_bought=" + this.getPriceBought() + ",description='" + this.getDescription() + "',date_upload=
324             }
325             else
326             {
327                 System.out.println("product upload check conditions failed");
328                 return false;
329             }
330
331             //Confirm that product is, in fact, inserted into DB.
332             ResultSet rs = st.executeQuery("SELECT * from product where id = "+id);
333             if(!rs.next()){
334                 System.out.println("updated product not found!");
335                 returnVal = false;
336             }
337             if(isTest)
338                 st.executeUpdate("DROP TABLE product;");
339             conn.close();
340         } catch (SQLException e) {
341             e.printStackTrace();
342             return false;
343         }
344         return returnVal;
345     }
346 }
```

generated on 2016-12-10 22:40