**FPT UNIVERSITY**

**FPT APTECH**

**- - ❖ - -**

**Mobile Phone**

**Management System**

**Project Document**

|  |  |  |  |
| --- | --- | --- | --- |
| **Faculty:** |  | LE HONG KY (Mr.) |  |
| **Batch No.:** |  | F2-13-1508T1 |  |
| **Group No.:** |  | 1 |  |
| **Students:** |  | Full name | Roll No. |
|  | 1 | Nguyen Le Minh Hoang (Leader) | Student921107 |
|  | 2 | Nguyen Minh Tri | Student919012 |
|  | 3 | Truong Tuan Son | Student919013 |

**TABLE OF CONTENTS**

REVIEW 1

1. PROBLEM DEFINITION 6

2. CUSTOMER REQUIREMENT SPECIFICATION 7

2.1. Admin - User 7

2.2. System functions 7

2.2.1. Login - logout 7

2.2.2. User (admin role) 7

2.2.3. User (non-admin role) 9

2.2.4. Customer 10

2.2.5. Customer Level 11

2.2.6. Supplier 12

2.2.7. Branch 13

2.2.8. Product 14

2.2.9. Inbound note 15

2.2.10. Order 17

2.2.11. Order details 18

2.2.12. Customer invoice 18

2.2.13. SalesOff 19

2.2.14. Employee 19

3. HARDWARE AND SOFTWARE REQUIREMENTS 20

3.1. Hardware requirements 21

3.2. Software requirements 21

4. ARCHITECTURE AND DESIGN OF THE PROJECT 22

5. FLOWCHARTS 23

5.1. Login to system 23

5.2. Add new item 24

5.3. Update item 25

5.4. Search item 26

5.5. Delete/ disable/ enable item: 27

6. BOUNDARIES OF THE PROJECT 28

7. TASK SHEET REVIEW 1 29

1. ENTITY RELATIONAL DIAGRAM 31

2. ENTITY - ATTRIBUTES 32

3. TABLE: 35

3.1. Users 35

3.2. Role 35

3.3. Functions 35

3.4. Permission 35

3.5. Customer 35

3.6. Supplier 36

3.7. Category 36

3.8. Product 36

3.9. Inbound 37

3.10. In\_Detail 37

3.11. Outbound 37

3.12. Out\_Detail 38

3.13. Order\_Main 38

3.14. Order\_Detail 38

3.15. Status 38

4. TABLE RELATIONSHIP 40

5. TASKSHEET REVIEW 2 41

1. INTERFACE DESIGN 43

1.1. Login 43

1.2. 14.2 Server connect 44

1.3. 14.3 Home 45

1.4. 14.4 User 46

1.4.1. 14.4.1 User Management 46

1.4.2. 14.4.2 Add new user 47

1.5. 14.5 Role (not supported yet) 48

1.6. 14.7 Permission 49

1.7. 14.8 Category 51

1.7.1. 14.8.1 Category Management 51

1.7.2. 14.8.2 Category Insert 53

1.7.3. 14.8.3 Category Update 54

1.8. 14.9 Product 56

1.8.1. 14.9.1 Product Management 56

1.8.2. 14.9.2 Product Insert 58

1.8.3. 14.9.3 Product Update 60

1.9. 14.10 Inbound Note 62

1.9.1. 14.10.1 Inbound Note Management 62

1.9.2. 14.10.2 Inbound note Insert 64

1.9.3. 14.10.3 Inbound note Update 66

1.10. 14.11 Outbound (not supported yet) 67

1.11. 14.12 Status (not supported yet) 68

1.12. 14.13 Order 68

1.12.1. 14.13.1 Order Management 68

1.12.2. 14.13.2 View Order Information 69

1.12.3. 14.13.3 Insert Order 70

1.12.4. 14.13.4 Update Order 72

1.12.5. 14.13.5 Add product to order 73

1.13. 14.14 Customer 76

1.13.1. 14.14.1 Customer Management 76

1.13.2. 14.14.2 View Customer Detail 77

1.13.3. 14.14.3 Insert Customer 77

1.13.4. 14.14.4 Update Customer 78

2. 15 Task Sheet Review 3 79

REVIEW 1

# PROBLEM DEFINITION

Nowadays, management is regard as the need of every corners of the globe, from enormous enterprises to small businesses and particularly in our case is mobile phone store.

Mobile Phone Management System is made to satisfy that need. This is a friendly interface that allow users to manage 5 main roles of a standard selling store: admin, manager, seller, technician, inventory employee as listed in details below:

All users have an ability to update their own password of account. For admin, he is able to manage all kind of utilities from delegating roles for other users to handling users (add, update or disable). For manager, he has an ability to manage categories, products, orders, order details, customers, suppliers, inbound note & outbound note, discount for events... For seller, he has his own account which calculate his salary and display role. And for technician, he will maintain all resources of the software to assure it is well-performed.

Our software also deal with inbound and outbound notes which all negotiate with supplier. So that the store can update for newer product as well as update for more quantity. Only then, salesman will be able to get the products and ready to pack and deliver them to the customers. Of course, after buying products from store, customers will receive order and their identity will be stored in our system with their phone numbers. Customers can be discounted some amount of money if they are regarded as VIP member which will achieve only by gathering scores from their previous pay-up.

In conclusion, with all above features, our system will help improving management quality and help leveraging the profits of mobile store as software user’s expection.

# CUSTOMER REQUIREMENT SPECIFICATION

## Admin - User

* Admin: can create, update & disable user and user’s roles, but does not have user’s permissions.
* User: does not have admin’s permission, but can manage categories, products, orders, order details, customers, suppliers, inbound note & outbound note, warranty, employee…

## System functions

### Login - logout

#### Login

|  |  |
| --- | --- |
| Description | Login |
| Input | * UserID * UserPassword |
| Process | * Check username and password from database. * Check user’s role to open exact working environment for user. |
| Output | * If login’s failed, display error. * If login’s successful, display relating screen for user. |

#### 

#### Logout

|  |  |
| --- | --- |
| Description | Logout |
| Input | * Click logout button |
| Process | * Close connection |
| Output | * Display login form |

#### 

### User (admin role)

#### Display user list

|  |  |
| --- | --- |
| Description | Display user list |
| Input | * Login as admin. |
| Process | * Get information of user from database |
| Output | * Display user list |

#### 

#### Add, update user

|  |  |
| --- | --- |
| Description | Add, update user |
| Input | * UserID (not editable in add/update) * UserPassword * EmpID * RoleID * UserEnabled |
| Process | * Update data in database when add, update |
| Output | * Display user list and update user information |

#### 

#### Disable/ enable user

|  |  |
| --- | --- |
| Description | Disable/ enable user |
| Input | * UserID |
| Process | * Update data in database: set status of this user to false (disabled), or true (enabled). |
| Output | * Disable/Enable successfully. |

#### 

#### Display role

|  |  |
| --- | --- |
| Description | Display role list |
| Input | * Click role button/section |
| Process | * Get information of role from database |
| Output | * Display role list |

#### 

#### Add, update role

|  |  |
| --- | --- |
| Description | Add, update role item |
| Input | * RoleID (not editable in update) * RoleName |
| Process | * Update data in database when add, update. * Root admin cannot be added or updated by anyone. And only root admin can add or update anyone. * Normal admin cannot add or update other admin. |
| Output | * Display new or edited role |

#### 

#### Delete role:

|  |  |
| --- | --- |
| Description | Delete role item |
| Input | * RoleID |
| Process | * Delete role(s) * Role can be deleted only if there is no user who owns that role. * Root admin cannot be deleted by anyone. And only root admin can delete anyone. * Normal admin cannot delete other admin. |
| Output | * Display deletion successfully or failed. |

#### 

### User (non-admin role)

#### Display user information

|  |  |
| --- | --- |
| Description | Display user information |
| Input | * Click user information button (UserID) |
| Process | * Get information of user from database |
| Output | * Display user information |

#### 

#### Update user password

|  |  |
| --- | --- |
| Description | Update user password |
| Input | * Click user update button |
| Process | * Update password of user to database |
| Output | * Display updating successfully. |

### Customer

#### Display customers

|  |  |
| --- | --- |
| Description | Display customer list |
| Input | * Click customer button/section |
| Process | * Get information of customer from database |
| Output | * Display customer list |

#### 

#### Add, update customer

|  |  |
| --- | --- |
| Description | Add, update customer |
| Input | * CusID (not editable when update) * CusName * CusAddress * CusPhone * CusLevelID * CusEnabled |
| Process | * Update data in database when add, update |
| Output | * Display new or edited customer |

#### 

#### Search customer

|  |  |
| --- | --- |
| Description | Search customer |
| Input | Either one or all of these:   * CusID * CusName * CusAddress * CusPhone * CusLevelID * CusEnabled |
| Process | * Query data from database to get matched customers |
| Output | * Display matched customer list |

#### 

#### Enable, disable customer

|  |  |
| --- | --- |
| Description | Enable/disable customer |
| Input | * CusID |
| Process | * A customer can only be disabled if there are no order linked to that customer * Delete customer data in database |
| Output | * Disable/Enable successfully. |

#### 

### Customer Level

#### Display customer level

|  |  |
| --- | --- |
| Description | Display customer level list |
| Input | * Click customer level button/section |
| Process | * Get information of customer level from database |
| Output | * Display customer level list |

#### Add, update customer level

|  |  |
| --- | --- |
| Description | Add, update customer |
| Input | * CusLevelID (not editable when update) * CusLevel * CusLevelName * CusDiscount |
| Process | * Update data in database when add, update |
| Output | * Display new or edited customer |

#### Delete customer level

|  |  |
| --- | --- |
| Description | Delete customer level |
| Input | * CusLevelID |
| Process | * A customer level can be deleted only if there is no customer who owns that customer level ID. Normal customer level cannot be deleted. * Delete customer level data in database |
| Output | * Display deleting successfully or failed. |

### Supplier

#### Display supplier list

|  |  |
| --- | --- |
| Description | Display supplier list |
| Input | * Click supplier button/section |
| Process | * Get information of supplier from database |
| Output | * Display supplier list |

#### 

#### Add, update supplier

|  |  |
| --- | --- |
| Description | Add, update supplier details |
| Input | * SupID (not editable when update) * SupName * SupAddress * SupPhone * SupEnabled |
| Process | * Update data in database when add, update |
| Output | * Display new or edited supplier |

#### 

#### Search supplier

|  |  |
| --- | --- |
| Description | Search supplier |
| Input | Either one or all of these:   * SupID * SupName * SupAddress * SupPhone * SupEnabled |
| Process | * Query data from database to get matched supplier |
| Output | * Display matched supplier list |

#### 

#### Enable, disable supplier

|  |  |
| --- | --- |
| Description | Disable/enable supplier |
| Input | * Supplier ID |
| Process | * Disable/enable supplier data in database |
| Output | * Disable/Enable successfully. |

#### 

### Branch

#### Display branch

|  |  |
| --- | --- |
| Description | Display branch list |
| Input | * Click branch list button/section |
| Process | * Get information of branch from database |
| Output | * Display branch list |

#### 

#### Search branch

|  |  |
| --- | --- |
| Description | Search branch |
| Input | Either one or all of these:   * BraID * BraName * BraEnabled |
| Process | * Query data from database to get matched branches |
| Output | * Display branch list |

#### 

#### Add, update branch

|  |  |
| --- | --- |
| Description | Add, update branch details |
| Input | * BraID (not editable) * BraName * BraEnabled |
| Process | * Update data in database when add, update |
| Output | * Display new or edited branch details |

#### 

#### Enable, disable branch

|  |  |
| --- | --- |
| Description | Enable/disable branch |
| Input | * BraID |
| Process | * Update data in database when disable/enable |
| Output | * Display successful disable/enable message |

#### 

### Product

#### Display product list

|  |  |
| --- | --- |
| Description | Display product list |
| Input | * Click product list button/section |
| Process | * Get information of product list from database |
| Output | * Display product list |

#### 

#### Search product

|  |  |
| --- | --- |
| Description | Search product |
| Input | Either one or all of these:   * ProID * BraName * ProName * ProPrice * ProStock * ProDescr * ProEnabled |
| Process | * Query data from database to get matched products |
| Output | * Display matched product list |

#### 

#### Add, update product details

|  |  |
| --- | --- |
| Description | Add, update product details |
| Input | * ProID (not editable in update) * BraID * ProName * ProPrice * ProDescr * ProEnabled * ProImage |
| Process | * Update data in database when add, update |
| Output | * Display new or edited product details |

#### 

#### Enable, disable product

|  |  |
| --- | --- |
| Description | Enable/disable product |
| Input | * ProID |
| Process | * A product can be disabled only if its stock is zero * Update data in database when disable/re-enable |
| Output | * Display successful disable/re-enable message |

### Inbound note

#### Display inbound note list

|  |  |
| --- | --- |
| Description | Display inbound note list |
| Input | * Click inbound note list button/section |
| Process | * Get information of inbound note list from database |
| Output | * Display inbound note list |

#### 

#### Display inbound note detail

|  |  |
| --- | --- |
| Description | Display inbound note detail |
| Input | * InID |
| Process | * Get information of inbound note from database |
| Output | * Display inbound note detail |

#### 

#### Search inbound note list

|  |  |
| --- | --- |
| Description | Search inbound note |
| Input | Either one or all of these:   * InID * InDate * SupID * SupInvoiceID * UserID * InEnabled |
| Process | * Query data from database to get matched inbound notes |
| Output | * Display matched inbound notes |

#### 

#### Search inbound note details

|  |  |
| --- | --- |
| Description | Search inbound note details |
| Input | Either one or all of these:   * InID * ProName * ProQty * ProCost * SupName * InvoiceID |
| Process | * Query data from database to get matched inbound note details |
| Output | * Display matched inbound note details |

#### 

#### Add, update inbound note details

|  |  |
| --- | --- |
| Description | Add, update inbound note |
| Input | * InID (not editable in update) * InDate (not editable in update) * SupID (not editable in update) * InvoiceID (not editable in update) * UserID (not editable in update) * ProID * ProQty * ProCost |
| Process | * Update data in database when add, update note & details |
| Output | * Display new or edited inbound note detail |

#### 

### Order

#### Display order list

|  |  |
| --- | --- |
| Description | Display order list |
| Input | * Click order button/section |
| Process | * Get information of order list from database |
| Output | * Display order list |

#### 

#### Add, update order list

|  |  |
| --- | --- |
| Description | Add, update order list |
| Input | * OrdID (not editable in update) * CusID * UserID * OrdDate * StatusID |
| Process | * Update data in database when add, update order list |
| Output | * Display new or edited order |

#### 

#### Search order list

|  |  |
| --- | --- |
| Description | Search order list |
| Input | Either one or all of these:   * OrdID * CusName * UserName * OrdDate * StatusName |
| Process | * Query data from database to get information of orders which match search require |
| Output | * If found, display information matched orders * If not found, display notification not found |

#### 

### Order details

#### Add, update order details

|  |  |
| --- | --- |
| Description | Add, update order detail |
| Input | * OrdID (not editable in update) * ProID * OrdQty * OrdProPrice * OrdCusDiscount |
| Process | * Update data in database when add, update order details |
| Output | * Display new or edited order detail |

#### 

### Customer invoice and print invoice

#### Display invoice

|  |  |
| --- | --- |
| Description | Display invoice list |
| Input | * Click invoice list button/section |
| Process | * Get information of invoice list from database |
| Output | * Display invoice list |

#### 

#### Search invoice

|  |  |
| --- | --- |
| Description | Search invoice of customer |
| Input | Either one or all of these:   * InvoiceID * InvoiceDate * OrdID * CusName |
| Process | * Query data from database to get matched invoice |
| Output | * Display matched invoice |

#### 

#### Print invoice

|  |  |
| --- | --- |
| Description | Print invoice for customer |
| Input | * Click print invoice button/section |
| Process | * Get information of customer invoice from database to print invoice |
| Output | * Printer or file |

### SalesOff

#### Display salesoff

|  |  |
| --- | --- |
| Description | Display salesoff list |
| Input | * Click salesoff button/section |
| Process | * Get information of salesoff list from database |
| Output | * Display salesoff list |

#### Add, update salesoff

|  |  |
| --- | --- |
| Description | Add, update order detail |
| Input | * SalesOffID (not editable in update) * SalesOffName * ProID * SalesOffStartDate * SalesOffEndDate * SalesOffAmount |
| Process | * Update data in database when add, update salesoff details |
| Output | * Display new or edited salesoff details |

#### Delete salesoff

|  |  |
| --- | --- |
| Description | Delete salesoff |
| Input | * SalesOffID |
| Process | * Update data in database when deleting. |
| Output | * Display deleting successfully |

### Employee

#### Display employee

|  |  |
| --- | --- |
| Description | Display employee list |
| Input | * Click employee button/section |
| Process | * Get information of employee from database |
| Output | * Display employee list |

#### Add, update employee

|  |  |
| --- | --- |
| Description | Add, update employee |
| Input | * EmpID (not editable in update) * Birthday * BasicSalary * Designation * DesBonus * WorkingStartDate * EmpEnabled |
| Process | * Update data in database when add, update employee management |
| Output | * Display new or edited employee management |

#### Enable, disable employee

|  |  |
| --- | --- |
| Description | Enable/disable employee |
| Input | * EmpID |
| Process | * Update data in database when disable/enable |
| Output | * Display successful disable/enable message |

### Customer feedback

#### Display customer feedback list

|  |  |
| --- | --- |
| Description | Display customer feedback list |
| Input | * Click customer care button/section |
| Process | * Get information of employee from database |
| Output | * Display employee list |

#### Add, update customer feedback

|  |  |
| --- | --- |
| Description | Add, update customer feedback |
| Input | * FBID (not editable in update) * CusID (not editable in update) * FBContent * FBDate * FBReply * FBReplyDate |
| Process | * Update data in database when add, update feedback |
| Output | * Display new or edited feedback |

#### Delete customer feedback

|  |  |
| --- | --- |
| Description | Delete customer feedback |
| Input | * FBID |
| Process | * Update data in database when deleting |
| Output | * Display successful deleting message |

### Statistics and print report

#### Display summary statistics

|  |  |
| --- | --- |
| Description | Display summary statistics about sales activities of store |
| Input | * Click statistics button/section |
| Process | * Get information of sale total amount, cost total amount, profits of store from database |
| Output | * Display summary statistics |

#### Print report

|  |  |
| --- | --- |
| Description | Print report for summary statistics |
| Input | * Click print report button/section |
| Process | * Get information of sale total amount, cost total amount, profits of store from database to print report |
| Output | * Printer or file |

### Service management

#### Display service list

|  |  |
| --- | --- |
| Description | Display service list |
| Input | * Click service button/section |
| Process | * Get information of employee from database |
| Output | * Display employee list |

#### Add, update service

|  |  |
| --- | --- |
| Description | Add, update service item |
| Input | * ServiceID (not editable in update) * UserID (not editable in update) * CusID * ServiceContent * ReceiveDate * ReturnDate * ServiceStatusID |
| Process | * Update data in database when add, update service item |
| Output | * Display new or edited service item |

#### Delete service

|  |  |
| --- | --- |
| Description | Delete service item |
| Input | * ServiceID |
| Process | * Update data in database when deleting |
| Output | * Display deleting successfully |

# HARDWARE AND SOFTWARE REQUIREMENTS

## Hardware requirements

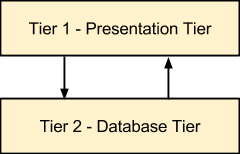
|  |  |  |
| --- | --- | --- |
| **Requirements** | **User** | **Developer** |
| Processor | 32-bit: 1.0 GHz  64-bit: 1.4 GHz | 2.0 GHz or faster |
| Memory (RAM) | 2.0 GB or more | 4.0 GB or more |
| Disk space | 2.0 GB or more | |

## Software requirements

|  |  |  |
| --- | --- | --- |
| **Requirements** | **User** | **Developer** |
| Operating System | Windows 7 or later | Windows 7 or later |
| SQL Server | 2012 or later | 2012 or later |
| JRE | 1.8.0-20 or later | 1.8.0-20 or later |
| Netbeans |  | Netbean 8.1 |
| JDK |  | 8u25 |

# ARCHITECTURE AND DESIGN OF THE PROJECT

The application will have a two-tier architecture.



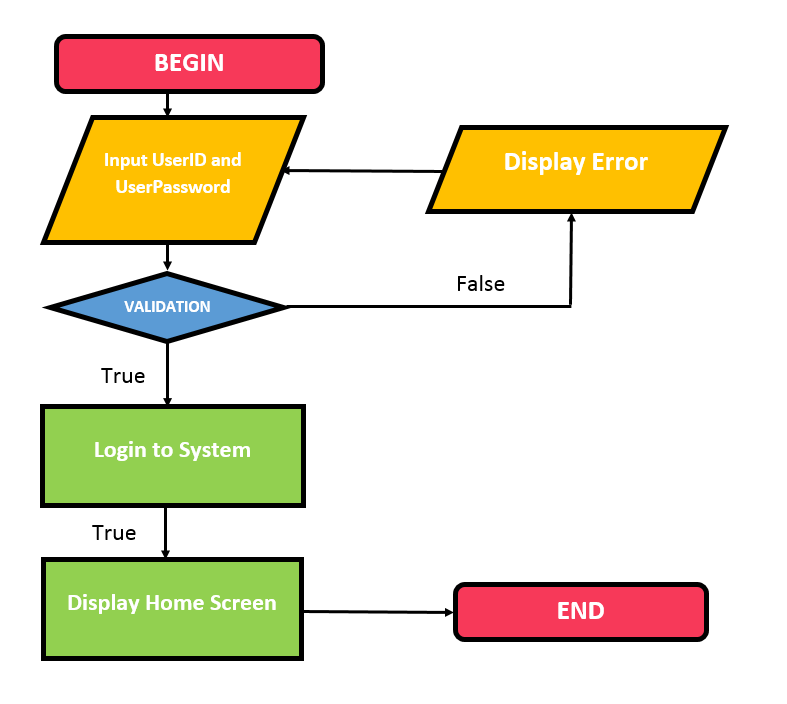
Two-tier architecture of the Project

**Management GUI (Tier 1):** This is a user interface that contains the frame enabled the form, each frame enabled the form is an interface. Layout is built depending on the requirements of the applications have separate functions.

**Data Access (Tier 2):** As a class, including the method of installation process connected database, to define the method of connection (connected by any type), called the method of connecting functions and the method of Static models for use in the classes in different packages.

# FLOWCHARTS

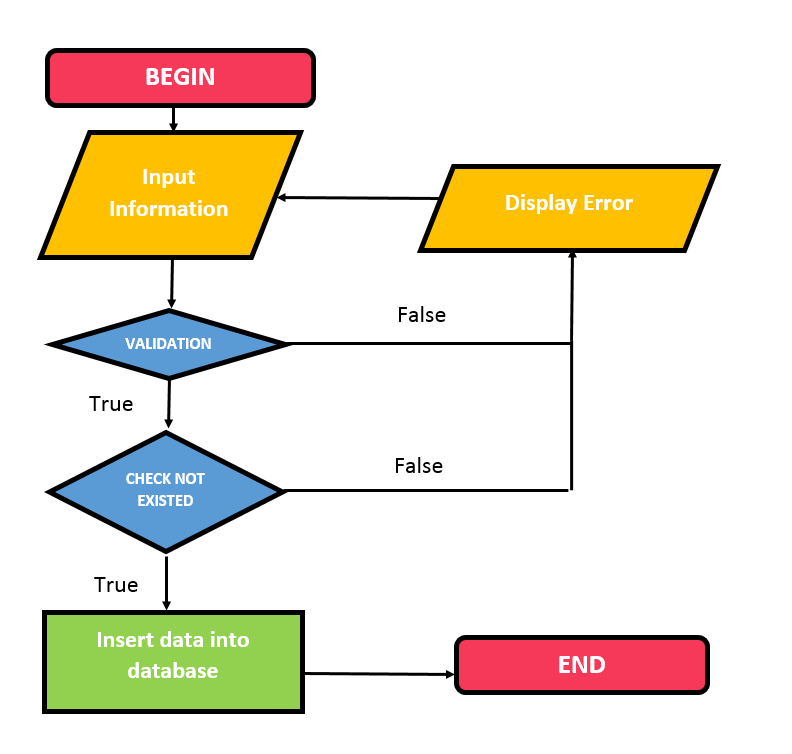
## Login to system



**Description:**

* Input username and password.
* If data input and in database is match, program will be show main program, otherwise failed a message error will be display.

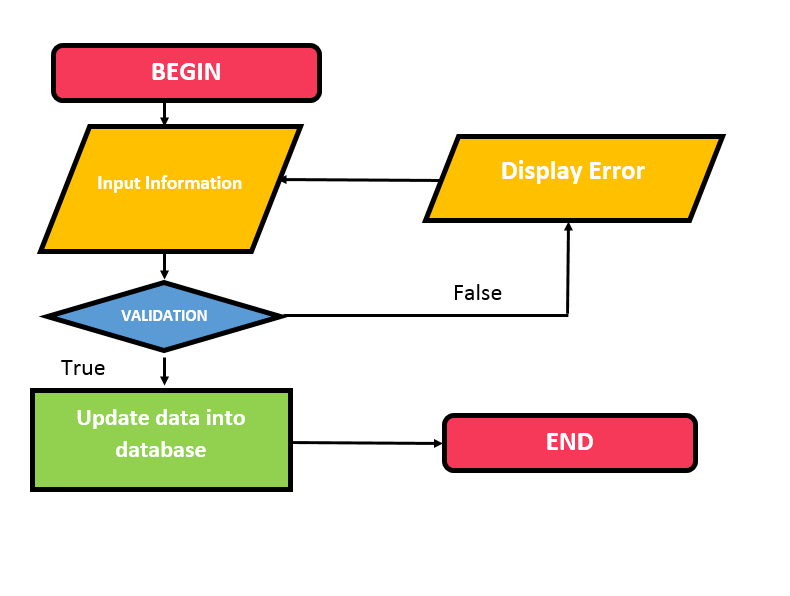
## Add new item



**Description:**

* Using the Add function in the system management.
* Input data field system requirements.
* If validating is success and new object does not existed, data will be add to database, otherwise failed a message error will be display.

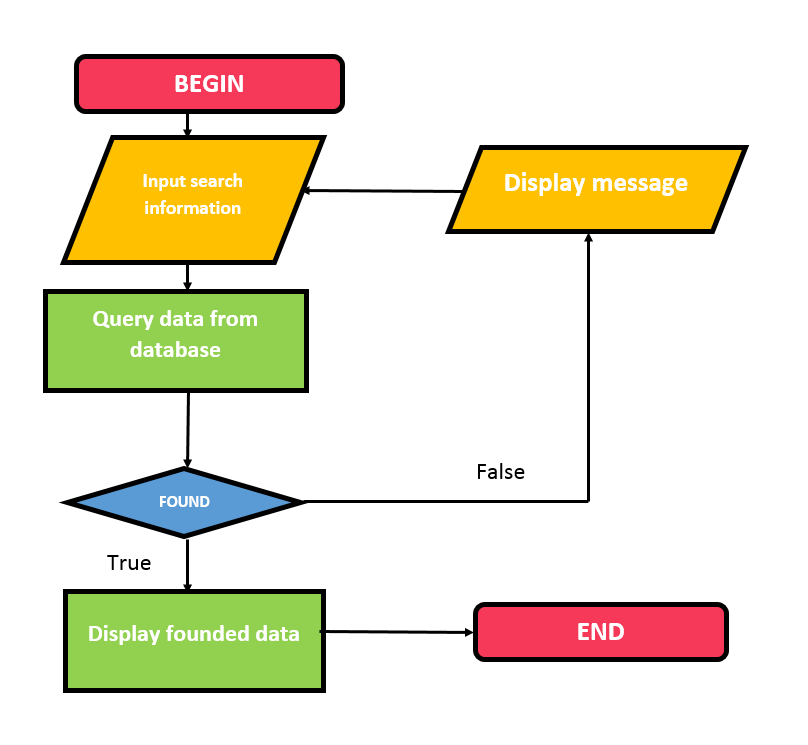
## Update item



**Description:**

* Using the Update function in the system management.
* Data entry needs update.
* If validating is success, data will be update to database, otherwise failed a message error will be display.

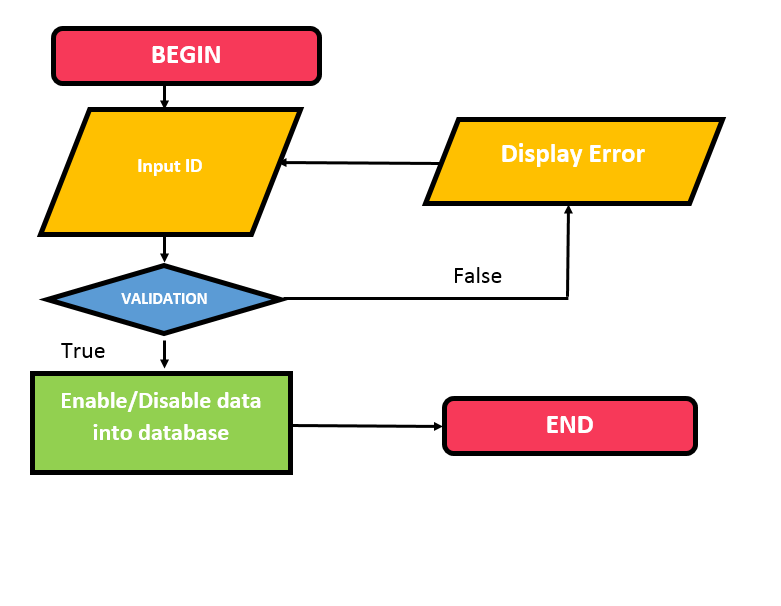
## Search item



**Description:**

* Input information to search with valid info.
* Program will query database. If not found, display message not found, otherwise display founded items.

## Delete/ disable/ enable item:



**Description:**

* Using the delete/ disable/ enable function in the system management.
* Input ID needs delete/ disable/ enable.

If ID invalid and there is no conflict (linked to other tables...), item will be delete/ set disabled/ set enabled from database, otherwise failed a message error will be displayed.

# BOUNDARIES OF THE PROJECT

* Not support products returning process from customer.
* Outbound note must link from an order, not support direct outbound note yet.

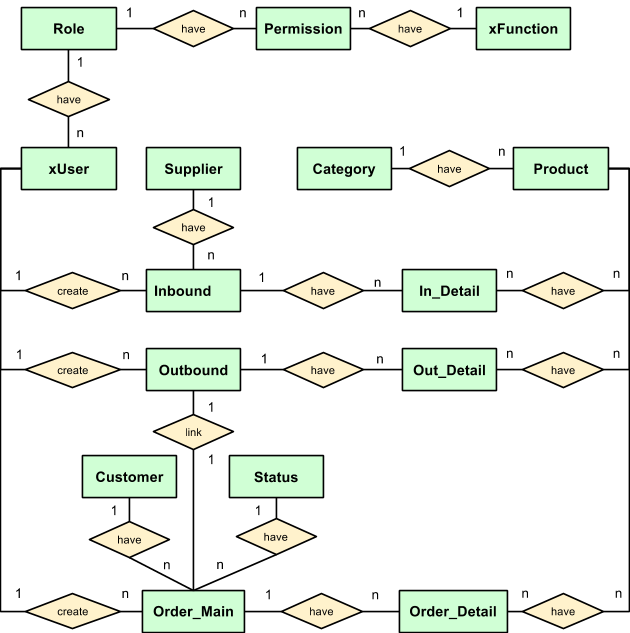
# TASK SHEET REVIEW 1

*\* CRS: Customer Requirement Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **Group members** | **Description of works** | | |
| **Work** | | **Status** |
| Nguyen Le Minh Hoang  (Student921107) | CRS – Customer,  CRS – Customer level,  CRS – SalesOff,  CRS – Order,  CRS – Order details,  CRS – Invoice. | | Completed |
| Architecture and Design of the Project | | Completed |
| Hardware - Software requirements | | Completed |
| Boundaries of the project | | Completed |
| Nguyen Minh Tri  (Student919012) | CRS – User,  CRS – Role,  CRS – Function,  CRS – Permission,  CRS – Employee. | | Completed |
| Truong Tuan Son  (Student919013) | Problem Definition | | Completed |
| CRS – Supplier,  CRS – Branch,  CRS – Product,  CRS – Inbound note. | | Completed |
| **Team Leader Signature** | | **Approved by Faculty** | |
|  | |  | |
| Mr. Nguyen Le Minh Hoang | | Mr. Le Hong Ky | |

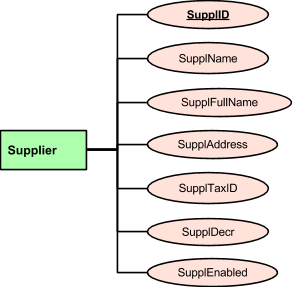
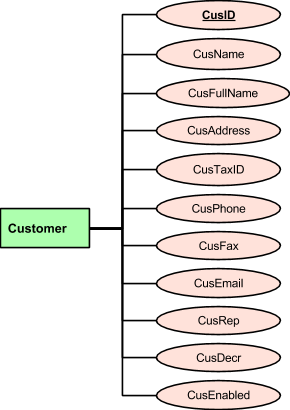
REVIEW 2

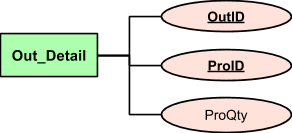
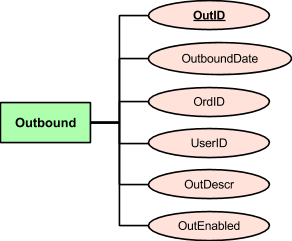
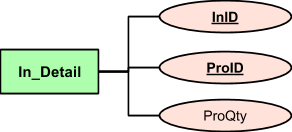
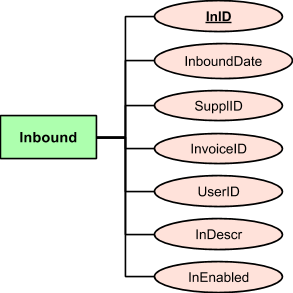
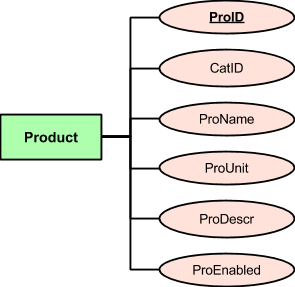
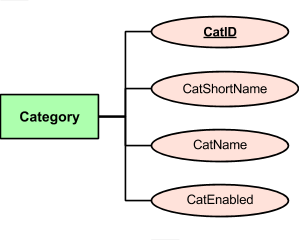
# ENTITY RELATIONAL DIAGRAM

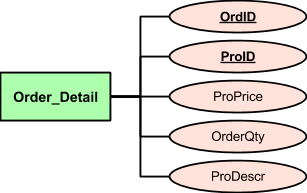
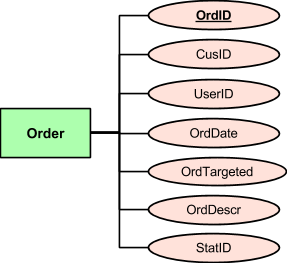


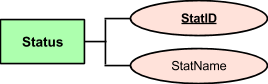
# ENTITY - ATTRIBUTES

|  |  |
| --- | --- |
|  |  |









# TABLE:

## Users

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| UserID | smallint | yes | no | PK |  |  | User's ID |
| Username | varchar(10) |  | no |  |  |  | Username to login |
| UserFirstName | nvarchar(10) |  | no |  |  |  | First name |
| UserFamilyName | nvarchar(25) |  | no |  |  |  | Family name |
| UserPassword | nvarchar(30) |  | no |  |  |  | Password |
| UserEmail | nvarchar(254) |  | yes |  |  |  | Email |
| RoleID | smallint |  | no |  | FK | Role (RoleID) | Role's ID |
| UserEnabled | bit |  | no |  |  |  | User's status: enable (1), disable (0) |

## Role

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| RoleID | smallint | yes | no | PK |  |  | Role's ID |
| RoleName | nvarchar(40) |  | no |  |  |  | Role's Name |

## Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| FunctionID | nvarchar(10) |  | no | PK |  |  | Function’s ID |
| FunctionGroup | nvarchar(40) |  | no |  |  |  | Function’s group |
| FunctionName | nvarchar(40) |  | no |  |  |  | Function’s name |

## Permission

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| RoleID | smallint |  | no | PK | FK | Role (RoleID) | Role's ID |
| FunctionID | nvarchar(10) |  | no | PK | FK | xFunction (FunctionID) | Function’s ID |

## Customer

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| CusID | smallint | yes | no | PK |  |  | Customer's ID |
| CusName | nvarchar(25) |  | no |  |  |  | Short name |
| CusFullName | nvarchar(254) |  | no |  |  |  | Full name |
| CusAddress | nvarchar(254) |  | no |  |  |  | Address |
| CusTaxID | varchar(20) |  | no |  |  |  | Tax ID |
| CusPhone | varchar(15) |  | yes |  |  |  | Phone |
| CusFax | varchar(15) |  | yes |  |  |  | Fax |
| CusEmail | nvarchar(254) |  | yes |  |  |  | Email |
| CusRep | nvarchar(40) |  | yes |  |  |  | Representative |
| CusDecr | nvarchar(254) |  | yes |  |  |  | Description |
| CusEnabled | bit |  | no |  |  |  | Cus's status: enable (1), disable (0) |

## Supplier

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| SupplID | smallint | yes | no | PK |  |  | Supplier's ID |
| SupplName | nvarchar(25) |  | no |  |  |  | Short name |
| SupplFullName | nvarchar(254) |  | no |  |  |  | Full name |
| SupplAddress | nvarchar(254) |  | no |  |  |  | Address |
| SupplTaxID | varchar(20) |  | no |  |  |  | Tax ID |
| SupplDecr | nvarchar(254) |  | yes |  |  |  | Description |
| SupplEnabled | bit |  | no |  |  |  | Supplier's status: enable (1), disable (0) |

## Category

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| CatID | smallint | yes | no | PK |  |  | Category's ID |
| CatShortName | nvarchar(12) |  | no |  |  |  | Short name |
| CatName | nvarchar(50) |  | no |  |  |  | Full name |

## Product

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| ProID | smallint | yes | no | PK |  |  | Product's ID |
| CatID | smallint |  | no |  | FK | Category (CatID) | Category's ID |
| ProName | nvarchar(70) |  | no |  |  |  | Product's name |
| ProUnit | nvarchar(10) |  | no |  |  |  | Product's unit |
| ProQtyStock | float |  | no |  |  |  | Product's stock quantity |
| ProPrice | float |  | no |  |  |  | Product's sell price |
| ProDescr | nvarchar(100) |  | no |  |  |  | Product's description |
| ProEnabled | bit |  | no |  |  |  | Product's status: enable (1), disable (0) |

## Inbound

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| InID | smallint | yes | no | PK |  |  | Inbound note's ID |
| InboundDate | datetime |  | no |  |  |  | Inbound date |
| SupplID | smallint |  | yes |  | FK | Supplier (SupplID) | Supplier's ID |
| InvoiceID | nvarchar(10) |  | yes |  |  |  | Purchasing invoice's ID |
| UserID | smallint |  | no |  | FK | Users (UserID) | User's ID |
| InDescr | nvarchar(254) |  | no |  |  |  | Inbound note's description |
| InEnabled | bit |  | no |  |  |  | Inbound note's status: enable (1), disable (0) |

## In\_Detail

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| InID | smallint |  | no | PK | FK | InBound(InID) | Inbound note's ID |
| ProID | smallint |  | no | PK | FK | Product (ProID) | Product's ID |
| ProQty | float |  | no |  |  |  | Product's inbound quantity |

## Outbound

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| OutID | smallint | yes | no | PK |  |  | Outbound note's ID |
| OutboundDate | datetime |  | no |  |  |  | Outbound date |
| OrdID | smallint |  | no |  | FK | Order\_Main (OrdID) | Order's ID |
| UserID | smallint |  | no |  | FK | Users (UserID) | User's ID |
| OutDescr | nvarchar(254) |  | no |  |  |  | Outbound note's description |
| OutEnabled | bit |  | no |  |  |  | Outbound note's status: enable (1), disable (0) |

## Out\_Detail

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| OutID | smallint |  | no | PK | FK | Outbound (OutID) | Outbound note's ID |
| ProID | smallint |  | no | PK | FK | Product (ProID) | Product's ID |
| ProQty | float |  | no |  |  |  | Product's outbound quantity |
| ProPrice | float |  | no |  |  |  | Product's sell price |

## Order\_Main

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| OrdID | smallint | yes | no | PK |  |  | Order's ID |
| CusID | smallint |  | no |  | FK | Customer (CusID) | Customer's ID |
| UserID | smallint |  | no |  | FK | Users (UserID) | User's ID |
| OrdDate | date |  | no |  |  |  | Order date |
| OrdTargeted | date |  | yes |  |  |  | Target date |
| OrdDescr | nvarchar(254) |  | yes |  |  |  | Order's description |
| StatID | int |  | no |  | FK | Status (StatID) | Status's ID |

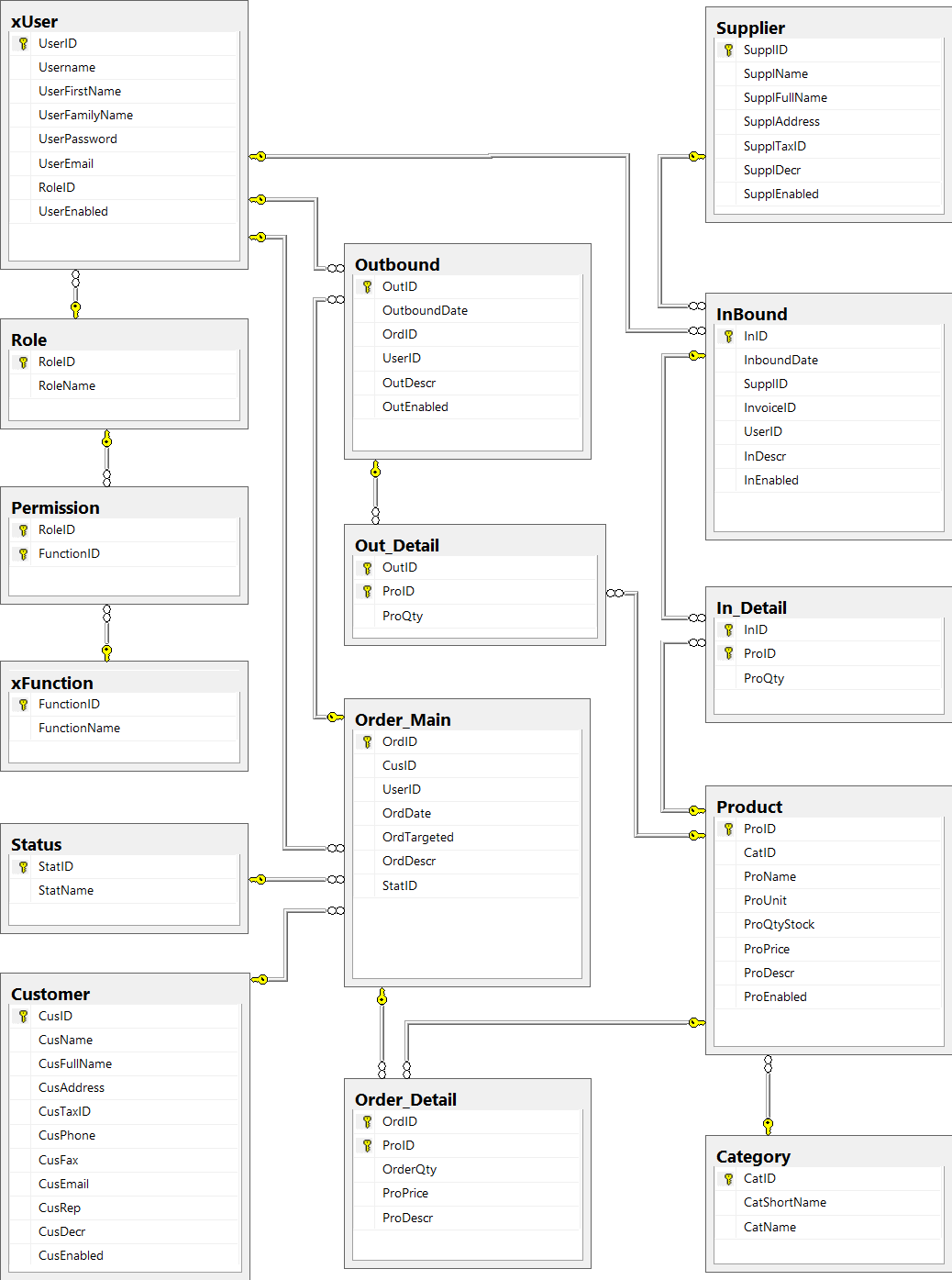
## Order\_Detail

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| OrdID | smallint |  | no | PK | FK | Order\_Main (OrdID) | Order's ID |
| ProID | smallint |  | no | PK | FK | Product(ProID) | Product's ID |
| OrderQty | float |  | no |  |  |  | Ordered quantity |
| ProPrice | float |  | no |  |  |  | Product's sell price |
| ProDescr | nvarchar(254) |  | yes |  |  |  | Product's description |

## Status

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Field name | Type (size) | Identity | NULL | PK | FK | References | Description |
| StatID | int |  | no | PK |  |  | Status's ID |
| StatName | nvarchar(15) |  | no |  |  |  | Status's name |

# TABLE RELATIONSHIP



# TASKSHEET REVIEW 2

|  |  |  |  |
| --- | --- | --- | --- |
| **Group members** | **Description of works** | | |
| **Work** | | **Status** |
| Le Van Thang  (student806165) | Entity Relational Diagram | | Completed |
| ED – Category, Product, Inbound, In\_Detail, Outbound, Out\_Detail | | Completed |
| Table relationship | | Completed |
| Le Minh Dong  (student806079) | ED – Users,  ED – Role,  ED – Functions,  ED – Permission, | | Completed |
| Nguyen Thi My Xuyen  (student803089) | ED – Customer,  ED – Supplier,  ED – Order,  ED – Status of Order | | Completed |
| **Team Leader Signature** | | **Approved by Faculty** | |
|  | |  | |
| Mr. Le Van Thang | | Mr. Le Hong Ky | |

REVIEW 3

# INTERFACE DESIGN

## Login

## pro3.png

**Description:** this is login form.

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | txtusername | JTextField | Input username |
| 2 | txtpassword | JPasswordField | Input password |
| 3 | btnLogin | JButton | Check validate and login by user |
| 4 | btnCancel | JButton | Cancel function and exit form |
| 5 | jmnConfig | JMenu | View connect server form |

**List of validation:**

|  |  |
| --- | --- |
| **Object** | **Validation** |
| txtusername | not null, is string |
| txtpassword | not null, is string |

**List of events**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnLoginActionPerformed | Check validate and login by user |
| btnCancelActionPerformed | Cancel function and exit form |
| jmnConfigMouseClicked | View connect server form |

### 

## 14.2 Server connect



**Description:** this is form connect server.

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | txtserver | JTextField | Input Server name |
| 2 | txtport | JTextField | Input port (1433) |
| 3 | txtdatabase | JTextField | Input database name |
| 4 | txtusername | JTextField | Input username |
| 5 | txtpassword | JTextField | Input password |
| 6 | btnConnect | JButton | Connect server |
| 7 | btnExit | JButton | Exit form |

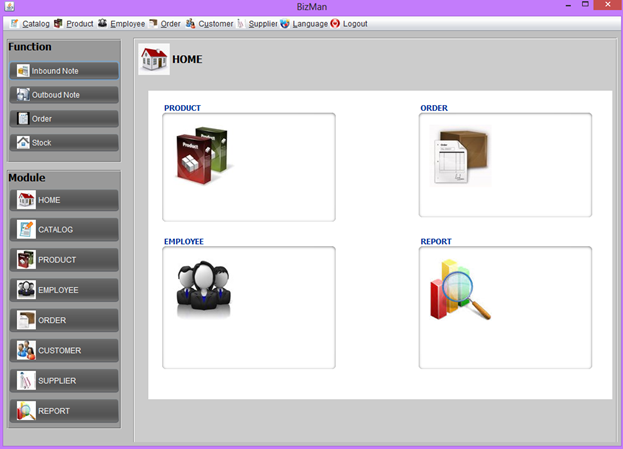
**List of validation:**

|  |  |
| --- | --- |
| **Object** | **Validation** |
| txtserver | not null, is string |
| txtport | not null, is string |
| txtdatabase | get database name, not null, is string |
| txtusername | Input username permission login, not null, is string, |
| txtpassword | Input password by username, not null, is string |

**List of events:**

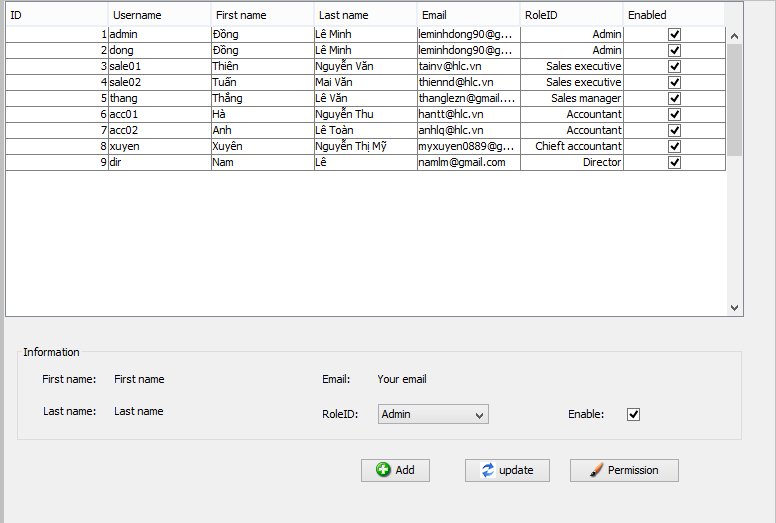
|  |  |
| --- | --- |
| **Event** | **Process** |
| btnconnectActionPerformed | Test connect and connect server |
| btnexitActionPerformed | Exit form |

## 14.3 Home



## 14.4 User

### 14.4.1 User Management



**Description:** this is user list form. can enable/disable user and go to permission form.

**List of objects:**

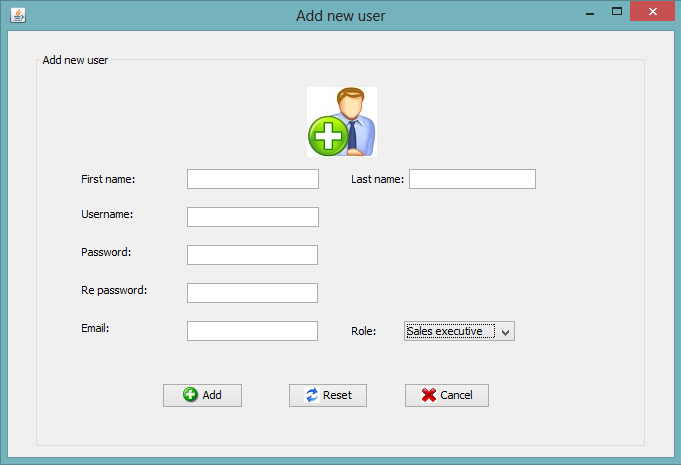
|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | tbllist | JTable | View user list |
| 2 | lblFirstname | JLabel | View Fist name of user selected |
| 3 | lblLastname | JLabel | View last name of user selected |
| 4 | lblEmail | JLabel | View email of user selected |
| 5 | cmbRoleID | JComBoBox | View Rolename of user selected |
| 6 | chkEnable | JCheckBox | View Enable/disable of user selected |
| 7 | btnAdd | JButton | View add new user form |
| 8 | btnUpdate | JButton | update information user |
| 9 | btnPermission | JButton | View Permission user form |

**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnAddActionPerformed | View add new user form |
| btnUpdateActionPerformed | update information user |
| btnpermissionActionPerformed | View Permission user form |

### 

### 14.4.2 Add new user



**Description:** this is add new user form.

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | txtFirstname | JTextField | Input first name |
| 2 | txtLastname | JTextField | Input last name |
| 3 | txtUsername | JTextField | Input username |
| 4 | txtpassword | JPasswordField | Input password |
| 5 | txtRepassword | JPasswordField | Input re password |
| 6 | txtEmail | JTextField | Input email |
| 7 | cmbRolename | JComBoBox | Select Rolename |
| 8 | btnAdd | JButton | Add new user |
| 9 | btnReset | JButton | Reset information form |
| 10 | btnCancel | JButton | Cancel and exit form |

**List of validation:**

|  |  |
| --- | --- |
| **Object** | **Validation** |
| txtFirstname | not null, is string |
| txtLastname | not null, is string |
| txtUsername | not null, is string |
| txtpassword | not null, is string |
| txtrepassword | not null, is string |
| txtEmail | not null, is string |
| cmbRolename | Select a Rolename item on list Rolename |

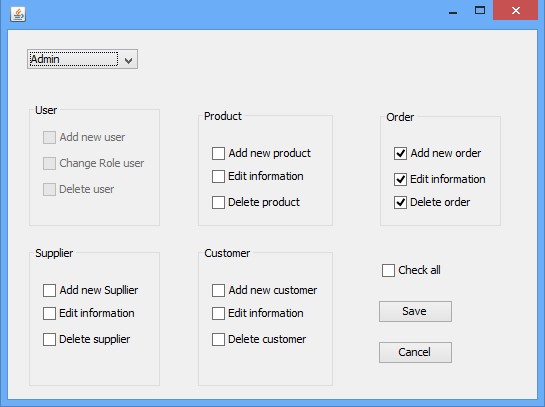
**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnAddActionPerformed | Add new user |
| btnresetActionPerformed | Reset information form |
| btncancelActionPerformed | Cancel and exit form |

## 14.5 Role (not supported yet)

14.6 Function (not supported yet)

## 14.7 Permission



**Description:** this is form permission user by Rolename.  
  
**List of objects:**

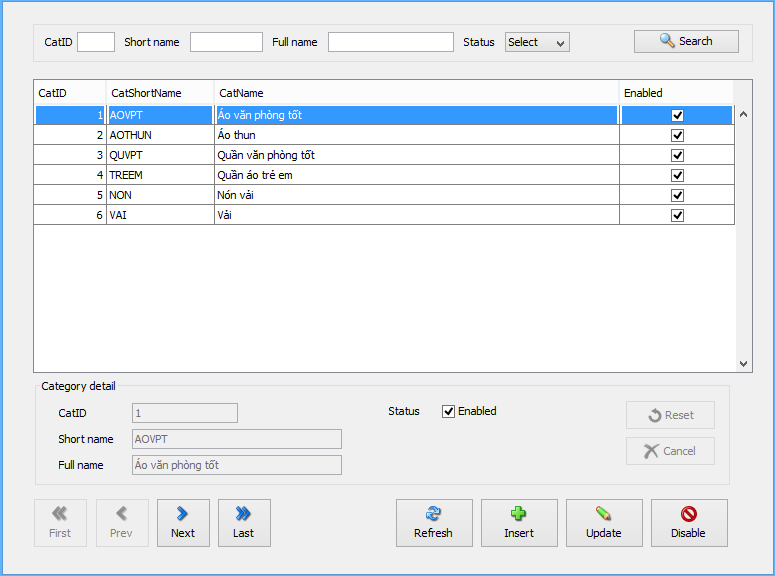
|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | cmbrolename | JComBoBox | Select role name. |
| 2 | chkadduser | JCheckBox | Can add new user |
| 3 | chkchangeuser | JCheckBox | Can change role user |
| 4 | chkdeleteuser | JCheckBox | Can delete user |
| 5 | chkaddproduct | JCheckBox | Can add new product |
| 6 | chkeditproduct | JCheckBox | Can edit information product |
| 7 | chkdeleteproduct | JCheckBox | Can delete product |
| 8 | chkaddorder | JCheckBox | Can add new order |
| 9 | chkeditorder | JCheckBox | Can edit information order |
| 10 | chkdeleteorder | JCheckBox | Can delete order |
| 11 | chkaddsupplier | JCheckBox | Can add new supplier |
| 12 | chkeditsupplier | JCheckBox | Can edit information supplier |
| 13 | chkdeletesupplier | JCheckBox | Can delete supplier |
| 14 | chkaddcustomer | JCheckBox | Can add new customer |
| 15 | chkeditcustomer | JCheckBox | Can edit information customer |
| 16 | chkdeletecustomer | JCheckBox | Can delete customer |
| 17 | chkcheckall | JCheckBox | Check all check box |
| 18 | btntsave | JButton | Save changed |
| 19 | btncancel | JButton | Cancel and dispose form |

**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnsaveActionPerformed | Save permission rolename |
| btncancelActionPerformed | Cancel and dispose form |

## 14.8 Category

### 14.8.1 Category Management



**Description:** This form is used to view the category list, filter, add, update, disable category.

**List of objects:**

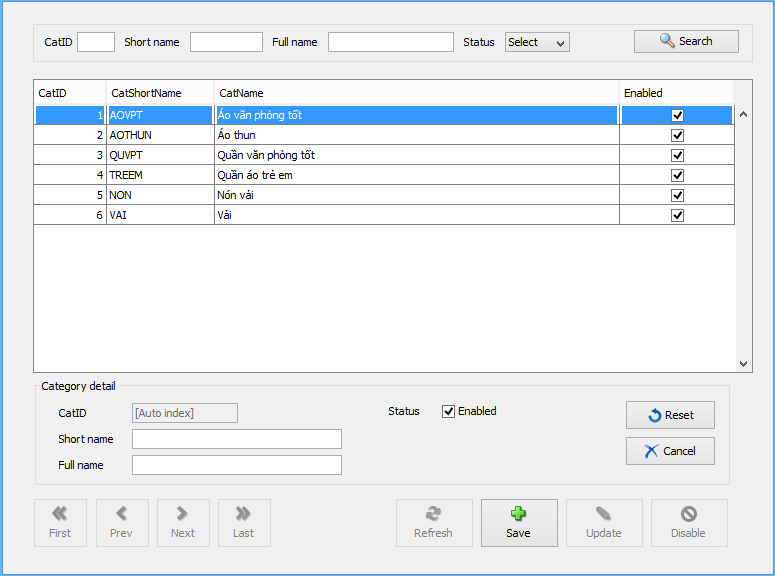
|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | txtCatID2 | JTextField | Input search parameter: CatID |
| 2 | txtShortName2 | JTextField | Input search parameter: CatShortName |
| 3 | txtFullName2 | JTextField | Input search parameter: CatFullName |
| 4 | cmbStatus | JComboBox | Input search parameter: CatStatus |
| 5 | btnSearch | JButton | Search category |
| 6 | tblList | JTable | View the category list |
| 7 | txtCatID | JTextField | View category’s ID |
| 8 | txtCatShortName | JTextField | View category’s short name |
| 9 | txtCatName | JTextField | View category’s full name |
| 10 | chkEnabled | JCheckBox | View category’s enable status |
| 11 | btnFirst | JButton | Go to the first row of table |
| 12 | btnPrevious | JButton | Go to the previous row of table |
| 13 | btnNext | JButton | Go to the next row of table |
| 14 | btnLast | JButton | Go to the last row of table |
| 15 | btnReset | JButton | Reset form entries when filter, insert, update |
| 16 | btnCancel | JButton | Set form back to view-mode |
| 17 | btnRefresh | JButton | Reload data from database |
| 18 | btnInsert | JButton | Set form to insert-mode/ Insert to database |
| 19 | btnUpdate | JButton | Set form to update-mode/ Update database |
| 20 | btnDisable | JButton | Toogle selected product status (True/ False) |

**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnFirstActionPerformed | Go to the first row of table |
| btnPreviousActionPerformed | Go to the previous row of table |
| btnNextActionPerformed | Go to the next row of table |
| btnLastActionPerformed | Go to the last row of table |
| btnRefreshActionPerformed | Reload data from database |
| btnSearchActionPerformed | Search category |
| btnInsertActionPerformed | Set form to insert-mode |
| btnUpdateActionPerformed | Set form to update-mode |
| btnDisableActionPerformed | Toogle selected category status (True/ False) |
| btnResetActionPerformed | Reset form entries |
| btnCancelActionPerformed | Set form to view-mode, cancel current mode |
| tblListMouseClicked | Detect table selection change & view the corresponding category details |

### 

### 14.8.2 Category Insert



**Description:** This view is used to insert/ add category. Activated when user click Insert button.

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | tblList | JTable | View the category list (disabled) |
| 2 | txtCatID | JTextField | Category’s ID (disabled) |
| 3 | txtCatShortName | JTextField | Input data to insert: category’s short name |
| 4 | txtCatName | JTextField | Input data to insert: category’s full name |
| 5 | chkEnabled | JCheckbox | Input data to insert: category’s status |
| 6 | btnReset | JButton | Reset form entries |
| 7 | btnCancel | JButton | Set form back to view-mode |
| 8 | btnSave | JButton | Save new entry to database |

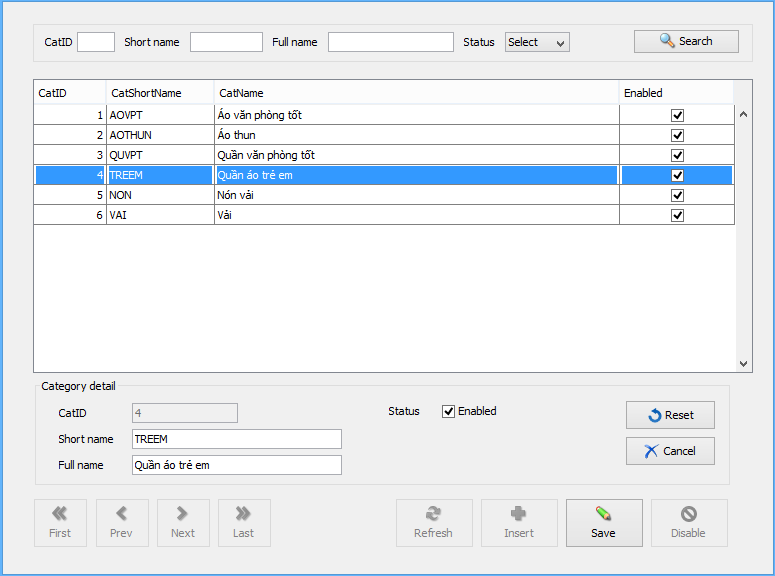
**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnSaveActionPerformed | Insert to database |
| btnResetActionPerformed | Reset form entries |
| btnCancelActionPerformed | Set form to view-mode, cancel current mode |

**List of Validations:**

|  |  |
| --- | --- |
| **Object** | **Validation** |
| txtCatShortName | not null, is string |
| txtCatName | not null, is string |

### 14.8.3 Category Update



**Description:** This view is used to update/ edit selected category information.

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | tblList | JTable | View the category list (disabled) |
| 2 | txtCatID | JTextField | Category’s ID (disabled) |
| 3 | txtCatShortName | JTextField | Input data to update: category’s short name |
| 4 | txtCatName | JTextField | Input data to update: category’s full name |
| 5 | chkEnabled | JCheckbox | Input data to update: category’s status |
| 6 | btnReset | JButton | Reset form entries |
| 7 | btnCancel | JButton | Set form back to view-mode |
| 8 | btnSave | JButton | Update to database |

**List of events:**

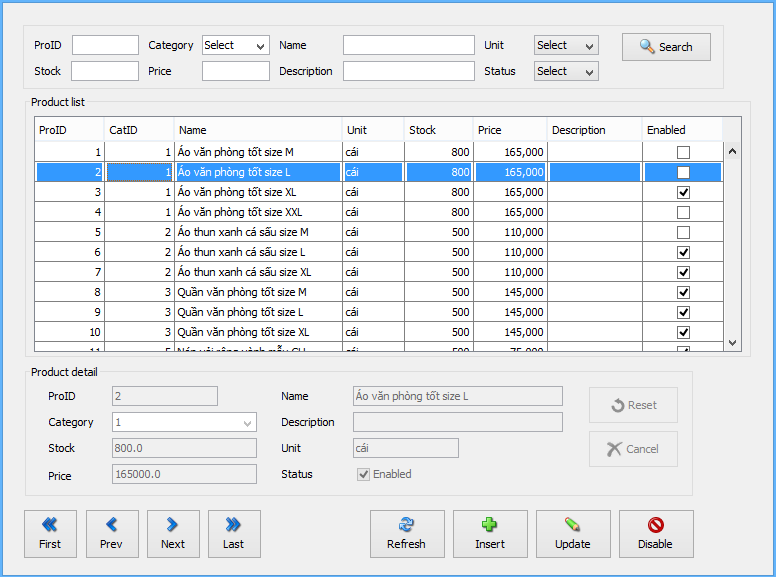
|  |  |
| --- | --- |
| **Event** | **Process** |
| btnSaveActionPerformed | Update to database |
| btnResetActionPerformed | Reset form entries |
| btnCancelActionPerformed | Set form to view-mode, cancel current mode |

**List of Validations:**

|  |  |
| --- | --- |
| **Object** | **Validation** |
| txtCatShortName | not null, is string |
| txtCatName | not null, is string |
| chkEnabled | Can disable only if none product linked to this category |

## 14.9 Product

### 14.9.1 Product Management



**Description:** This form is used to view the product list, filter, add, update, disable product.

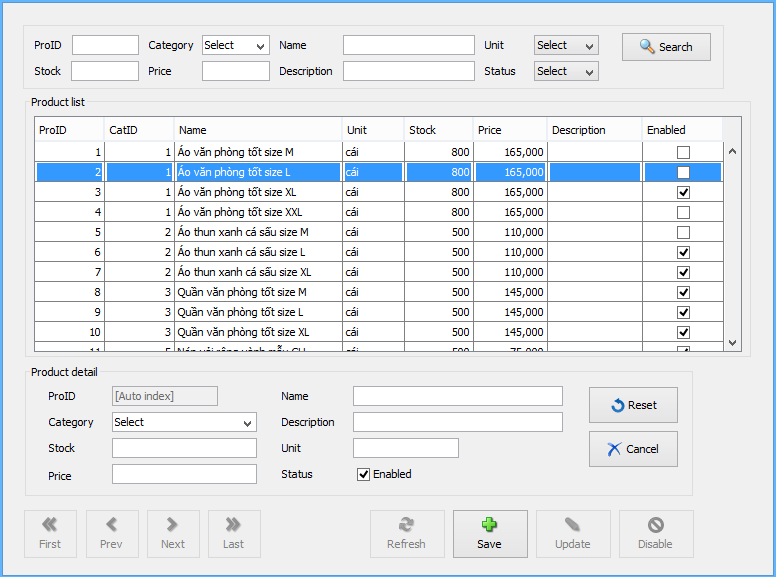
**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | txtProID2 | JTextField | Input search parameter: product’s ID |
| 2 | cmbCategory | JComboBox | Input search parameter: product’s Category ID |
| 3 | txtProName2 | JTextField | Input search parameter: product’s name |
| 4 | cmbUnit | JComboBox | Input search parameter: product’s unit |
| 5 | txtProStock2 | JTextField | Input search parameter: product’s stock quantity |
| 6 | txtProPrice2 | JTextField | Input search parameter: product’s price |
| 7 | txtProDescr | JTextField | Input search parameter: product’s description |
| 8 | cmbEnabled | JComboBox | Input search parameter: product’s enable status |
| 9 | btnSearch | JButton | Search product |
| 10 | tblList | JTable | View the product list |
| 11 | txtProID | JTextField | View product’s ID |
| 12 | cmbCategory | JComboBox | View product’s Category ID |
| 13 | txtProName | JTextField | View product’s name |
| 14 | txtProUnit | JTextField | View product’s unit |
| 15 | txtProStock | JTextField | View product’s stock quantity |
| 16 | txtProPrice | JTextField | View product’s price |
| 17 | txtProDescr | JTextField | View product’s description |
| 18 | chkProEnabled | JCheckBox | View product’s enable status |
| 19 | btnFirst | JButton | Go to the first row of table |
| 20 | btnPrevious | JButton | Go to the previous row of table |
| 21 | btnNext | JButton | Go to the next row of table |
| 22 | btnLast | JButton | Go to the last row of table |
| 23 | btnReset | JButton | Reset form entries when filter, insert, update |
| 24 | btnCancel | JButton | Set form back to view-mode |
| 25 | btnRefresh | JButton | Reload data from database |
| 26 | btnInsert | JButton | Set form to insert-mode/ Insert to database |
| 27 | btnUpdate | JButton | Set form to update-mode/ Update database |
| 28 | btnDisable | JButton | Toogle selected product status (True/ False) |

**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnFirstActionPerformed | Go to the first row of table |
| btnPreviousActionPerformed | Go to the previous row of table |
| btnNextActionPerformed | Go to the next row of table |
| btnLastActionPerformed | Go to the last row of table |
| btnRefreshActionPerformed | Reload data from database |
| btnSearchActionPerformed | Search product |
| btnInsertActionPerformed | Set form to insert-mode |
| btnUpdateActionPerformed | Set form to update-mode |
| btnDisableActionPerformed | Toogle selected product status (True/ False) |
| btnResetActionPerformed | Reset form entries |
| btnCancelActionPerformed | Set form to view-mode, cancel current mode |
| tblListMouseClicked | Detect table selection change & view that product’s details |

### 14.9.2 Product Insert



**Description:** This view is used to insert/ add product. Activated when user click Insert button from Product management form.

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | tblList | JTable | View the product list (disabled) |
| 2 | txtProID | JTextField | Product’s ID (disabled) |
| 3 | cmbCategory | JComboBox | Input data to insert: product’s Category ID |
| 4 | txtProName | JTextField | Input data to insert: product’s name |
| 5 | txtProUnit | JTextField | Input data to insert: product’s unit |
| 6 | txtProStock | JTextField | Input data to insert: product’s stock quantity |
| 7 | txtProPrice | JTextField | Input data to insert: product’s price |
| 8 | chkProEnabled | JCheckBox | Input data to insert: product’s enable status |
| 9 | btnReset | JButton | Reset form entries |
| 10 | btnCancel | JButton | Set form back to view-mode |
| 11 | btnSave | JButton | Save new entry to database |

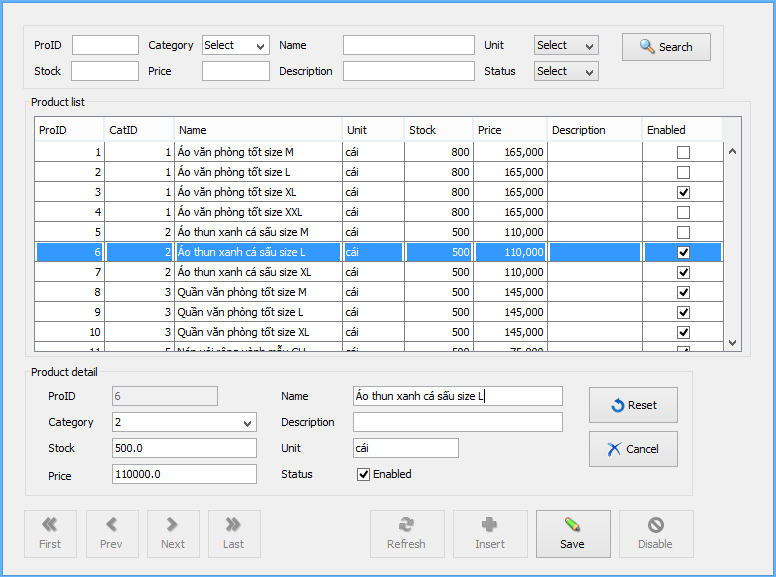
**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnSaveActionPerformed | Insert to database |
| btnResetActionPerformed | Reset form entries |
| btnCancelActionPerformed | Set form to view-mode, cancel current mode |

**List of Validations:**

|  |  |
| --- | --- |
| **Object** | **Validation** |
| cmbCategory | not null, is number |
| txtProName | not null, is string |
| txtProUnit | not null, is string |
| txtProStock | not null, is number (>=0) |
| txtProPrice | not null, is number (>=0) |

### 14.9.3 Product Update



**Description:** This view is used to update/ edit selected product information.

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | tblList | JTable | View the product list (disabled) |
| 2 | txtProID | JTextField | Product’s ID (disabled) |
| 3 | cmbCategory | JComboBox | Input data to update: product’s Category ID |
| 4 | txtProName | JTextField | Input data to update: product’s name |
| 5 | txtProUnit | JTextField | Input data to update: product’s unit |
| 6 | txtProStock | JTextField | Input data to update: product’s stock quantity |
| 7 | txtProPrice | JTextField | Input data to update: product’s price |
| 8 | chkProEnabled | JCheckBox | Input data to update: product’s enable status |
| 9 | btnReset | JButton | Reset form entries |
| 10 | btnCancel | JButton | Set form back to view-mode |
| 11 | btnUpdate | JButton | Save edited entry to database |

**List of events:**

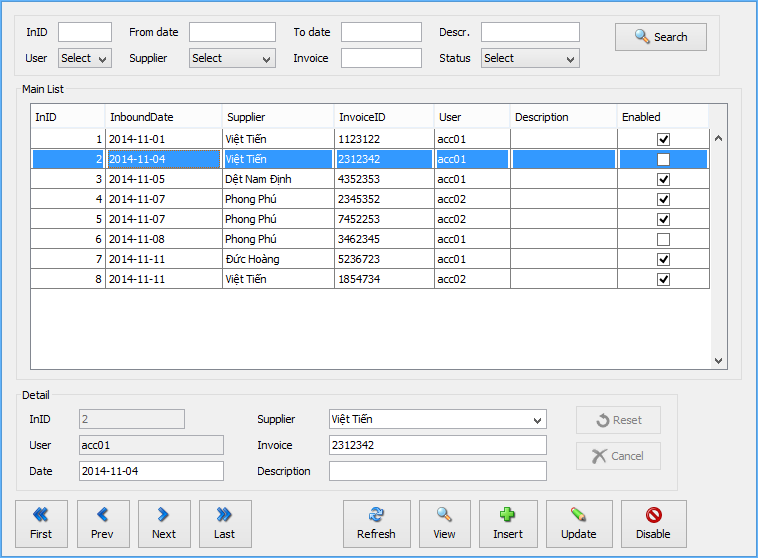
|  |  |
| --- | --- |
| **Event** | **Process** |
| btnUpdateActionPerformed | Update to database |
| btnResetActionPerformed | Reset form entries |
| btnCancelActionPerformed | Set form to view-mode, cancel current mode |

**List of Validations:**

|  |  |
| --- | --- |
| **Object** | **Validation** |
| cmbCategory | not null, is number |
| txtProName | not null, is string |
| txtProUnit | not null, is string |
| txtProStock | not null, is number (>=0) |
| txtProPrice | not null, is number (>=0) |

## 14.10 Inbound Note

### 14.10.1 Inbound Note Management



**Description:** This form is used to view the inbound note list, filter, add, update, disable inbound notes.

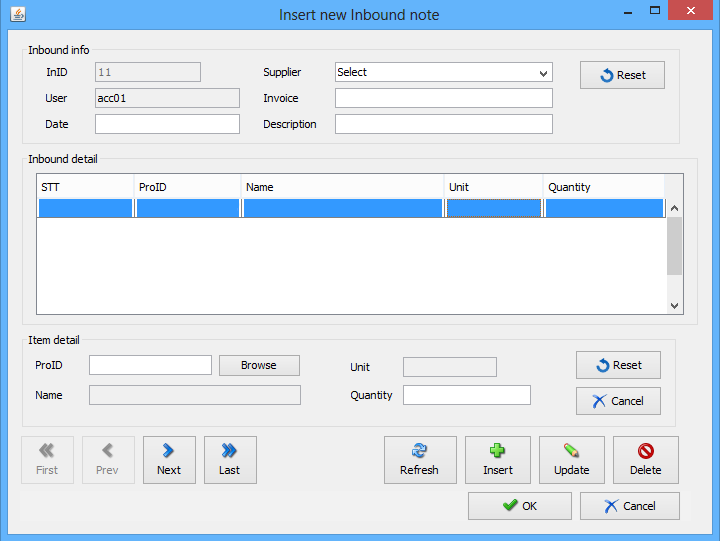
**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | txtID | JTextField | Input search parameter: InID |
| 2 | cmbUser | JComboBox | Input search parameter: UserID |
| 3 | txtFromDate | JTextField | Input search parameter: Min InboundDate |
| 4 | txtToDate | JTextField | Input search parameter: Max InboundDate |
| 5 | txtDescr2 | JTextField | Input search parameter: InDescr |
| 6 | cmbUser | JComboBox | Input search parameter: UserID |
| 7 | cmbSupplier2 | JComboBox | Input search parameter: SupplID |
| 8 | txtInvoice2 | JTextField | Input search parameter: InvoiceID |
| 9 | btnSearch | JButton | Search inbound note |
| 10 | tblList | JTable | View the inbound note list |
| 11 | txtID | JTextField | View inbound note’s ID |
| 12 | txtUser | JTextField | View inbound note’s user (whom created note) |
| 13 | txtInboundDate | JTextField | View inbound note’s date |
| 14 | cmbSupplier | JComboBox | View inbound note’s supplier |
| 15 | txtInvoice | JTextField | View inbound note’s invoice |
| 16 | txtDescr | JTextField | View inbound note’s description |
| 17 | btnFirst | JButton | Go to the first row of table |
| 18 | btnPrevious | JButton | Go to the previous row of table |
| 19 | btnNext | JButton | Go to the next row of table |
| 20 | btnLast | JButton | Go to the last row of table |
| 21 | btnReset | JButton | Reset form entries when filter, insert, update |
| 22 | btnCancel | JButton | Set form back to view-mode |
| 23 | btnRefresh | JButton | Reload data from database |
| 24 | btnView | JButton | Set form to view-mode |
| 25 | btnInsert | JButton | Set form to insert-mode |
| 26 | btnUpdate | JButton | Set form to update-mode |
| 27 | btnDisable | JButton | Toogle selected inbound note status (True/ False) |

**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnFirstActionPerformed | Go to the first row of table |
| btnPreviousActionPerformed | Go to the previous row of table |
| btnNextActionPerformed | Go to the next row of table |
| btnLastActionPerformed | Go to the last row of table |
| btnRefreshActionPerformed | Reload data from database |
| btnSearchActionPerformed | Search inbound note |
| btnViewActionPerformed | Set form to view-mode |
| btnInsertActionPerformed | Set form to insert-mode |
| btnUpdateActionPerformed | Set form to update-mode |
| btnDisableActionPerformed | Toogle selected inbound note status (True/ False) |
| btnResetActionPerformed | Reset form entries |
| btnCancelActionPerformed | Set form to view-mode, cancel current mode |
| tblListMouseClicked | Detect table selection change & view the details |

### 14.10.2 Inbound note Insert



**Description:** This form is used to insert/ add inbound note & detail.

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | txtID | JTextField | View inbound note’s ID (auto index) |
| 2 | txtUser | JTextField | View current user (not editable) |
| 3 | txtInboundDate | JTextField | Input inbound note’s date |
| 4 | cmbSupplier | JComboBox | Input inbound note’s supplier |
| 5 | txtInvoice | JTextField | Input inbound note’s invoice |
| 6 | txtDescr | JTextField | Input inbound note’s description |
| 7 | btnReset | JButton | Reset inbound note main info entries |
| 8 | tblList | JTable | View the inbound note detail list |
| 9 | txtProID | JTextField | View product’s ID |
| 10 | btnBrowse | JButton | View product list to select |
| 11 | txtName | JTextField | View product’s name (not editable) |
| 12 | txtUnit | JTextField | View product’s unit (not editable) |
| 13 | txtQty | JTextField | View product quantity |
| 14 | btnReset2 | JButton | Reset form entries |
| 15 | btnCancel2 | JButton | Set “Item detail” panel back to view-mode |
| 16 | btnRefresh | JButton | Refresh form |
| 17 | btnInsert | JButton | Set “Item detail” panel to insert-mode / Add new row to table |
| 18 | btnUpdate | JButton | Set “Item detail” panel to update-mode / Update selected row to table |
| 19 | btnDelete | JButton | Delete selected row from table |
| 20 | btnOK | JButton | Insert Inbound note & detail to database |
| 21 | btnCancel | JButton | Close form, back to view Inbound note List |

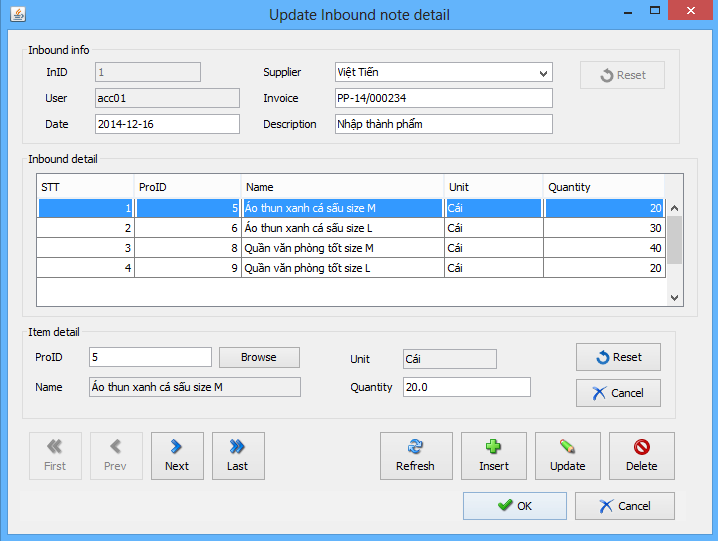
**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnResetActionPerformed | Reset inbound note main info entries |
| btnInsertActionPerformed | Set “Item detail” panel to insert-mode / Add new row to table |
| btnReset2ActionPerformed | Reset “Item detail” panel entries |
| btnCancel2ActionPerformed | Set “Item detail” panel to view-mode, cancel current mode |
| btnOKActionPerformed | Insert Inbound note & Inbound note detail to database |
| btnCancelActionPerformed | Close form, back to view Inbound note List |

**List of Validations:**

|  |  |
| --- | --- |
| **Object** | **Validation** |
| txtInboundDate | Not null, is date |
| txtSupplier | Not null |
| tblList | At least one row |
| txtProID | Not null, is number, existed product |
| txtQty | Not null, is number > 0 |

### 14.10.3 Inbound note Update



**Description:** This form is used to update/ edit inbound note information & detail.

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | txtID | JTextField | View inbound note’s ID (not editable) |
| 2 | txtUser | JTextField | View current user (not editable) |
| 3 | txtInboundDate | JTextField | Input data to update: inbound note’s date |
| 4 | cmbSupplier | JComboBox | Input data to update: inbound note’s supplier |
| 5 | txtInvoice | JTextField | Input data to update: inbound note’s invoice |
| 6 | txtDescr | JTextField | Input data to update: inbound note’s description |
| 7 | btnReset | JButton | Reset inbound note main info entries |
| 8 | tblList | JTable | View the inbound note detail list |
| 9 | txtProID | JTextField | View product’s ID |
| 10 | btnBrowse | JButton | View product list to select |
| 11 | txtName | JTextField | View product’s name (not editable) |
| 12 | txtUnit | JTextField | View product’s unit (not editable) |
| 13 | txtQty | JTextField | View product quantity |
| 14 | btnReset2 | JButton | Reset form entries |
| 15 | btnCancel2 | JButton | Set “Item detail” panel back to view-mode |
| 16 | btnRefresh | JButton | Refresh form |
| 17 | btnInsert | JButton | Set “Item detail” panel to insert-mode / Add new row to table |
| 18 | btnUpdate | JButton | Set “Item detail” panel to update-mode / Update selected row to table |
| 19 | btnDelete | JButton | Delete selected row from table |
| 20 | btnOK | JButton | Update Inbound note & Inbound note detail to database |
| 21 | btnCancel | JButton | Close form, back to view Inbound note List |

**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnResetActionPerformed | Reset inbound note main info entries |
| btnInsertActionPerformed | Set “Item detail” panel to insert-mode / Add new row to table |
| btnReset2ActionPerformed | Reset “Item detail” panel entries |
| btnCancel2ActionPerformed | Set “Item detail” panel to view-mode, cancel current mode |
| btnOKActionPerformed | Update Inbound note & Inbound note detail to database |
| btnCancelActionPerformed | Close form, back to view Inbound note List |

**List of Validations:**

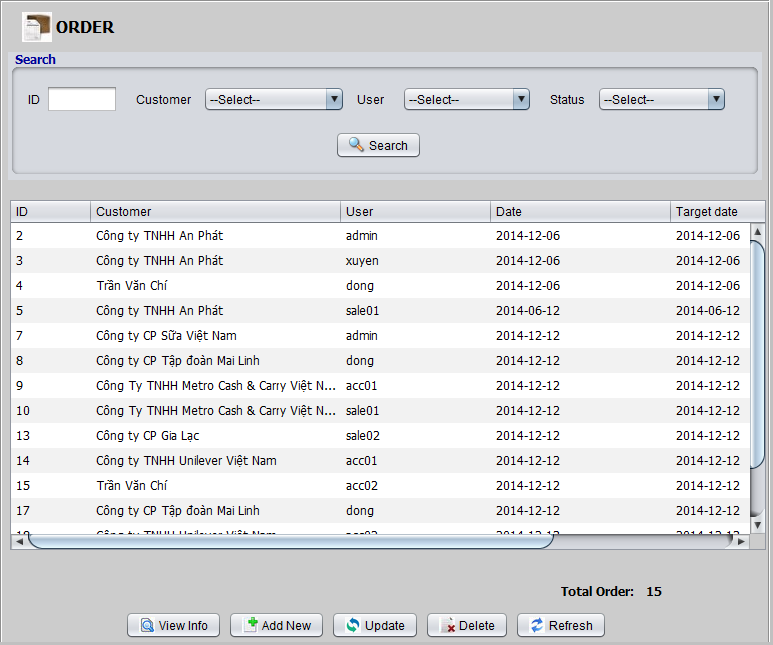
|  |  |
| --- | --- |
| **Object** | **Validation** |
| txtInboundDate | Not null, is date |
| txtSupplier | Not null |
| tblList | At least one row |
| txtProID | Not null, is number, existed product |
| txtQty | Not null, is number > 0 |

## 14.11 Outbound (not supported yet)

## 14.12 Status (not supported yet)

## 14.13 Order

### 14.13.1 Order Management



**Description:** This form is used to view the order list, search order, add, update, delete/ disable order.

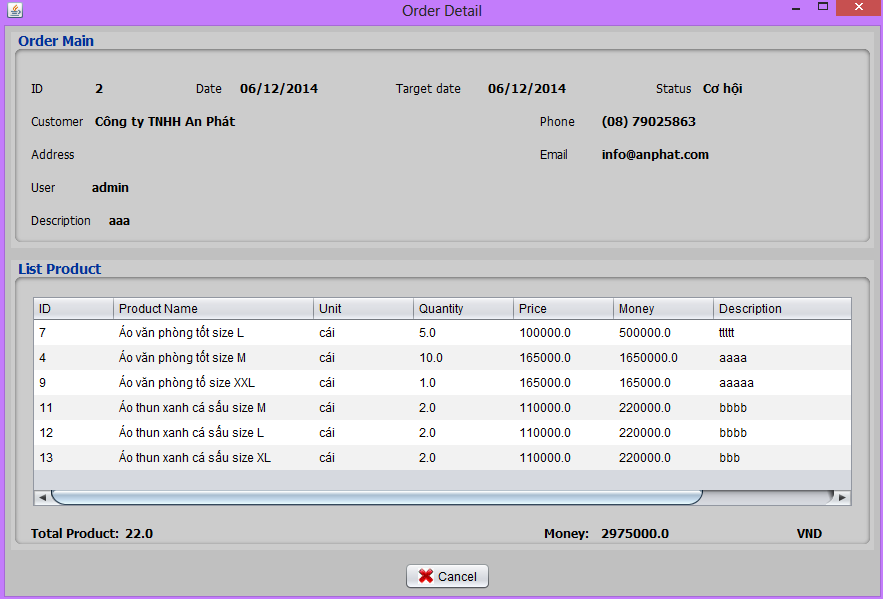
**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | txtID | JTextField | Fill in the ID of order want to search |
| 2 | cmbCus | JComboBox | Select customer of order want to search |
| 3 | cmbUser | JComboBox | Select user of order want to search |
| 4 | cmbStatus | JComboBox | Select status of order want to search |
| 5 | tabOrder | JTable | Display list order |
| 6 | btnView | JButton | Display information of an order |
| 7 | btnAdd | JButton | Add new order |
| 8 | btnUpdate | JButton | Update an order |
| 9 | btnDelete | JButton | Delete an order |
| 10 | btnRefresh | JButton | Refresh list order then Add, Update |

**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnSearchActionPerformed | Mouse click on button search then show orders want to search |
| btnViewInfo ActionPerformed | Select a line want to see then click the view button will display detailed information form the order |
| btnAddNewActionPerformed | Mouse click on button then show form add new an order |
| btnUpdateActionPerformed | Click the button and then displays information form users update the details of the order |
| btnDeleteActionPerformed | Select a line want delete then click the delete button |
| btnRefreshActionPerformed | After adding or updating click button to refresh the list of orders |

### 14.13.2 View Order Information



**Description:** This form is used to display detailed information from the order.

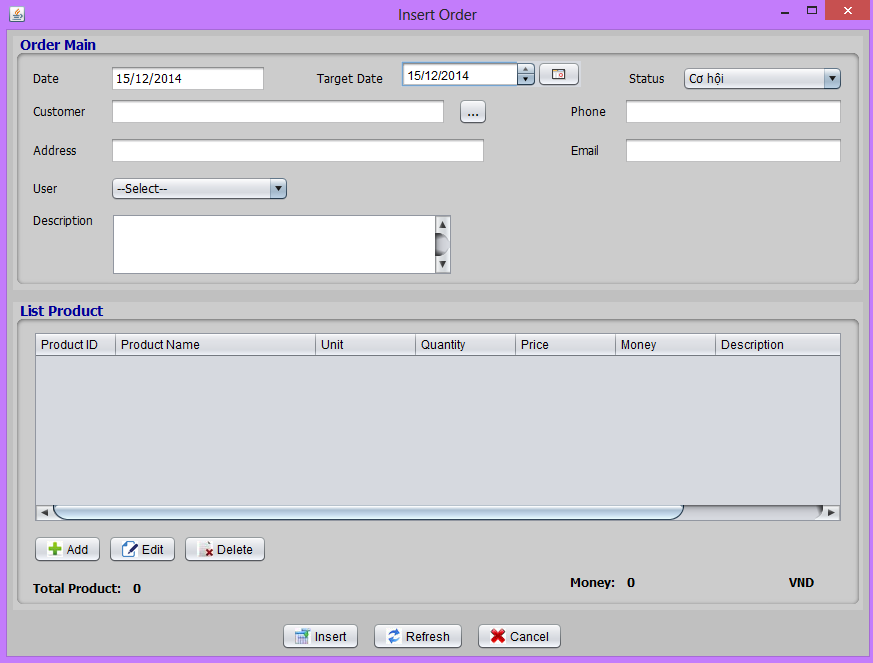
**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | label | JLabel | Display information of orders |
| 2 | tableProduct | JTable | Displays a list of products in order |
| 3 | btnCancel | JButton | Used to close the form |

**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnCancelActionPerformed | Mouse click on button to close the from being displayed |

### 14.13.3 Insert Order



**Description:** This form is used to add new an order.

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | txtDate | JTextField | Default get current system date |
| 2 | dateTarget | JDateChooser | Choose the date of receipt of the product |
| 3 | cmbStatus | JComboBox | Select status of order |
| 4 | txtCustomer | JTextField | Display customer name |
| 5 | btnCus | JButton | Click to view list customer |
| 6 | txtPhone | JTextField | Display customer phone |
| 7 | txtAddress | JTextField | Display customer address |
| 8 | txtEmail | JTextField | Display customer email |
| 9 | cmbUser | JComboBox | Display users to enter order |
| 10 | txtDescr | JTextArea | Input description an order |
| 11 | tableProduct | JTable | Display list product |
| 12 | btnAdd | JButton | Add new a product |
| 13 | btnEdit | JButton | Edit a product in list product |
| 14 | btnDelete | JButton | Delete a product in list product |
| 15 | labTotal | JLabel | Display total product in order |
| 16 | labMoney | JLabel | Display total money in order |
| 17 | btnInsert | JButton | Insert order in database |
| 18 | btnRefresh | JButton | Delete all the data entered |
| 19 | btnCancel | JButton | Close form |

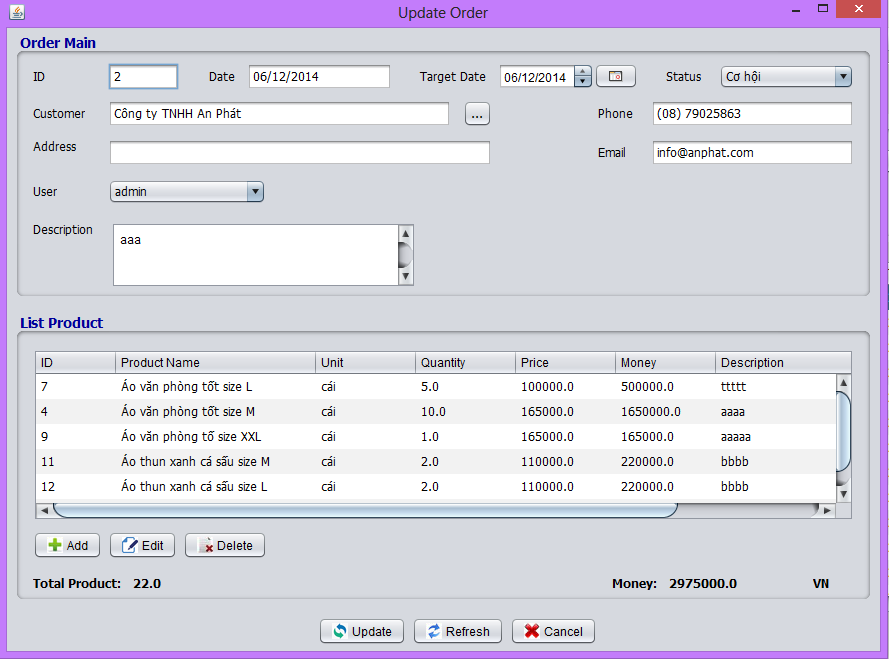
**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnCusActionPerformed | Click to view list customer |
| btnAddActionPerformed | Click to view form add new a product |
| btnEditActionPerformed | Click to view form edit a product in table product |
| btnDeleteActionPerformed | Click to delete a product in table product |
| btnInsertActionPerformed | Click to insert all information in database |
| btnRefreshActionPerformed | Click to refresh form |
| btnCancelActionPerformed | Click to close form |

**List of Validations:**

|  |  |
| --- | --- |
| **Object** | **Validation** |
| txtDate | Get date current, not null |
| dateTarget | Date target after date |
| txtCustomer | Not null |
| cmbUser | Select a user other item select |
| tableProduct | At least one product |

### 14.13.4 Update Order



**Description:** This form is used to update an order.

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | txtID | JTextField | Non edit |
| 2 | txtDate | JTextField | Non edit |
| 3 | dateTarget | JDateChooser | Choose the date of receipt of the product |
| 4 | cmbStatus | JComboBox | Select status of order |
| 5 | txtCustomer | JTextField | Display customer name |
| 6 | btnCus | JButton | Click to view list customer |
| 7 | txtPhone | JTextField | Display customer phone |
| 8 | txtAddress | JTextField | Display customer address |
| 9 | txtEmail | JTextField | Display customer email |
| 10 | cmbUser | JComboBox | Display users to enter order |
| 11 | txtDescr | JTextArea | Input description an order |
| 12 | tableProduct | JTable | Display list product |
| 13 | btnAdd | JButton | Add new a product |
| 14 | btnEdit | JButton | Edit a product in list product |
| 15 | btnDelete | JButton | Delete a product in list product |
| 16 | labTotal | JLabel | Display total product in order |
| 17 | labMoney | JLabel | Display total money in order |
| 18 | btnUpdate | JButton | Update order in database |
| 19 | btnRefresh | JButton | Delete all the data entered |
| 20 | btnCancel | JButton | Close form |

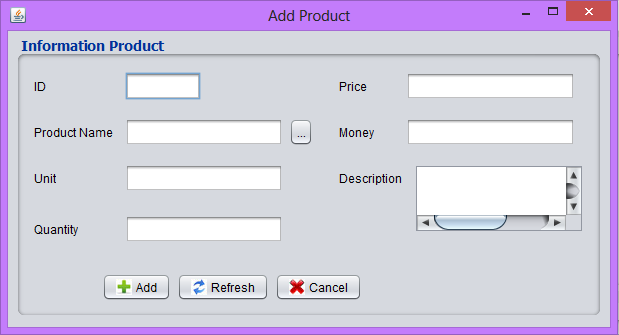
**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnCusActionPerformed | Click to view list customer |
| btnAddActionPerformed | Click to view form add new a product |
| btnEditActionPerformed | Click to view form edit a product in table product |
| btnDeleteActionPerformed | Click to delete a product in table product |
| btnUpdateActionPerformed | Click to update all information in database |
| btnRefreshActionPerformed | Click to refresh form |
| btnCancelActionPerformed | Click to close form |

**List of Validations:**

|  |  |
| --- | --- |
| **Object** | **Validation** |
| txtCustomer | not null |
| dateTarget | Date target after date |
| tableProduct | at least one product |

### 14.13.5 Add product to order



**Description:** This form is used to add new a product in order.

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | txtID | JTextField | Not edit |
| 2 | txtPro | JTextField | Display product name |
| 3 | btnPro | JButton | View product list |
| 4 | Unit | JTextField | Product unit |
| 5 | txtQty | JTextField | Ordered quantity |
| 6 | txtPrice | JTextField | Price of one product |
| 7 | txtmoney | JTextField | The total amount of product |
| 8 | txaDersc | JTextArea | Description product |

**List of events:**

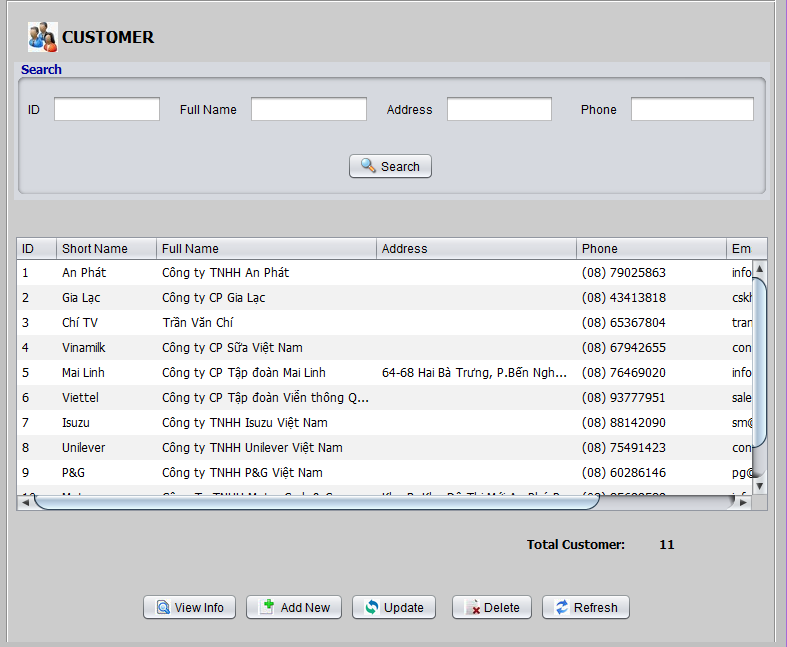
|  |  |
| --- | --- |
| **Event** | **Process** |
| btnProActionPerformed | View list product to select |
| btnAddActionPerformed | Add new product in table in order |
| btnRefreshActionPerformed | Delete all data in form |
| btnCancelActionPerformed | Close form |

**List of Validations:**

|  |  |
| --- | --- |
| **Object** | **Validation** |
| txtID | not null |
| txtQty | not null, is number |
| txtPrice | not null, is number |

## 14.14 Customer

### 14.14.1 Customer Management



**Description:** This form is used to view the order list, search order, add, update, delete/ disable order.

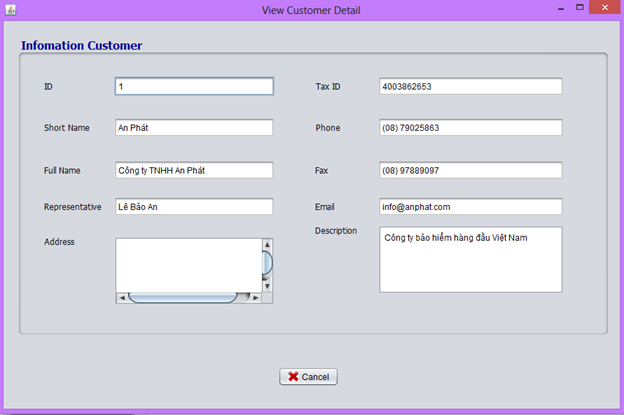
**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | txtID | JTextField | Fill in the ID of customer want to search |
| 2 | txtname | JTextField | Fill in the name of customer want to search |
| 3 | txtAddress | JTextField | Fill in the address of customer want to search |
| 4 | txtPhone | JTextField | Fill in the phone of customer want to search |
| 5 | btnSearch | JButton | Click to search customer |
| 6 | btnView | JButton | Display information of a customer |
| 7 | btnAdd | JButton | Add new a customer |
| 8 | btnUpdate | JButton | Update a customer |
| 9 | btnDelete | JButton | Delete a customer |
| 10 | btnRefresh | JButton | Refresh list customer then Add, Update |

**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnViewActionPerformed | Display form information a customer |
| btnAddActionPerformed | Display form add new a customer |
| btnUpdateActionPerformed | Display form update a customer |
| btnDeleteActionPerformed | Delete a customer in list |
| btnRefreshActionPerformed | Refresh list customer then add, update |

### 14.14.2 View Customer Detail



**Description:** This form is used to view the product list, add, update, delete/ disable product.

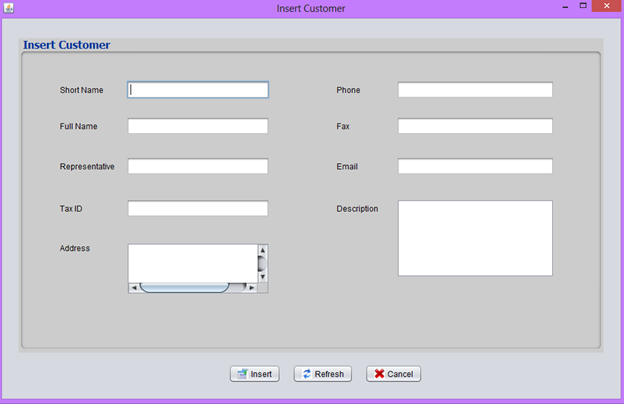
**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | TextField | JTextField | Display information of customer |
| 2 | btnCancel | JButton | To close form |

**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnCancel  ActionPerformed | Click to close form |

### 14.14.3 Insert Customer



**Description:** This form is used to add new a customer

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | TextField | JTextField | Enters information of customer |
| 2 | btnInsert | JButton | Insert data in database |
| 3 | btnRefresh | JButton | Refresh form |
| 4 | btnCancel | JButton | Close from |

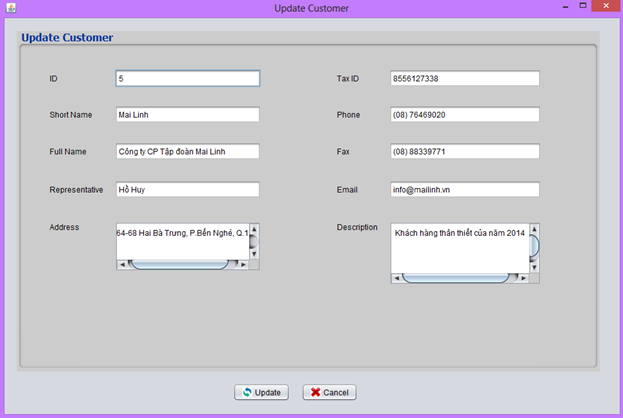
**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnInsertActionPerformed | Get all data in from insert in database and close from |
| btnRefreshActionPerformed | Delete all the data entered in form |
| btnCancelActionPerformed | Close form |

**List of Validations:**

|  |  |
| --- | --- |
| **Object** | **Validation** |
| txtName, txtFullName, txtTaxID | not null |
| txtEmail | null, if input must have right format |

### 14.14.4 Update Customer



**Description:** This form is used to update a customer

**List of objects:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Description** |
| 1 | TextField | JTextField | Display information of customer |
| 2 | btnUpdate | JButton | Update data in database |
| 3 | btnCancel | JButton | Close form |

**List of events:**

|  |  |
| --- | --- |
| **Event** | **Process** |
| btnUpdateActionPerformed | Get all data in from update in database and close from |
| btnCancelActionPerformed | Close form |

**List of Validations:**

|  |  |
| --- | --- |
| **Object** | **Validation** |
| txtName, txtFullName, txtTaxID | not null |
| txtEmail | null, if input must have right format |

# 

# 15 Task Sheet Review 3

|  |  |  |  |
| --- | --- | --- | --- |
| **Group members** | **Description of works** | | |
| **Work** | | **Status** |
| Le Van Thang  (student806165) | 1.6. Category  1.6.1. Category Management  1.6.2. Category Filter  1.6.3. Category Insert  1.6.4. Category Update  1.7. Product  1.7.1. Product Management  1.7.2. Product Filter  1.7.3. Product Insert  1.7.4. Product Update  1.8. Inbound Note  1.8.1. Inbound Note Management  1.8.2. Inbound note Filter  1.8.3. Inbound note Insert  1.8.4. Inbound note Update | | Completed |
| Le Minh Dong  (student806079) | 1.2. User  1.2.1. Login  1.2.2. User Management  1.2.3. Add new user  1.2.4. Server connect  1.5. Permission | | Completed |
| Nguyen Thi My Xuyen  (student803089) | 1.11. Order  1.11.1. Order Management  1.11.2. View Order Information  1.11.3. Insert Order  1.11.4. Update Order  1.11.5. Add product to order  1.12. Customer  1.12.1. Customer Management  1.12.2. View Customer Detail  1.12.3. Insert Customer  1.12.4. Update Customer | | Completed |
| **Team Leader Signature** | | **Approved by Faculty** | |
|  | |  | |
| Mr. Le Van Thang | | Mr. Le Hong Ky | |