SetUpMyUser.java

```
package ed.jdbc;

import java.util.ArrayList;
import java.util.Scanner;


public class SetUpMyUser {

   public static void main(String[] args) {
      MyDB mydb = new MyDB();

      System.out.println("Welcome to the database");
      while (true) {
         System.out.println("1. Check current record\n" +
                  "2. Create a record\n" +
                  "3. Update a user\n" +
                  "4. Delete a record\n" +
                  "5. Exit");
         Scanner option1 = new Scanner(System.in);
         int choice = Integer.parseInt(option1.nextLine());

         if (choice == 1) {
            Scanner option2 = new Scanner(System.in);
            System.out.println("Enter User ID");
            String userID = option2.nextLine();
            Myuser user = mydb.getRecord(userID);
            System.out.println("Name: " + user.getName() + "\n" +
                     "Password: " + user.getPassword() + "\n" +
                     "Email: " + user.getEmail() + "\n" +
                     "Phone: " + user.getPhone() + "\n" +
                     "Address: " + user.getAddress() + "\n" +
                     "SecQn: " + user.getSecQn() + "\n" +
                     "SecAns: " + user.getSecAns() + "\n"
                     );

         } else if (choice == 2) {
            Scanner s = new Scanner(System.in);

            System.out.println("Enter UserID");
            String userID = s.nextLine();

            System.out.println("Name: ");
            String name = s.nextLine();
```

```
            System.out.println("Password: ");
            String password = s.nextLine();

            System.out.println("Email: ");
            String email = s.nextLine();

            System.out.println("Phone: ");
            String phone = s.nextLine();

            System.out.println("Address: ");
            String address = s.nextLine();

            System.out.println("Secqn:");
            String secqn = s.nextLine();

            System.out.println("Secans:");
            String secans = s.nextLine();

            Myuser newUser = new Myuser(userID, name, password, email, phone, address,
secqn, secans);
            mydb.createRecord(newUser);
        } else if (choice == 3) {
            Scanner s = new Scanner(System.in);

            System.out.println("Enter UserID to Update");
            String userID = s.nextLine();

            System.out.println("Name: ");
            String name = s.nextLine();

            System.out.println("Password: ");
            String password = s.nextLine();

            System.out.println("Email: ");
            String email = s.nextLine();

            System.out.println("Phone: ");
            String phone = s.nextLine();

            System.out.println("Address: ");
            String address = s.nextLine();

            System.out.println("Secqn:");
            String secqn = s.nextLine();

            System.out.println("Secans:");
```

```
            String secans = s.nextLine();

            Myuser updatedUser = new Myuser(userID, name, password, email, phone,
address, secqn, secans);
            mydb.updateRecord(updatedUser);
         } else if (choice == 4) {

            Scanner s = new Scanner(System.in);
            System.out.println("Enter UserID to Delete");
            String userID = s.nextLine();
            mydb.deleteRecord(userID);

         } else if (choice == 5) {
            System.exit(1);
         } else {
            System.out.println("Invalid Selection");
         }
      }

   }

   public static ArrayList<Myuser> prepareMyuserData() { ArrayList<Myuser> myList = new
ArrayList<Myuser>();
      Myuser myuser1 = new Myuser("000001", "Peter Smith", "123456",
"psmith@swin.edu.au", "9876543210", "Swinburne EN510f", "What is my name?", "Peter");
      Myuser myuser2 = new Myuser("000002", "James T. Kirk", "234567",
"jkirk@swin.edu.au", "8765432109", "Swinburne EN511a", "What is my name?", "James");
      Myuser myuser3 = new Myuser("000003", "Sheldon Cooper", "345678",
"scooper@swin.edu.au", "7654321098", "Swinburne EN512a", "What is my last name?",
"Cooper");
      Myuser myuser4 = new Myuser("000004", "Clark Kent", "456789",
"ckent@swin.edu.au", "6543210987", "Swinburne EN513a", "What is my last name?",
"Kent");
      Myuser myuser5 = new Myuser("000005", "Harry Potter", "567890",
"hpotter@swin.edu.au", "6543210987", "Swinburne EN514a", "What is my last name?",
"Potter");
      myList.add(myuser1);
      myList.add(myuser2);
      myList.add(myuser3);
      myList.add(myuser4);
      myList.add(myuser5);
      return myList;
   }
}
```

MyDB.java

```
package ed.jdbc;

import java.io.IOException;
import java.sql.*;
import java.util.ArrayList;

public class MyDB {

   public MyDB() {

   }

   public void createMyuserTable() {
      Connection cnnct = null;
      Statement stmnt = null;
      try {
         cnnct = getConnection();
         stmnt = cnnct.createStatement();
         stmnt.execute("CREATE TABLE MYUSER ( "
               + " UserId CHAR(6) CONSTRAINT PK_CUSTOMER PRIMARY KEY, " + " Name
CHAR(30), Password CHAR(6), Email CHAR(30), " + " Phone CHAR(10), Address CHAR(60), "
               + " SecQn CHAR(60), SecAns CHAR(60))");
      } catch (SQLException ex) {
         while (ex != null) {
            ex.printStackTrace();
            ex = ex.getNextException();
         }
      } catch (IOException ex) {
         ex.printStackTrace();
      } finally {
         if (stmnt != null) {
            try {
               stmnt.close();
            } catch (SQLException e) {
            }
         }
         if (cnnct != null) {
            try {
               cnnct.close();
            } catch (SQLException sqlEx) {
            }
         }
      }
   }
```

```
public void dropMyuserTable() {
   Connection cnnct = null;
   Statement stmnt = null;
   try {
      cnnct = getConnection();
      stmnt = cnnct.createStatement();
      stmnt.execute("DROP TABLE MYUSER");
   } catch (SQLException ex) {
      while (ex != null) {
         ex.printStackTrace();
         ex = ex.getNextException();
      }
   } catch (IOException ex) {
      ex.printStackTrace();
   } finally {
      if (stmnt != null) {
         try {
            stmnt.close();
         } catch (SQLException e) {
         }
      }
      if (cnnct != null) {
         try {
            cnnct.close();
         } catch (SQLException sqlEx) {
         }
      }
   }
}



public static Connection getConnection() throws SQLException, IOException {
   System.setProperty("jdbc.drivers", "org.apache.derby.jdbc.ClientDriver");
   String url = "jdbc:derby://localhost/sun-appserv-samples;create=true";
   String username = "APP";
   String password = "APP";
   return DriverManager.getConnection(url, username, password);
}

public void addRecords(ArrayList<Myuser> myUsers) {
   Connection cnnct = null;
   PreparedStatement pStmnt = null;
   try {
      cnnct = getConnection();
      String preQueryStatement
```

```
            = "INSERT INTO MYUSER VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
        pStmnt = cnnct.prepareStatement(preQueryStatement);
        for
        (Myuser myuser : myUsers) {
            pStmnt.setString(1, myuser.getUserid());
            pStmnt.setString(2, myuser.getName());
            pStmnt.setString(3, myuser.getPassword());
            pStmnt.setString(4, myuser.getEmail());
            pStmnt.setString(5, myuser.getPhone());
            pStmnt.setString(6, myuser.getAddress());
            pStmnt.setString(7, myuser.getSecQn());
            pStmnt.setString(8, myuser.getSecAns());
            int rowCount = pStmnt.executeUpdate();
            if (rowCount == 0) {
                throw new SQLException("Cannot insert records!");
            }
        }
    } catch  (SQLException ex) {
        while (ex != null) {
            ex.printStackTrace();
            ex = ex.getNextException();
        }
    } catch (IOException ex) {
        ex.printStackTrace();
    } finally {
        if (pStmnt != null) {
            try {
                pStmnt.close();
            } catch (SQLException e) {

            }
        }
        if (cnnct != null) {
            try {
                cnnct.close();
            } catch (SQLException sqlEX) {

            }
        }
    }
}

public boolean createRecord(Myuser myuser)  {
    Connection cnnct = null;
    PreparedStatement pStmnt = null;
    try {
        cnnct = getConnection();
```

```
        String preQueryStatement
              = "INSERT INTO MYUSER VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
        pStmnt = cnnct.prepareStatement(preQueryStatement);
        pStmnt.setString(1, myuser.getUserid());
        pStmnt.setString(2, myuser.getName());
        pStmnt.setString(3, myuser.getPassword());
        pStmnt.setString(4, myuser.getEmail());
        pStmnt.setString(5, myuser.getPhone());
        pStmnt.setString(6, myuser.getAddress());
        pStmnt.setString(7, myuser.getSecQn());
        pStmnt.setString(8, myuser.getSecAns());
        int rowCount = pStmnt.executeUpdate();
        if (rowCount == 0) {
           throw new SQLException("Cannot insert records!");
        }
     } catch (SQLException ex) {
        while (ex != null) {
           ex.printStackTrace();
           ex = ex.getNextException();
        }
        return false;
     } catch (IOException ex) {
        ex.printStackTrace();
        return false;
     } finally  {
        System.out.println("User Created");
        return true;
     }
  }

  public Myuser getRecord(String userId) {
     Connection cnnct = null;
     PreparedStatement pStmnt = null;
     try {
        cnnct = getConnection();
        String preQueryStatement
              = "SELECT * FROM APP.MYUSER WHERE USERID=(?)";
        pStmnt = cnnct.prepareStatement(preQueryStatement);
        pStmnt.setString(1, userId);

        ResultSet resultSet = pStmnt.executeQuery();

        if (resultSet.next() == false) {
           System.out.println("Result set is empty");
           return null;
        } else {
           String userID = resultSet.getString("USERID");
```

```
            String name = resultSet.getString("NAME");
            String password = resultSet.getString("PASSWORD");
            String email = resultSet.getString("EMAIL");
            String phone = resultSet.getString("PHONE");
            String address = resultSet.getString("ADDRESS");
            String secqn = resultSet.getString("SECQN");
            String secans = resultSet.getString("SECANS");

            Myuser user = new Myuser(userID, name, password, email, phone, address, secqn,
secans);
            return user;
         }
      } catch (SQLException ex) {
         while (ex != null) {
            ex.printStackTrace();
            ex = ex.getNextException();
         }
      } catch (IOException ex) {
         ex.printStackTrace();
      }
      return null;
   }

   public boolean updateRecord(Myuser myuser) {
      Connection cnnct = null;
      PreparedStatement pStmnt = null;
      try {
         cnnct = getConnection();
         String preQueryStatement
               = "UPDATE MYUSER SET " +
               "NAME=(?)," +
               "PASSWORD=(?)," +
               "EMAIL=(?)," +
               "PHONE=(?)," +
               "ADDRESS=(?)," +
               "SECQN=(?)," +
               "SECANS=(?)" +
               "WHERE USERID=(?)";
         pStmnt = cnnct.prepareStatement(preQueryStatement);
         pStmnt.setString(1, myuser.getName());
         pStmnt.setString(2, myuser.getPassword());
         pStmnt.setString(3, myuser.getEmail());
         pStmnt.setString(4, myuser.getPhone());
         pStmnt.setString(5, myuser.getAddress());
         pStmnt.setString(6, myuser.getSecQn());
         pStmnt.setString(7, myuser.getSecAns());
         pStmnt.setString(8, myuser.getUserid());
```

```
            int rowCount = pStmnt.executeUpdate();
            if (rowCount == 0) {
               System.out.println("User id does not exist");
            }else {
               System.out.println("User Id:" + myuser.getUserid() + " has been updated");
               return true;
            }
         } catch (SQLException ex) {
            while (ex != null) {
               ex.printStackTrace();
               ex = ex.getNextException();
            }
         } catch (IOException ex) {
            ex.printStackTrace();
         }
         return false;
      }

      public boolean deleteRecord(String userId) {
         Connection cnnct = null;
         PreparedStatement pStmnt = null;

         try {
            cnnct = getConnection();
            String sql
                  = "DELETE FROM MYUSER WHERE USERID=(?)";

            pStmnt = cnnct.prepareStatement(sql);
            pStmnt.setString(1, userId);

            pStmnt.executeUpdate();
            System.out.println("Record deleted successfully");
            return true;
         } catch (SQLException ex) {
            while (ex != null) {
               ex.printStackTrace();
               ex = ex.getNextException();
            }
         } catch (IOException ex) {
            ex.printStackTrace();
         }
         return false;
      }
   }
```

```
run:
Welcome to the database
1. Check current record
2. Create a record
3. Update a user
4. Delete a record
5. Exit
1
Enter User ID
000001
Name: PeterSmith
Password: 123456
Email: psmith@swin.edu.au
Phone: 9876543210
Address: Swinburne EN510f
SecQn: What is my name?
SecAns: Peter

1. Check current record
2. Create a record
3. Update a user
4. Delete a record
5. Exit
2
Enter UserID
000006
Name:
Duy
Password:
111111
Email:
kevin@gmail.com
Phone:
0123456789
Address:
1 Melbourne ave
Secqn:
How r u
Secans:
im gud
User Created
```

```
User Created
1. Check current record
2. Create a record
3. Update a user
4. Delete a record
5. Exit
1
Enter User ID
000006
Name: Duy
Password: 111111
Email: kevin@gmail.com
Phone: 0123456789
Address: 1 Melbourne ave
SecQn: How r u
SecAns: im gud

1. Check current record
2. Create a record
3. Update a user
4. Delete a record
5. Exit
3
Enter UserID to Update
000006
Name:
KEVIN
Password:
111111
Email:
kevin@yahoo
Phone:
000000
Address:
Road of Mel
Secqn:
U good
Secans:
yes
User Id:000006 has been updated
1. Check current record
```

```
1. Check current record
2. Create a record
3. Update a user
4. Delete a record
5. Exit
1
Enter User ID
000006
Name: KEVIN
Password: 111111
Email: kevin@yahoo
Phone: 000000
Address: Road of Mel
SecQn: U good
SecAns: yes

1. Check current record
2. Create a record
3. Update a user
4. Delete a record
5. Exit
4
Enter UserID to Delete
000006
Record deleted successfully
1. Check current record
2. Create a record
3. Update a user
4. Delete a record
5. Exit
5
```

4.1 MyUser acts as a DTO as it encapsulates the data and transfer it between classes and modules of the application. MyUser only contains private fields for the data and constructors.

4.1.1 MyDB takes the role of a DAO because it is responsible for hiding implementation details about how your data is stored and how it is retrieved. Data Access Object means it is responsible for the operating system of the application, where all the actions, commands, implementations taking place.

4.2 If Machine A did so, it will change the corresponding value in the database server. Because it acts a DTO, the data will hence be transferred across the classes and modules of the application.

4.3 If Machine B did so, it will not change the corresponding value in the database server. Because it acts as a DAO, meaning it only consists the design of the system but not able to consist or modify data.