|  |  |
| --- | --- |
| **Project Case** |  |
| ISYS6197  Business Application Development |
| **Computer Science** | **O212-ISYS6197-TJ05-00** |
| ***Valid on*** *Odd Semester Year 2020/2021* | **Revision 00** |

1. Seluruh kelompok tidak diperkenankan untuk:

*The whole group is not allowed to:*

* + 1. Melihat sebagian atau seluruh proyek kelompok lain,

*Seeing a part or the whole project from other groups*

* + 1. Menyadur sebagian maupun seluruh proyek dari buku,

*Adapted a part or the whole project from the book*

* + 1. Mendownload sebagian maupun seluruh proyek dari internet,

*Downloading a part or the whole project from the internet,*

* + 1. Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal proyek,

*Working with another theme which is not in accordance with the existing theme in the matter of the project,*

* + 1. Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + 1. Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika kelompok terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai kelompok** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the group is proved to the actions described in point 1 above, the score of the group which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan proyek, segala jenis pengumpulan proyek di luar jadwal tidak dilayani.

*Pay attention to the submission schedule for the project, all kinds of submission outside the project schedule will not be accepted*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| 40% | 60% | - |

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

|  |
| --- |
| **Software**  *Software* |
| Java 8  Eclipse 2020.6  XAMPP 7.4.7  MySQL Java Connection Library 5.1.49 |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri, proyek dan uap untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment, project, and final exam collection for this subject are described as follows:*

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* |
| JAVA, CLASS | JAVA, CLASS, SQL |

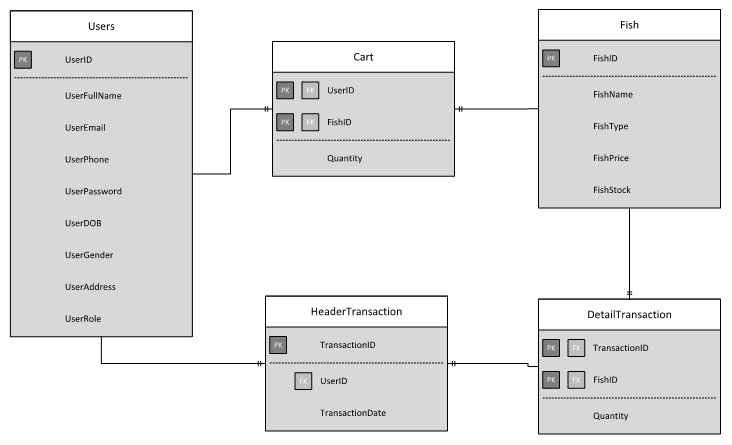
## Soal

*Case*

**Fish Shop**

**Fish Shop** is a shop that sells freshwater, seawater, and brackish fish. To manage all the transactions and manage system of the fish, Fish Shop’s owner asked you to create the program for the store. The program should be made using **Java Programming Language**.

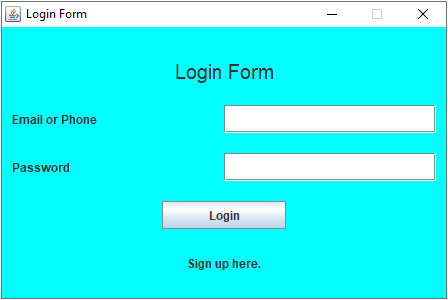
There are some requirements to build this application:



Execute the provided create and insert query file on “**fish\_shop**” database! (“create+insert.sql”)

1. **Login Form**

This is the first form which will appear when the program is run. **Login** **Form** is used by users to **Login** so the users can access the **Main** **Form**.



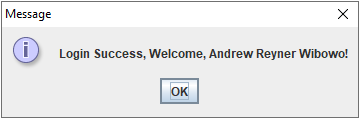
**Figure 1. Login Form**

**Components**:

* **JLabel** for **Title**, **Email or Phone**, **Password**, and **Sign Up**.
* **JTextField** for **input Email** **or** **Phone**.
* **JPasswordField** for **input Password**.
* **JButton** for **Login**.

**Description**:

* If user clicks **Login** button, then **validate**:
  + **Email** **or** **Phone** field **must** **be** **filled**.
  + **Password** field **must** **be** **filled**.
  + (**Email** **or** **Phone**) and **Password** **must** be **correct** according to the **existing data in database**.
* If user has successfully signed in, display a **JOptionPane** along with the user’s name.



**Figure 2. Success Login Message**

* + After user close the message pane, program will **validate** either its role is “**User**” or “**Admin**”. Lastly, the **Login** **Form** will be closed, and **Main** **Form** will be displayed (Figure 1 or Figure 2 depends on user’s role).
* If user failed to login then display warning message.
* If user click “**Sign** **up** **here**.” label, then program will close **Login** **Form** and open **Register** **Form**.

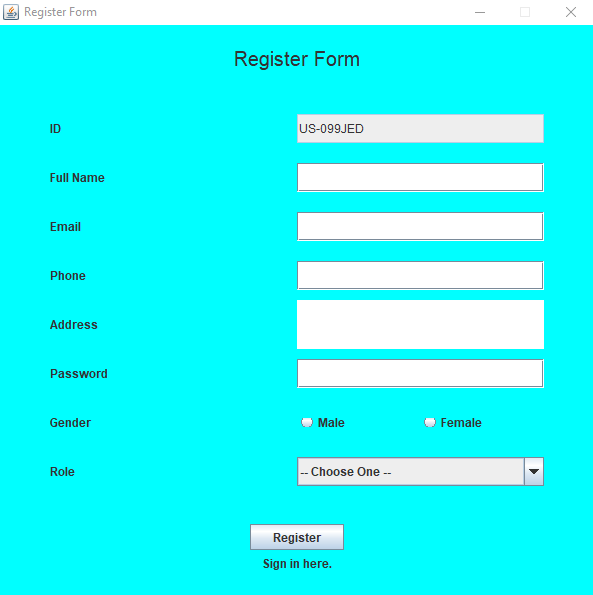


**Figure 3. Failed Login Warning**

(**additional** **note**: **every** **error** occurrence will be shown on error message **JOptionPane**).

1. **Register Form**

**Register Form** is used by users to **Register**.



**Figure 4. Register Form**

**Components**:

* **JLabel** for **Title**, **ID**, **Full Name**, **Email**, **Phone**, **Address**, **Password**, **Gender**, **Role**, and **Sign In here**.
* **JTextField** for **input ID** (**read**-**only**), **Full Name**, **Email**, and **Phone**.
* **JTextArea** for **input Address**.
* **JPasswordField** for **input Password**.
* **JRadioButton** for **input Gender** (“**Male**” or “**Female**”).
* **JComboBox** for choose **user Role** which has 3 options:
  + **– Choose One – (Default Option)**
  + **User**
  + **Admin**
* **JButton** for Register.

**Description**:

* If user clicks **Register** button, then **validate**:
  + **ID is generated** when Register Form is opened with **format**:

**US-XXXXXX**

Where **X**: Random Digit (0-9) or Uppercase Character (A-Z)

Example: US-A1R0W2, US-12UI9P, US-P9150H

* + **Full** **Name** must be between **5 - 30 characters**.
  + **Email** must **have not been taken** and be in valid **format**:

**[email]@[provider].[domain]**

* + - Character ‘@’ must **not be next to** ‘.’.
    - **Must not starts** and **ends** with ‘@’ nor ‘.’.
    - **Must contain** **exactly** **one** ‘@’.
    - **Must contain** **exactly** **one** ‘.’ after ‘@’ for separating [**provider**] and [**domain**].

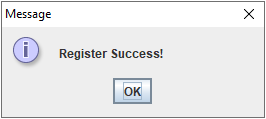


**Figure 5. Invalid Email Format Warning**

* + **Phone** must **have not been taken** and consist only **10 - 12 digits** only.
  + **Address** must consist of **10 or more characters** and **ends with** ‘ Street’.
  + **Password** must **8 - 20 length of character** and **digit** (must at least contain 1 character and 1 digit).
  + **Gender** must be selected either “**Male**” or “**Female**”.
  + **Role** must be chosen either “**User**” or “**Admin**”.

(**additional** **note**: **every** **error** occurrence will be shown on error message **JOptionPane**).

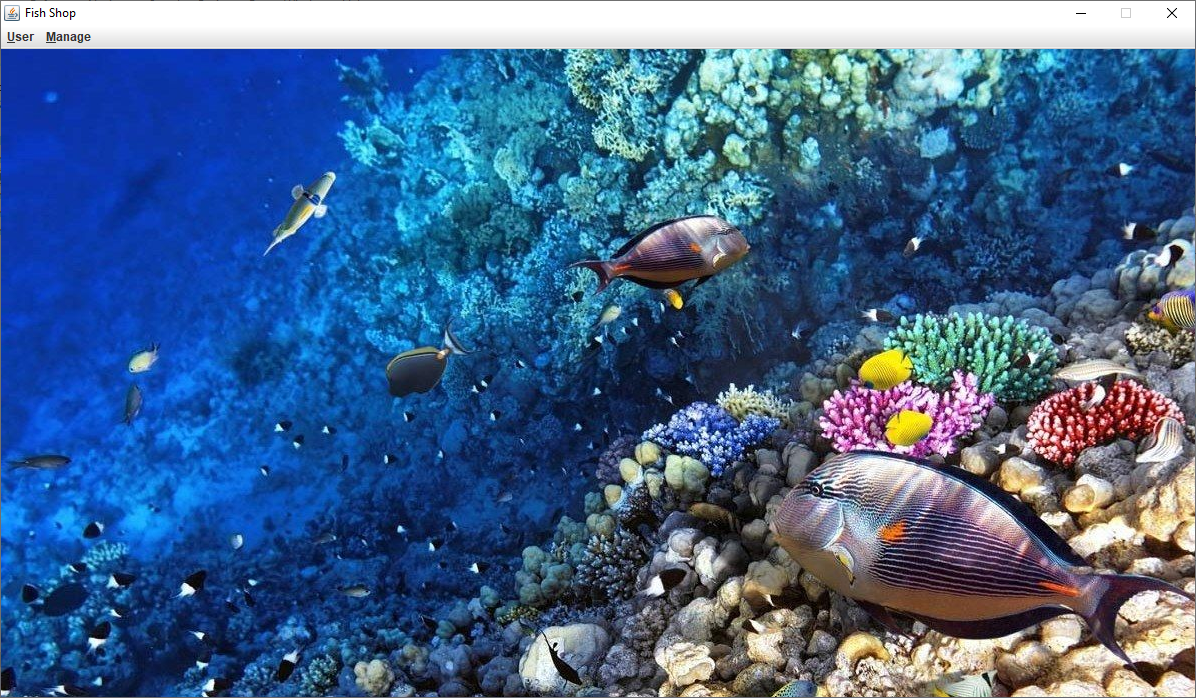
* If user successfully register, **insert** new **user** data **to database**, then program will display a success insert message. Lastly, program will close **Register** **Form** and open **Login** **Form**.



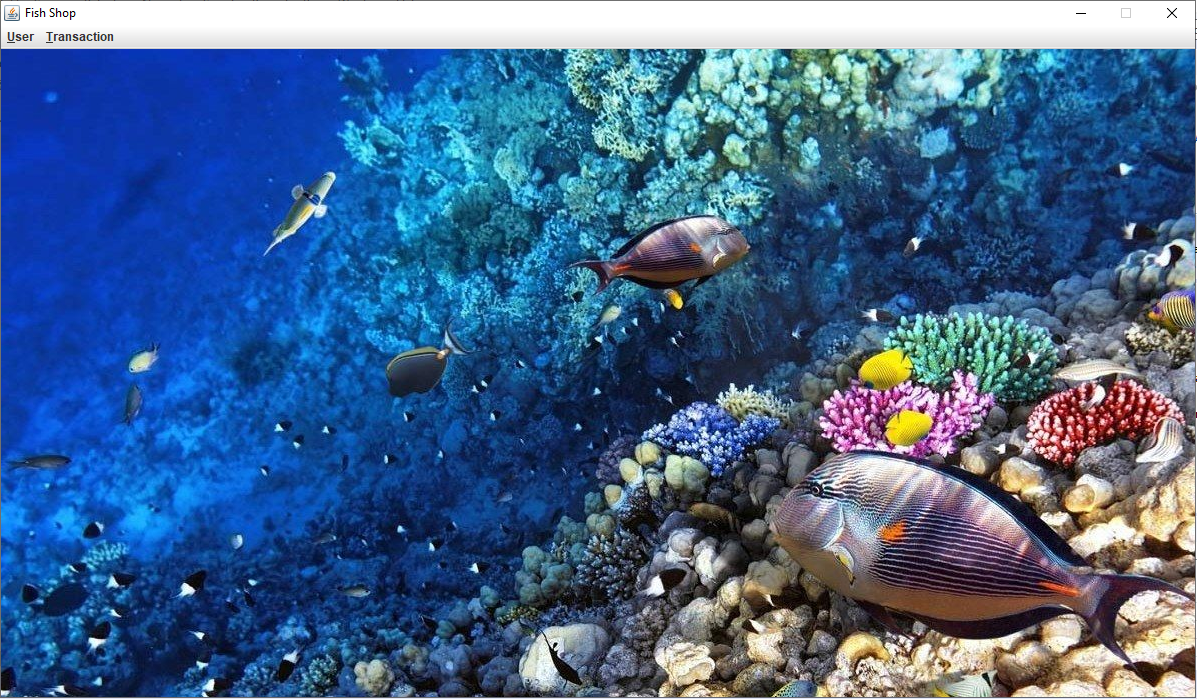
**Figure 6. Register Success Message**

1. **Main Form**

This is the main form of the program and will be opened if the user successfully logged in. In this form the user can select menu from the menu bar, such as **User** **Menu** (**Profile**, **Logoff**, and **Exit**), **Transaction** **Menu** (**Buy Fish** and **View Transaction History**) and **Manage** **Menu** (**Manage Fish**). The menu that will show is determined by the user role. If the user role is “**User**”, then the menus that available are **User** **Menu** and **Transaction** **Menu** only. If the user role is “**Admin**”, then the menus that available are **User** **Menu** and **Manage** **Menu**.



**Figure 6. User Main Form**



**Figure 7. Admin Main Form**

**Components**:

* **User** Role:
  + **JMenu** for **Menu User** and **Menu Transaction**.
  + **JMenuItem** for **Profile**, **Logoff**, **Exit**, **Buy Fish**, and **View Transaction History**.
* **Admin** Role:
  + **JMenu** for **Menu User** and **Menu Manage**.
  + **JMenuItem** for **Profile**, **Logoff**, **Exit**, and **Manage** **Fish**.

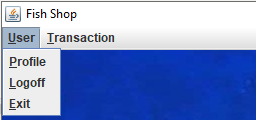
**Description**:

There are some **JMenuItem** that available (**to be clicked**):

* **User** Role

There will be two **JMenu** displayed:

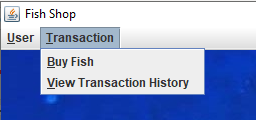
* + - * **User**



**Figure 8. User Menu**

There are 3 **JMenuItem** available **(to be clicked**) in **User Menu** (**Profile**, **Logoff**, and **Exit**).

* + - * + If user clicks **Profile**, program will open **Profile** **Form** inside the **Main** **Form**.
        + If user clicks **Logoff**, program will close **Main** **Form** and open **Login** **Form**.
        + If user clicks **Exit**, program will be **terminated**.
      * **Transaction**



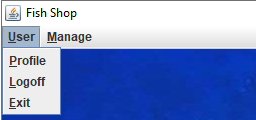
**Figure 9. Transaction Menu**

There are 2 **JMenuItem** available (**to be clicked**) in **Transaction Menu** (**Buy Fish** and **View Transactioin History**).

* + - * + If user click **Buy Fish**, program will open **Buy Fish Form** inside the **Main Form**.
        + If user click **View Transaction History**, program will open **Transaction Form** inside the **Main** **Form**.
* **Admin** Role

There will be two **JMenu** displayed:

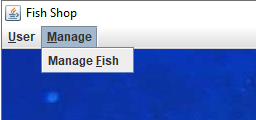
* + - * **User**



**Figure 10. User Form**

There are 3 **JMenuItem** available (**to be clicked**) in **User** **Menu** (**Profile**, **Logoff**, and **Exit**).

* + - * + If user clicks **Profile**, program will open Profile Form inside the Main Form.
        + If user clicks **Logout**, program will close Main Form and open Login Form.
        + If user clicks **Exit**, program will be terminated.
      * **Manage**



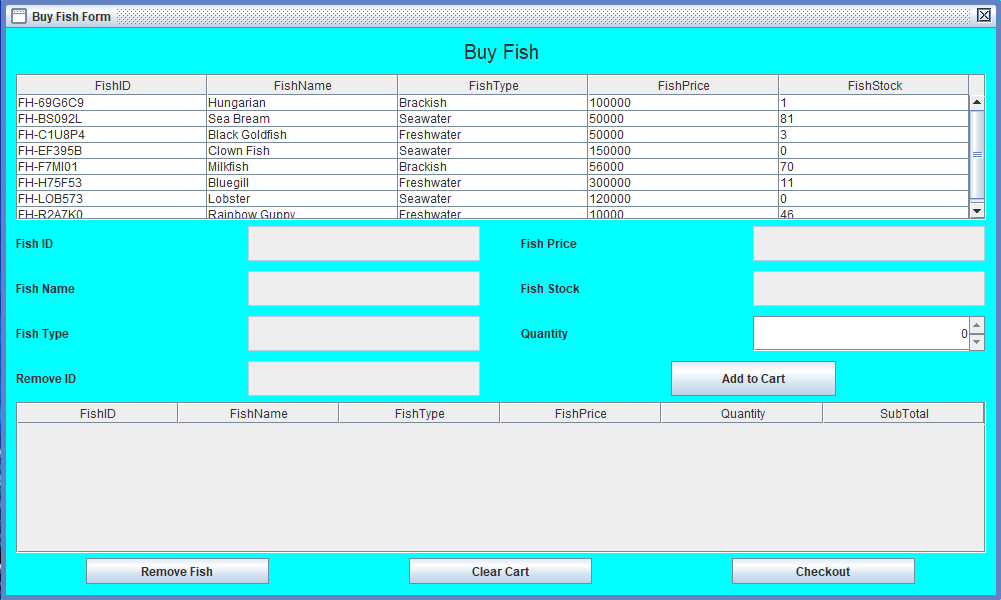
**Figure 11. Manage Menu**

There are 1 **JMenuItem** available (**to be clicked**) in **Manage** **Menu** (**Manage** **Fish**).

* + - * + If user clicks **Manage** **Fish**, program will open **Manage** **Fish** **Form** inside the **Main** **Form**.

1. **Buy Fish Form**

This form **displays all the currently available fish**. In this form, user can click any product to see the fish detail and add it to cart. But if the Fish Stock is 0, so the fish detail will not be shown in the form and cannot be added to cart. In this form, cart is also shown at the bottom view of the form. This form is made with **JInternalFrame**.



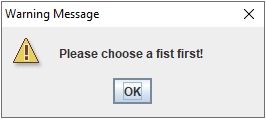
**Figure 12. Buy Fish Form**

**Components**:

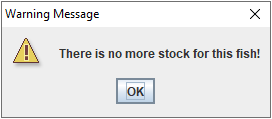
* **JTable** for **Fish Table** and **Cart Table**.
* **JScrollPane** for **Fish** and **Cart Table Container**.
* **JLabel** for **Title**, **Fish ID**, **Fish** **Name**, **Fish** **Type**, **Fish** **Price**, **Fish** **Stock**, **Quantity**, and **Remove** **ID**.
* **JTextField** (**read**-**only**) for **Fish ID**, **Fish** **Name**, **Fish** **Type**, **Fish** **Price**, **Fish** **Stock**, **Quantity**, and **Remove** **ID**.
* **JSpinner** for **input Quantity**.
* **JButton** for **Add to Cart**, **Remove Fish**, **Clear Cart**, and **Checkout**.

**Description**:

User can **only edit quantity field** and the **other fields cannot be edited**. If user clicks **Add to Cart** button, program will **validate** is there any **fish chosen** or not. If the fish already chosen, then program will **validate** the **quantity** that **cannot be less than 1** and **cannot be more than its Fish Stock**. If user choose a fish that currently **stock is 0**, then program will **show warning message**.



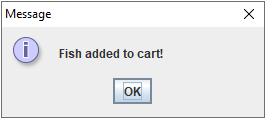
**Figure 13. Choose Fish Warning**



**Figure 14. No Stock Warning**

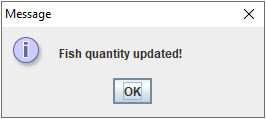
If all conditions are fulfilled, then the fish will be added to the user’s cart:

* If the fish **has not existed yet**, **insert** new **fish** to user’s cart and show message.



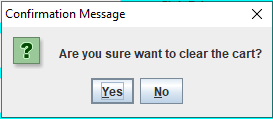
**Figure 15. Fish Added Message**

* If the fish **has existed**, **update** fish **quantity** **with the addition of quantity** and show message.



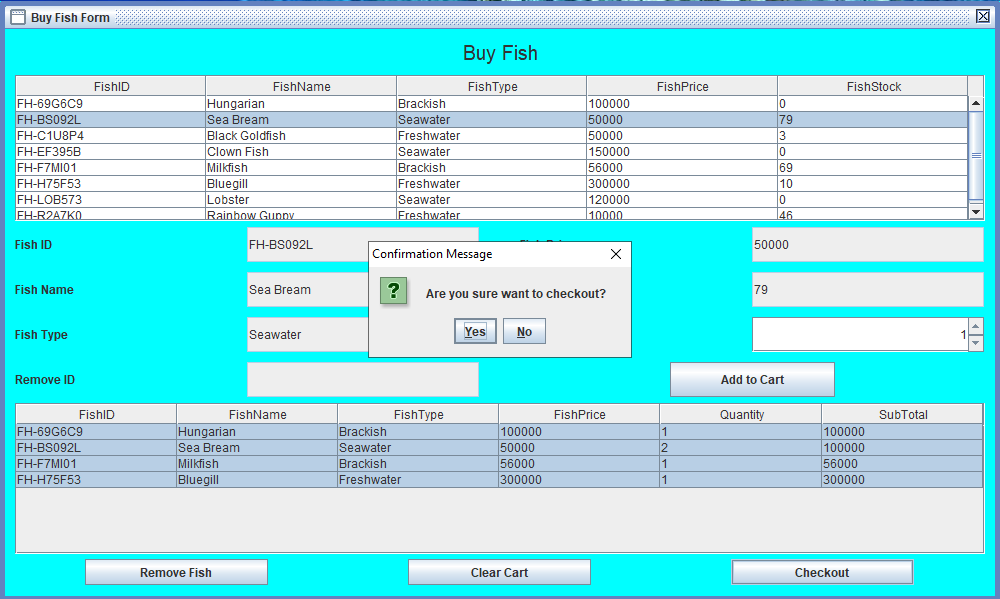
**Figure 16. Fish Quantity Added Message**

* After the **fish added to cart**, **decrease Fish Stock** equals to the quantity.
* User can also remove fish form cart by using **Remove Fish** button. **Validate** that the **fish** from cart **must be chosen** first by click the fish row that want to be deleted. To see the fish that want to deleted is already clicked or chosen, Remove ID will be contains the selected Fish ID that want to be deleted. After remove fish from cart, shows message.
* After the fish **removed from cart**, **increase Fish Stock** equals to the quantity.
* If user clicks **Clear Cart** button, program will **ask** **confirmation** first and if user choose **yes**, then program will **clear the cart** and **add each fish quantity** to **Fish** **Stock**.

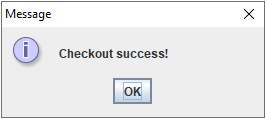


**Figure 17. Clear Cart Confirmation**

* If user clicks **Checkout** button, program will **ask** **confirmation** first and if user choose **yes**, then program will **clear the cart** and **insert all fish** to **Transaction** (**Header** and **Detail**) and shows message.



**Figure 19. Checkout Confirmation**

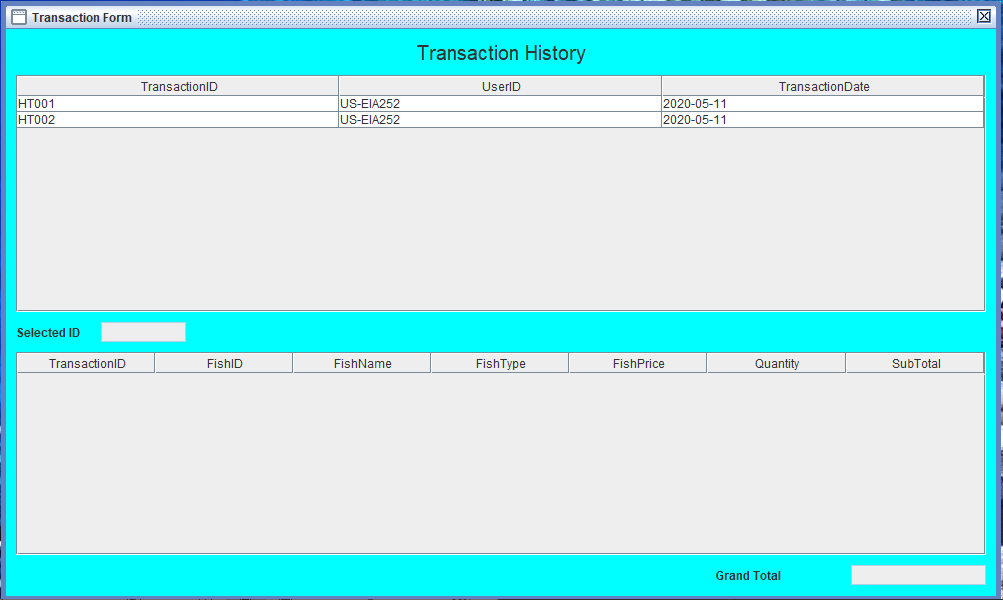


**Figure 20. Checkout Success Message**

(**additional** **note**: **every** **error** occurrence will be shown on error message **JOptionPane**).

1. **Transaction Form**

**Transaction** **Form** allows user to **view all the transaction history** or all transaction that user has made. This form is made with **JInternalFrame**.



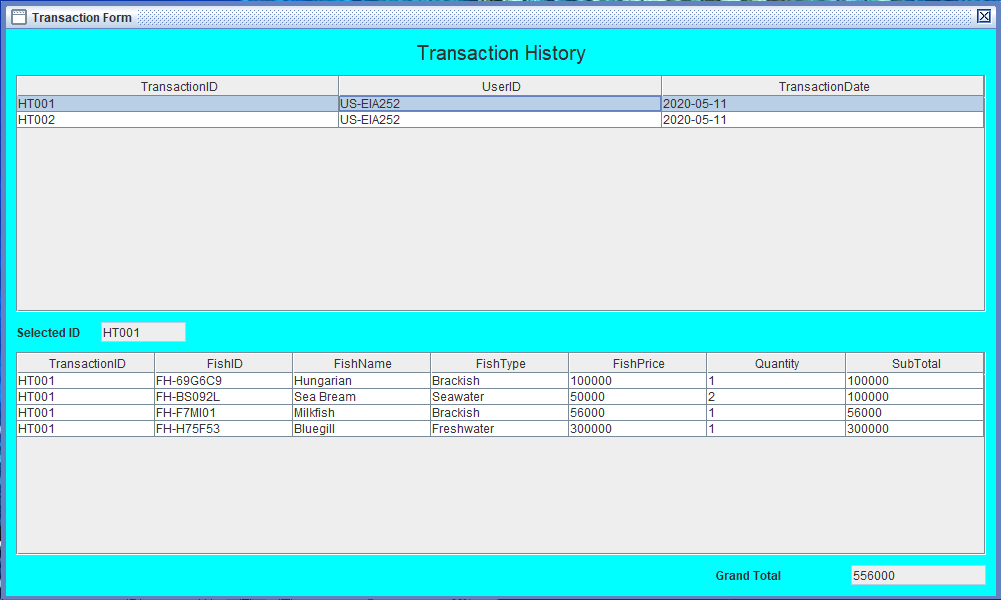
**Figure 21. Transaction Form**

**Components**:

* **JLabel** for **Title**, **Selected ID**, **Grand Total**.
* **JTable** for **Header Transaction Table** and **Detail Transaction Table**.
* **JScrollPane** for **Header** and **Detail** **Transaction** **Table** **container**.
* **JTextField** (**read**-**only**) for **Selected ID** and **Grand Total**.

**Description**:

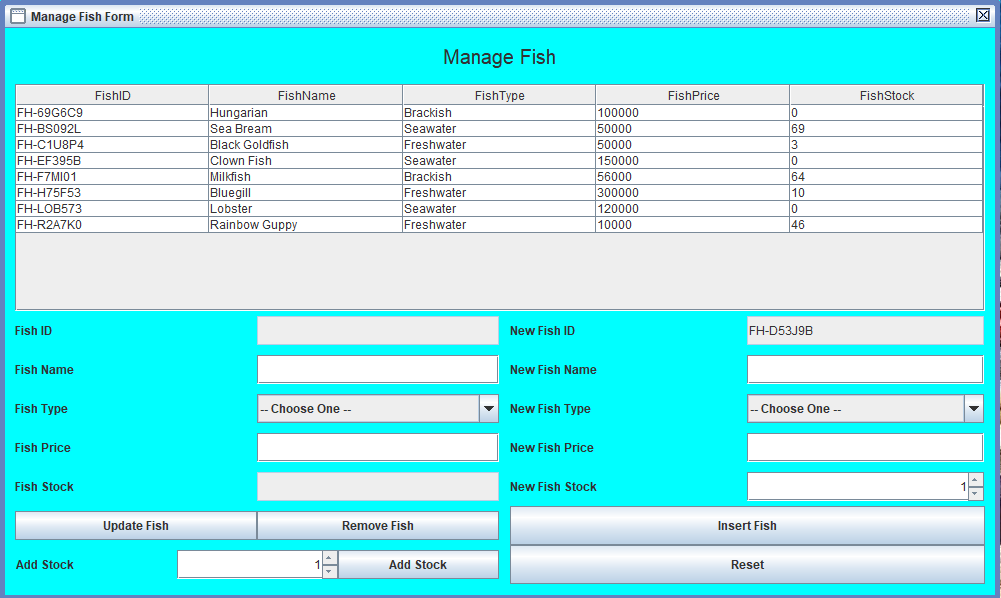
If user **clicks** one of the **header transaction table**, it will **show** all the **detail** **transaction** that user bought on that transaction. It also **triggers** program to **update** the **Selected** ID based on user clicks and **Grand** **Total** by **sum** all the **Sub** **Total** of the detail transaction. **Sub** **Total** in detail transaction is obtained from **calculation** between **Quantity** times **Fish** **Price**.



**Figure 22. Detail Transaction View**

1. **Manage Fish Form**

**Manage** **Fish** **Form** is made for admin to manage fish. Admin can do **insert** new fish, **update** any existing fish data, **remove**/**delete** fish, and **add** **fish** **stock**. This form is made with **JInternalFrame**.



**Figure 23. Manage Fish Form**

**Components**:

* **JLabel** for **Title**, **Fish ID**, **Fish Name**, **Fish Type**, **Fish Price**, **Fish Stock**, **Add Stock**, **New Fish ID**, **New Fish Name**, **New Fish Type**, **New Fish Price**, and **New Fish Stock**.
* **JTable** for **Fish Table**.
* **JScrollPane** for **Fish Table container**.
* **JTextField** for **input Fish ID** (**read**-**only**), **Fish Name**, **Fish Price**, **Fish Stock** (**read**-**only**), **New Fish ID** (**read**-**only**), **New Fish Name**, and **New Fish Price**.
* **JComboBox** for **choose Fish Type** and **New Fish Type**.
* **JSpinner** for **input Add Stock** and **New Fish Stock**.
* **JButton** for **Update Fish**, **Remove Fish**, **Add Stock**, **Insert Fish**, and **Reset**.

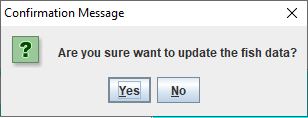
**Description**:

* All the fields can be edited **except** for Fish ID, New Fish ID, and Fish Stock.
* Admin can do **Insert**, **Update**, **Delete**, and **Add Stock** fish from database.
  + If admin clicks Update Fish button, then **validate**:
    - **Fish** **ID** is **not** **empty** (by click a row in fish table).



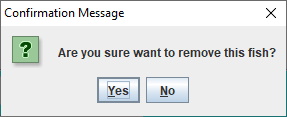
**Figure 24. Choose Fish Warning**

* + - **Fish** **Name** must consist of **3 - 25 characters**.
    - **Fish** **Type** must be **chosen** either **Freshwater**, **Seawater**, or **Brackish**.
    - **Fish** **Price** must between **1000 - 10000000**.
  + If all conditions for **Update** **Fish** are fulfilled, then program will **ask** **confirmation** first and if use choose **yes**, program will **update** **fish’s** **data** **in** **database**.



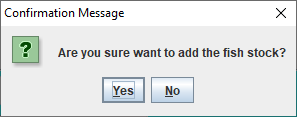
**Figure 25. Update Fish Confirmation**

* + If admin clicks **Remove** **Fish** button, then **validate**:
    - **Fish** **ID** is **not** **empty** (by click a row in fish table).
  + If all conditions for **Remove** **Fish** are fulfilled, then program will **ask** **confirmation** first and if use choose **yes**, program will **delete** **fish** **from** **database**.



**Figure 26. Remove Fish Confirmation**

* + If admin clicks **Add** **Stock** button, then **validate**:
    - **Fish** **ID** is not **empty** (by click a row in fish table).
    - **Add** **Stock** must between **1 -** **100**.
  + If all conditions for **Add** **Stock** are fulfilled, then program will **ask** **confirmation** first and if use choose **yes**, program will **update fish’s stock in database with the addition of Added Stock**.



**Figure 27. Add Fish Stock Confirmation**

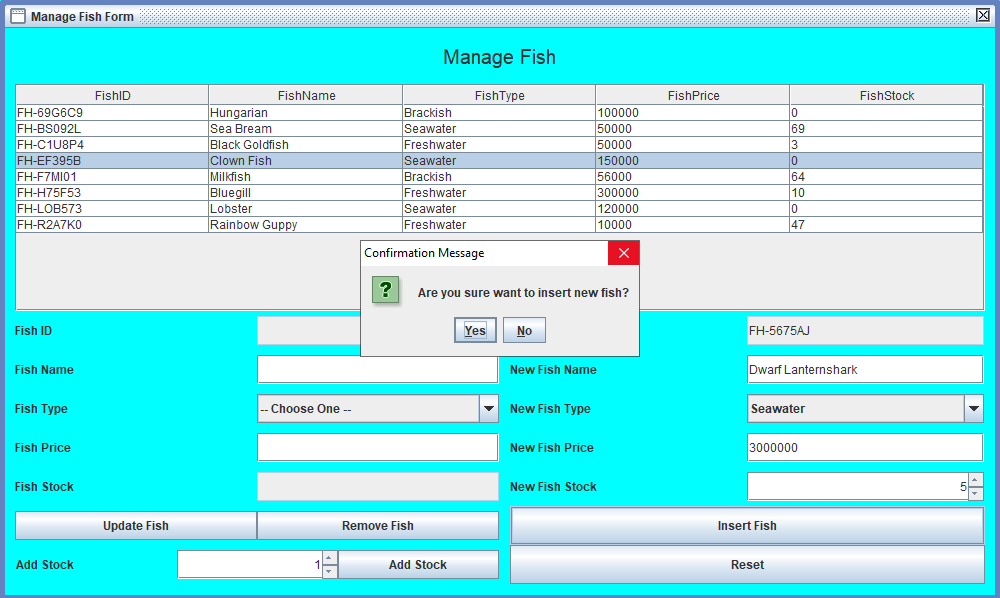
* + If admin clicks **Insert** **Fish** button, then **validate**:
    - **New Fish ID** is generated when **Manage** **Fish** **Form** open with format:

**FH-XXXXXX**

Where **X**: Random Digit (0-9) or Uppercase Character (A-Z)

Example: FH -A1R0W2, FH -12UI9P, FH-P9150H

* + - **New Fish Name** must consist of **3 - 25 characters**.
    - **New Fish Type** must be **chosen** either **Freshwater**, **Seawater**, or **Brackish**.
    - **New Fish Price** must between **1000 - 10000000**.
    - **New Fish Stock** must between **1 - 100**.
  + If all conditions for **Insert** **Fish** are fulfilled, then program will **ask** **confirmation** first and if use choose **yes**, program will **insert** **new** **fish** **to** **database**.



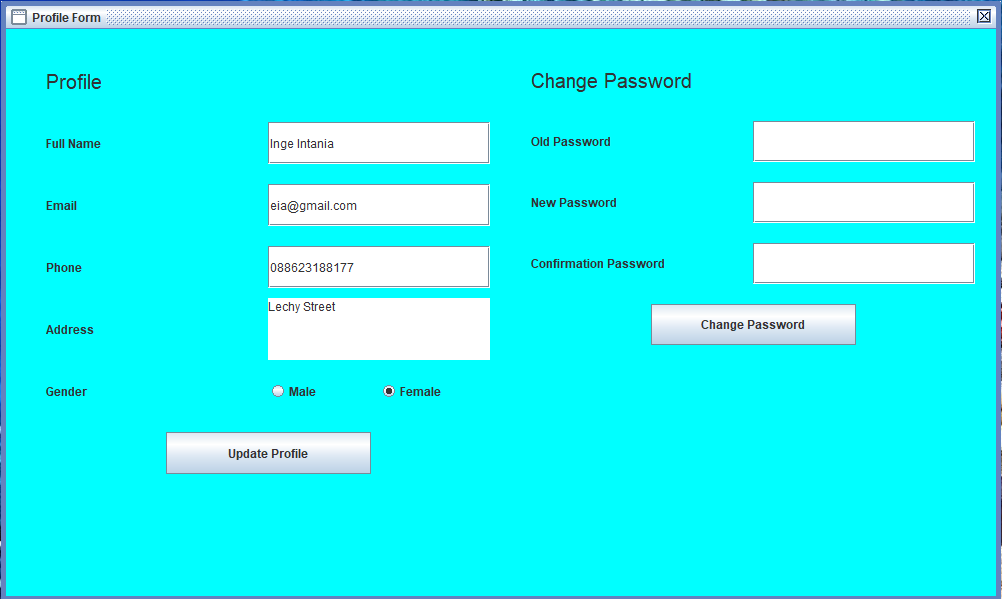
**Figure 28. Insert New Fish Confirmation**

* + If admin clicks **Reset** button, then **reset all field**.

(**additional** **note**: **every** error occurrence will be shown on error message **JOptionPane**).

1. **Profile Form**

**Profile** **Form** allows users to see and update their profile. This form is made with **JInternalFrame**.



**Figure 29. Profile Form**

**Components**:

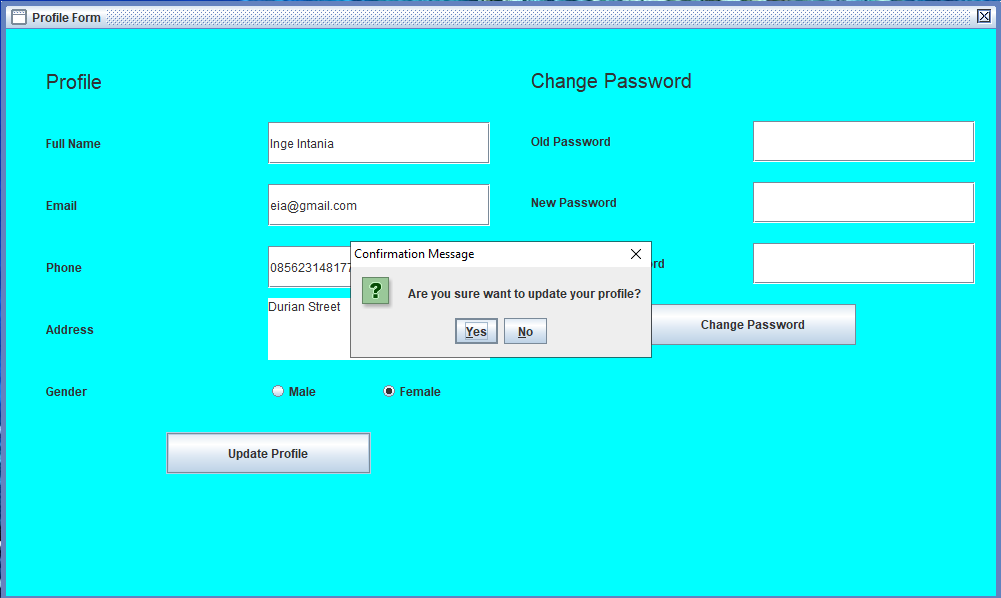
* **JLabel** for **Profile Title**, **Full Name**, **Email**, **Phone**, **Address**, **Gender**, **Change Password Title**, **Old Password**, **New Password**, and **Confirmation Password**.
* **JTextField** for **input Full Name**, **Email**, and **Phone**.
* **JTextArea** for **input Address**.
* **JRadioButton** for **choose** **Gender** (“**Male**” or “**Female**”).
* **JPasswordField** for **input Old Password**, **New Password**, and **Confirmation Password**.
* **JButton** for **Update Profile** and **Change Password**.

**Description**:

* If user clicks **Update** **Profile** button, program will **validate**:
  + **Full** **Name** must be between **5 - 30 characters**.
  + **Email** must **have not been taken from other account** and be in valid **format**:

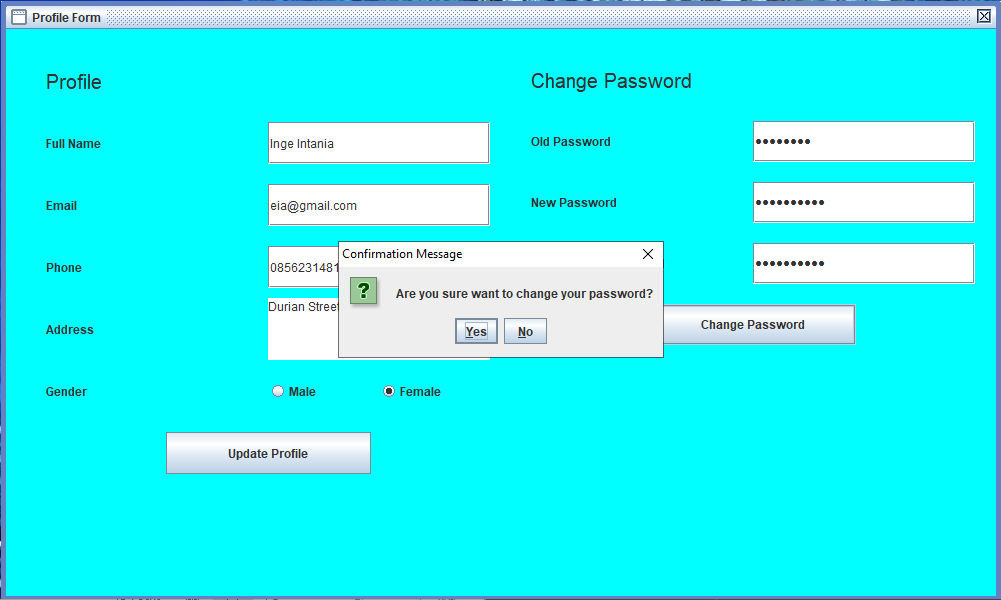
**[email]@[provider].[domain]**

* + - Character ‘@’ **must not be next to** ‘.’.
    - **Must not starts** and **ends** with ‘@’ nor ‘.’.
    - **Must contain exactly one** ‘@’.
    - **Must contain exactly one** ‘.’ after ‘@’ for separating [**provider**] and [**domain**].
  + **Phone** must **have not been taken from other account** and consist only **10 - 12 digits** only.
  + **Address** must consist of **10 or more characters** and **ends** **with** ‘ Street’.
  + **Gender** must be **selected** **either** “**Male**” or “**Female**”.
* If all conditions for **Update** **Profile** are fulfilled, then program will **ask** **confirmation** first and if user choose **yes**, program will **update** **user’s** **data** **in** **database**.



**Figure 30. Update Profile Confirmation**

* If user clicks **Change** **Password** button, program will **validate**:
  + **Old** **Password** must **match** with **user** **current** **password**.
  + **New** **Password** must **8 - 20 length of character** and **digit** (must at least contain 1 character and 1 digit).
  + **Confirmation** **Password** must **match** with **New** **Password**.
* If all conditions for **Change** **Password** are fulfilled, then program will **ask** **confirmation** first and if user choose **yes**, program will **update user’s password in database**.



**Figure 31. Change Password Confirmation**

(**additional** **note**: **every** **error** occurrence will be shown on error message **JOptionPane**).

Here are the rules that you must follow to create your project:

1. Use appropriate software for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya.
2. Additional rules:
   * All connections and data manipulation to the database must use the JDBC driver. The database file (.sql) must be submitted as well.
   * Use **LayoutManager** to design your application (You are **not allowed** to use **free** / **absolute** / **null** layout).
   * All validations must use the techniques taught during practicum. (Another language or Regular Expression is prohibited).
3. Collect appropriate files for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya.
4. Include the other files that can support your project, such as:
   * All files in your project.
   * Other files (image, audio, video, etc.) used in your project.
   * \*.DOC file (documentation of your project) that contains all pages in your project, reference links of additional files (image, audio, video, etc.) used in your project, the description about how to use your application, etc.

**References**:

* <https://id.pinterest.com/pin/364087951127927864/>

**Please ask your teaching assistant if there are any project related questions.**