

```

1 package application;
2
3 import java.sql.Connection;
13 /**
14  * expense - adds to existing catagories
15  *
16  * @author lokeshbudda
17  *
18  */
19 public class Expense extends Transaction implements Presistable {
20
21     private String expenseCatagory;
22     public static ArrayList<String> categoryOptions = new ArrayList<String> (
23         List.of("Food", "Transportation", "Entertainment", "Gym", "Shopping", "Other"));
24
25     public Expense() {
26         super();
27     }
28
29     public Expense(Date date, String title, Double amount, String type, String description, String userName,
30         Boolean pending) {
31         super(date, title, amount, type, description, userName, pending);
32     }
33
34     /**
35      * save(Abstract method) - saves all data to the correct categories.
36      */
37     @Override
38     public void save() {
39         Connection conn;
40         try {
41             conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/LoginDatabase", "root", "password");
42
43             PreparedStatement ps = conn.prepareStatement(
44                 "INSERT INTO `MYAPP`.`DETAILS` (title, description, subType) VALUES (?, ?, ?)",
45                 Statement.RETURN_GENERATED_KEYS);
46
47             ps.setString(1, this.getTitle());
48             ps.setString(2, this.getDescription());
49             ps.setString(3, this.getExpenseCatagory());
50             ps.executeUpdate();
51             ResultSet rs = ps.getGeneratedKeys();
52             if (rs.next()) {
53                 int detailId = (int) rs.getLong(1);
54                 PreparedStatement ps1 = conn.prepareStatement(
55                     "INSERT INTO `MYAPP`.`TRANSACTIONS` (transactionDate, fromUser, toUser, amount, detailId) VALUES (?, ?, ?, ?, ?)");
56
57                 ps1.setDate(1, this.getDate());
58                 ps1.setString(2, this.getUserName());
59                 ps1.setString(3, "NONE");
60                 ps1.setDouble(4, this.amount);
61                 ps1.setInt(5, detailId);
62                 ps1.executeUpdate();
63             }
64         }
65     }

```

```
66     } catch (SQLException e) {
67         // TODO Auto-generated catch block
68         e.printStackTrace();
69     }
70 }
71
72 // Abstract method implementation
73 @Override
74 public Expense read() {
75     return null;
76 }
77
78 // Abstract method implementation
79 @Override
80 public void showTransaction() {
81 }
82
83 // getter and setter
84 public String getExpenseCatagory() {
85     return expenseCatagory;
86 }
87
88 public void setExpenseCatagory(String expenseCatagory) {
89     this.expenseCatagory = expenseCatagory;
90 }
91
92 @Override
93 public ArrayList<String> getCategoryOptions() {
94     return this.categoryOptions;
95 }
96
97 }
98
99 }
100
```