# Design and plans

1/- Database

- Create database and tables (accounts, category, products)
- Insert into data for testing

2/- Model and ADO

- 2.1/- DTO => Entity class (account, category, product)
- 2.2/- Interface: Accessible with common methods needed
- 2.3/- Data Access Object: => define DAO for all entities class implements Accessible (AccountDAO, CategoryDAO, ProductDAO)

Example

Entity class

```
  6  package model;
  7
  8  import java.sql.Date;
  9
 10  /** Account DTO class ...16 lines */
 26  public class Account {
 27      private String account;
 28      private String pass;
 29      private String lastName;
 30      private String firstName;
 31      private Date birthday;
 32      private boolean gender;
 33      private String phone;
 34      private boolean isUse;
 35      private int roleInSystem;
 36      /** Default constructor ...3 lines */
 39      public Account() {...4 lines }
 43      /** Constructor with attributes ...12 lines */
 55      public Account(String account, String pass, String lastName, String firstName,
 56                     Date birthday, boolean gender, String phone, boolean isUse,
 57                     int roleInSystem) {...11 lines }
 68      public String getAccount() {...3 lines }
 71      public void setAccount(String account) {...3 lines }
 74      public String getPass() {...3 lines }
 77      public void setPass(String pass) {...3 lines }
 80      public String getLastName() {...3 lines }
 83      public void setLastName(String lastName) {...3 lines }
 86      public String getFirstName() {...3 lines }
 89      public void setFirstName(String firstName) {...3 lines }
 92      public Date getBirthday() {...3 lines }
 95      public void setBirthday(Date birthday) {...3 lines }
 98      /** True: male | False: female ...4 lines */
102      public boolean isGender() {...3 lines }
105      public void setGender(boolean gender) {...3 lines }
108      public String getPhone() {...3 lines }
111      /** Only digits, begin with 03|05|07|08|09 ...4 lines */
115      public void setPhone(String phone) {...3 lines }
118      /** True: being used | False: is prevented ...4 lines */
122      public boolean isIsUse() {...3 lines }
125      public void setIsUse(boolean isUse) {...3 lines }
128      /** Role in system {1: admin - others: staff} ...4 lines */
132      public int getRoleInSystem() {...3 lines }
135      /** Role in system {1: admin - others: staff} ...4 lines */
139      public void setRoleInSystem(int roleInSystem) {...3 lines }
142  }
```

```java
package model;

/** Category DTO class ...10 lines */
public class Category {
    private int typeId;
    private String categoryName;
    private String memo;
    /** Default constructor ...3 lines */
    public Category() {...3 lines }
    /** Constructor with attributes ...6 lines */
    public Category(int typeId, String categoryName, String memo) {...5 lines }

    public int getTypeId() {...3 lines }

    public void setTypeId(int typeId) {...3 lines }

    public String getCategoryName() {...3 lines }

    public void setCategoryName(String categoryName) {...3 lines }

    public String getMemo() {...3 lines }

    public void setMemo(String memo) {...3 lines }
}
```

```java
package model;

import java.sql.Date;

/** Product DTO class ...17 lines */
public class Product {
    private String productId;
    private String productName;
    private String productImage;
    private String brief;
    private Date postedDate;
    private Category type;
    private Account account;
    private String unit;
    private int price;
    private int discount;
    /** Default constructor ...3 lines */
    public Product() {...4 lines }
    /** Constructor with attributes ...13 lines */
    public Product(String productId, String productName, String productImage,
                   String brief, Date postedDate, Category type, Account account,
                   String unit, int price, int discount) {...12 lines }
    public String getProductId() {...3 lines }
    public void setProductId(String productId) {...3 lines }
    public String getProductName() {...3 lines }
    public void setProductName(String productName) {...3 lines }
    public String getProductImage() {...3 lines }
    public void setProductImage(String productImage) {...3 lines }
    public String getBrief() {...3 lines }
    public void setBrief(String brief) {...3 lines }
    public Date getPostedDate() {...3 lines }
    public void setPostedDate(Date postedDate) {...3 lines }
    public Category getType() {...3 lines }
    public void setType(Category type) {...3 lines }
    public Account getAccount() {...3 lines }
    public void setAccount(Account account) {...3 lines }
    public String getUnit() {...3 lines }
    public void setUnit(String unit) {...3 lines }
    public int getPrice() {...3 lines }
    public void setPrice(int price) {...3 lines }
    public int getDiscount() {...3 lines }
    public void setDiscount(int discount) {...3 lines }
}
```

## 2.2/- Interface

```java
package model.dao;

import java.util.List;

/**
 * Common methods used to perform business actions for Data Access Object
 * (Using generic type for all of DAO classes)
 * @author Huy Nguyễn Mai
 */
public interface Accessible<T> {
    int insertRec(T obj);
    int updateRec(T obj);
    int deleteRec(T obj);
    T getObjectById(String id);
    List<T> listAll();
}
```

## 2.3/- DAO classes

```java
package model.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletContext;
import model.Account;
import utilities.ConnectDB;

/** accounts( ...14 lines */
public class AccountDAO implements Accessible<Account>{
    private ServletContext sc;
    private Connection con;
    /** Default constructor ...4 lines */
    public AccountDAO() throws ClassNotFoundException, SQLException{...4 lines }
    /** Constructor to update ServletConfig object ...4 lines */
    public AccountDAO(ServletContext sc)
                    throws ClassNotFoundException, SQLException{...4 lines }
    /**...7 lines */
    private Connection getConnect(ServletContext sc)
                            throws ClassNotFoundException, SQLException{...3 lines }
    /** Add a new account to database ...5 lines */
    @Override
    public int insertRec(Account o) {...27 lines }
    /** Update information of an account ...5 lines */
    @Override
    public int updateRec(Account o) {...29 lines }
    /** Remove an account from database ...5 lines */
    @Override
    public int deleteRec(Account o) {...19 lines }
    /** Get all account by Role in system ...5 lines */
    public List<Account> listByRole(int role){...35 lines }
    /** Get all accounts from database ...4 lines */
    @Override
    public List<Account> listAll() {...34 lines }
    /** Get an account object by Account name ...5 lines */
    @Override
    public Account getObjectById(String id) {...33 lines }
    /** Update the status of isUsed field of the account ...6 lines */
    public int updateIsUsed(String acc, boolean isUsed){...21 lines }
    /** Check the login information ...6 lines */
    public Account loginSuccess(String acc, String pass){...4 lines }
}
```

```
6   package model.dao;
7
8   import java.sql.Connection;
9   import java.sql.PreparedStatement;
10  import java.sql.ResultSet;
11  import java.sql.SQLException;
12  import java.sql.Statement;
13  import java.util.ArrayList;
14  import java.util.List;
15  import java.util.logging.Level;
16  import java.util.logging.Logger;
17  import javax.servlet.ServletContext;
18  import model.Category;
19  import utilities.ConnectDB;
20
21  /**...4 lines */
25  public class CategoryDAO implements Accessible<Category>{
        private ServletContext sc;
        private Connection con;
28      /** Default constructor ...3 lines */
        public CategoryDAO() throws SQLException, ClassNotFoundException{...4 lines }
35      /** Constructor to update ServletConfig object ...4 lines */
39      public CategoryDAO(ServletContext sc)
                        throws ClassNotFoundException, SQLException{...4 lines }
44      /** Return connection with default parameter or from web descriptor ...7 lines */
51      private Connection getConnect(ServletContext sc)
                                throws ClassNotFoundException, SQLException{...3 lines }
55      /** Add a new category to database ...5 lines */
60      @Override
        public int insertRec(Category o) {...23 lines }
84      /** Update information of a category ...5 lines */
89      @Override
        public int updateRec(Category o) {...23 lines }
113     /** Remove a category from database ...5 lines */
118     @Override
        public int deleteRec(Category o) {...19 lines }
138     /** Get all categories from database ...4 lines */
142     @Override
        public List<Category> listAll() {...26 lines }
169     /** Get a category object by Id ...5 lines */
174     @Override
        public Category getObjectById(String id) {...25 lines }
200 }
```

```java
package model.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletContext;
import model.Account;
import model.Category;
import model.Product;
import utilities.ConnectDB;

/** products( ...15 lines */
public class ProductDAO implements Accessible<Product>{
    private ServletContext sc;
    private Connection con;
    /** Default constructor ...3 lines */
    public ProductDAO() throws ClassNotFoundException, SQLException{...4 lines }
    /** Constructor to update ServletConfig object ...4 lines */
    public ProductDAO(ServletContext sc)
                    throws ClassNotFoundException, SQLException{...4 lines }
    /** Return connection with default parameter or from web descriptor ...7 lines */
    private Connection getConnect(ServletContext sc)
                                throws ClassNotFoundException, SQLException{...3 lines }
    /** Add a new product to database ...5 lines */
    @Override
    public int insertRec(Product o) {...30 lines }
    /** Update information of a product ...5 lines */
    @Override
    public int updateRec(Product o) {...30 lines }
    /** Remove product from database ...5 lines */
    @Override
    public int deleteRec(Product o) {...18 lines }
    /** Get all products by categoryId ...5 lines */
    public List<Product> listByCategory(int categoryId) {...43 lines }
    /** Get all products from database ...4 lines */
    @Override
    public List<Product> listAll() {...42 lines }
    /** Get a product object by Id ...5 lines */
    @Override
    public Product getObjectById(String id) {...41 lines }
}
```

## 2.4/- ConnectDB class and Context parameters to configure the connection string

- Web descriptor (web.xml)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="htt
    <servlet>
    <servlet-mapping>
    <context-param>
        <param-name>hostAddress</param-name>
        <param-value>localhost</param-value>
    </context-param>
    <context-param>
        <param-name>instance</param-name>
        <param-value>BODUA_GROUP\\BODUAGROUP</param-value>
    </context-param>
    <context-param>
        <param-name>dbName</param-name>
        <param-value>ProductIntro</param-value>
    </context-param>
    <context-param>
        <param-name>userName</param-name>
        <param-value>productDemo</param-value>
    </context-param>
    <context-param>
        <param-name>userPass</param-name>
        <param-value>123456</param-value>
    </context-param>
    <context-param>
        <param-name>dbPort</param-name>
        <param-value>1433</param-value>
    </context-param>
    <session-config>
</web-app>
```

ConnectDB with 2 constructor

```java
package utilities;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.servlet.ServletContext;

/**...4 lines */
public class ConnectDB {
    private String hostName;
    private String instance;
    private String port;
    private String dbName;
    private String user;
    private String pass;
    /** Default constructor ...3 lines */
    public ConnectDB() {
        this.hostName="127.0.0.1";                          //--- localhost
        this.instance="BODUA_GROUP\\BODUAGROUP";
        this.port="1433";
        this.dbName="Human";
        this.user="qlHuman";
        this.pass="111111";
    }
    /** Constructor to read some parameters from web ...4 lines */
    public ConnectDB(ServletContext sc) {
        this.hostName=sc.getInitParameter("hostAddress");
        this.instance=sc.getInitParameter("instance");
        this.dbName=sc.getInitParameter("dbName");
        this.port=sc.getInitParameter("dbPort");
        this.user=sc.getInitParameter("userName");
        this.pass=sc.getInitParameter("userPass");
    }
    /** Format string using parameters to connection to SQL Server database ...4 lines */
    public String getURLString(){
        String fm = "jdbc:sqlserver://%s\\%s:%s;databaseName=%s;user=%s;password=%s;";
        return String.format(fm, this.hostName, this.instance.trim(),
                                 this.port, this.dbName, this.user,this.pass );
    }
    /** Get a connection to database ...6 lines */
    public Connection getConnection() throws ClassNotFoundException, SQLException{
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        return DriverManager.getConnection(getURLString());
    }
}
```

3/- Pages structure

Public pages

- Index.html
- product portfolio
- product details
- login

Private pages

- Dashboard
- addAccount
- accounts
- updateAccount

- addCategory
- categories
- updatecategory
- addProduct
- products
- updateProduct