**Janji Jywa**

**Janji Jywa** is a shop that sells many type of beverage. To manage all the transactions and manage system of the beverage, Janji Jywa’s want to create the program for the store. The program is made using **Java Programming Language**.

There are some requirements to build this application:

Diagram

Description automatically generated

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**.

Graphical user interface, website

Description automatically generated

Figure 1. Login Form

**Components**:

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

**Constraint**:

* If user clicks **Login** button:
  + **Email** field **must** **be** **filled**.
  + **Password** field **must** **be** **filled**.
  + **Email** and **Password** **must** be **correct** according to the **existing data in database**.
* If user has successfully signed in, **Login Form** will be closed and **Main Form** will be displayed.
* 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**.

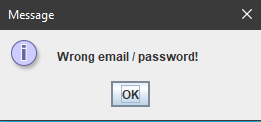


Figure 2. Failed Login Warning

1. **Register Form**

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

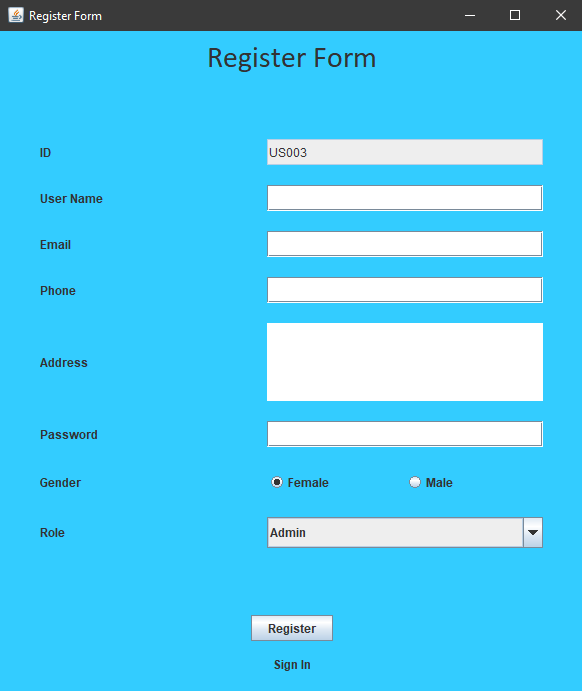


Figure 3. Register Form

**Components**:

* **JLabel** for Title, ID, Username, Email, Phone, Address, Password, Gender, Role, and Sign In.
* **JTextField** for input ID (**read**-**only**), Username, 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 2 options:
  + **Admin**
  + **Customer**
* **JButton** for Register.

**Constraint**:

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

**USXXX**

Where **X**: Increment digit

Example: US001, US002, US003

* + **User**name must be between **5 - 30 characters**.
  + **Email** must 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**].

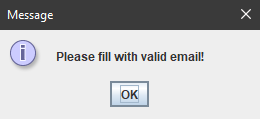


Figure 4. Invalid Email Format Warning

* + **Phone** number must be numeric and more than equals **12 digits**.
  + **Address** must consist of **10 or more characters** and **ends with** ‘ Street’.
  + **Password** must **5 - 30 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 “**Admin**” or “**Customer**”.
* 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**.

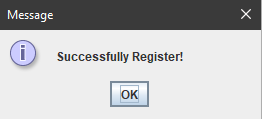


Figure 5. Register Success Message

1. **Main Form**

This is the main form for the program. At this form the user can select menu from the menu bar, such as **User** **Menu** (**Profile**, **Logoff**, and **Exit**), **Transaction** **Menu** (**Buy Beverage** and **View Transaction History**) and **Manage** **Menu** (**Manage Beverage**). The menu that will show is determined by the user role. If the user role is “**Customer**”, 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**.

Chart, text, treemap chart

Description automatically generated

Figure 1. Admin Main Form

Chart, text, treemap chart

Description automatically generated

Figure 2. Customer Main Form

**Components**:

* **User** Role:
  + **JMenu** for Menu Profile and Menu Transaction.
  + **JMenuItem** for Edit Profile, Log Off, Exit, Buy Beverage, and View Transaction History.
* **Admin** Role:
  + **JMenu** for Menu Profile and Menu Manage.
  + **JMenuItem** for Edit Profile, Log Off, Exit and Manage Beverage.

**Constraint**:

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

* **Both** Role
  + If user clicks **Edit Profile** Menu Item, then the program will open **Profile Form**.
  + If user clicks **Log Off** Menu Item, then the program will close **Main Form** and open **Login Form**.
  + If user clicks **Exit** Menu Item, then the program will be **terminated**.
* **Customer** Role

There will be two **JMenu** displayed:

* + - * **Profile**

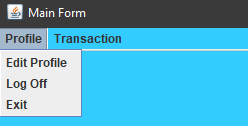


Figure 3. Customer: Profile Menu

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

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

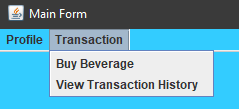


Figure 4. Customer: Transaction Menu

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

* + - * + If user click **Buy Beverage**, program will open **Buy Beverage 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:

* + - * **Profile**

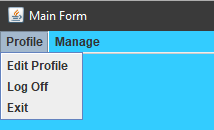


Figure 5. Admin: User Menu

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

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

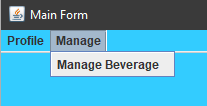


Figure 6. Admin: Manage Menu

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

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

1. **Buy Beverage Form**

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

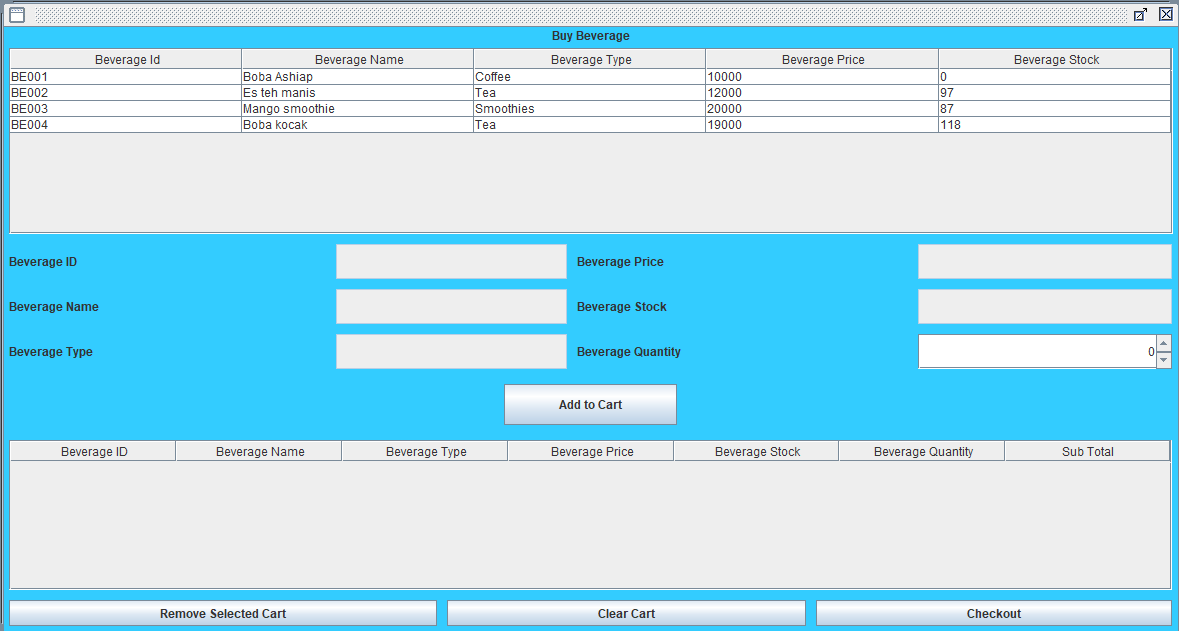


Figure 13. Buy Beverage Form

**Components**:

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

**Description**:

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

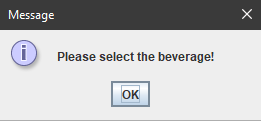


Figure 14. Choose Beverage Message

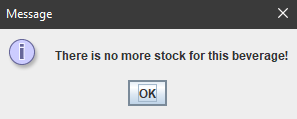


Figure 15. No Stock Message

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

* If the beverage **hasn’t existed yet**, **insert** new **beverage** to user’s cart and show message.

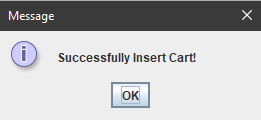


Figure 16. Success Insert Cart Message

* If the beverage **has existed**, **update** beverage **quantity** **with the addition of quantity**
* User can also remove beverage form cart by using **Remove Selected Cart** button. **Validate** that the **beverage** from cart **must be chosen** first by click the beverage row that want to be deleted. After remove beverage from cart, shows message.
* 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 beverage quantity** to **Beverage** **Stock**.

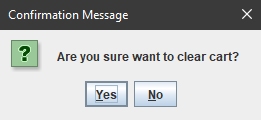


Figure 18. Clear Cart Confirmation

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

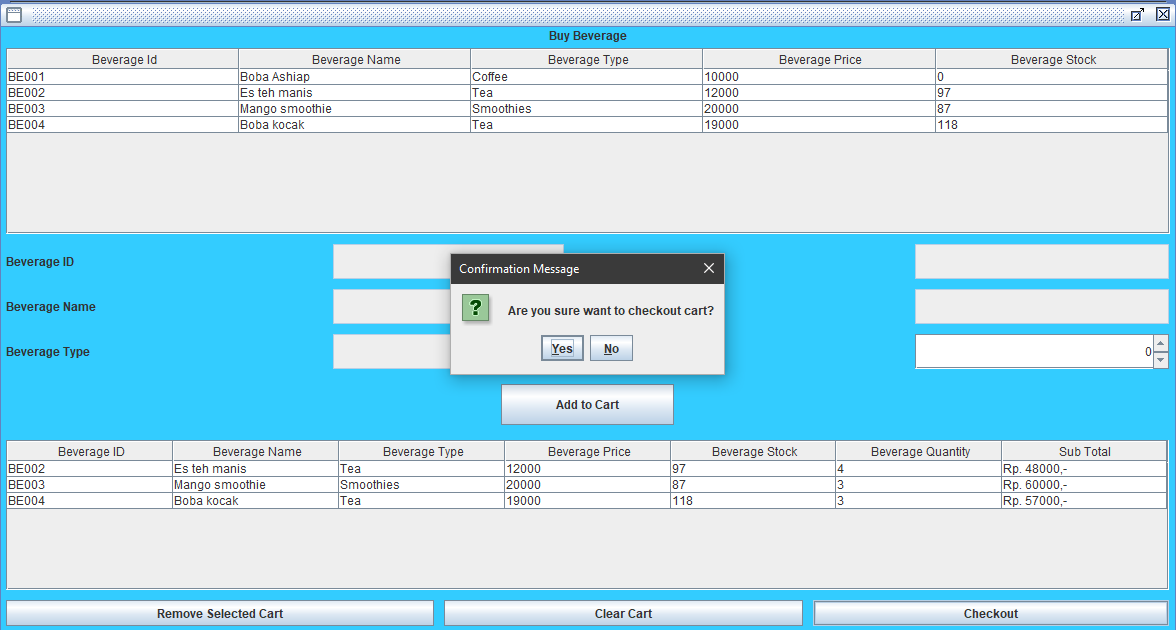


Figure 19. Checkout Confirmation

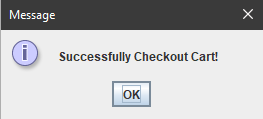


Figure 20. Checkout Success Message

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**.

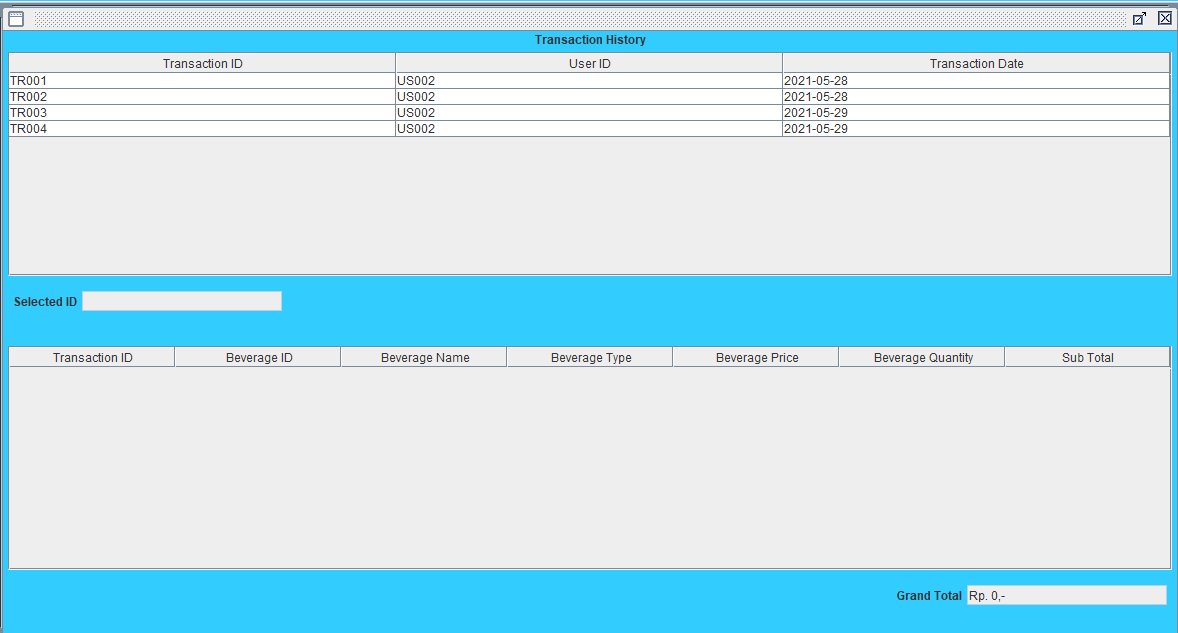


Figure 21. Transaction Form

**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.

**Constraint**:

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 **Beverage** **Price**.

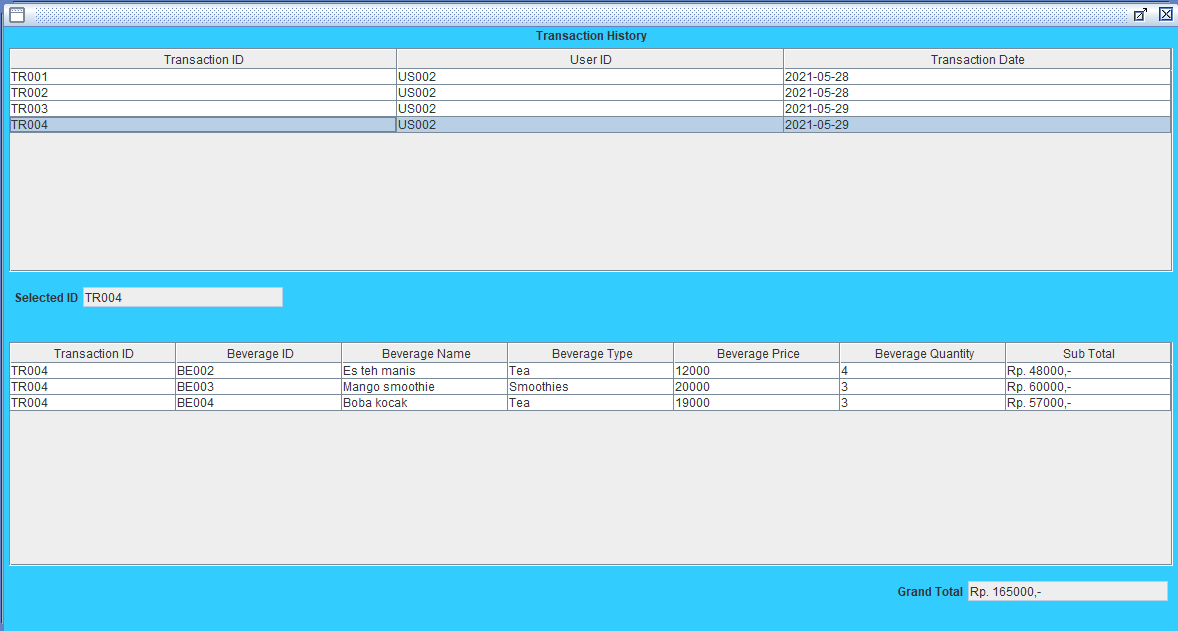


Figure 22. Detail Transaction View

1. **Manage Beverage Form**

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

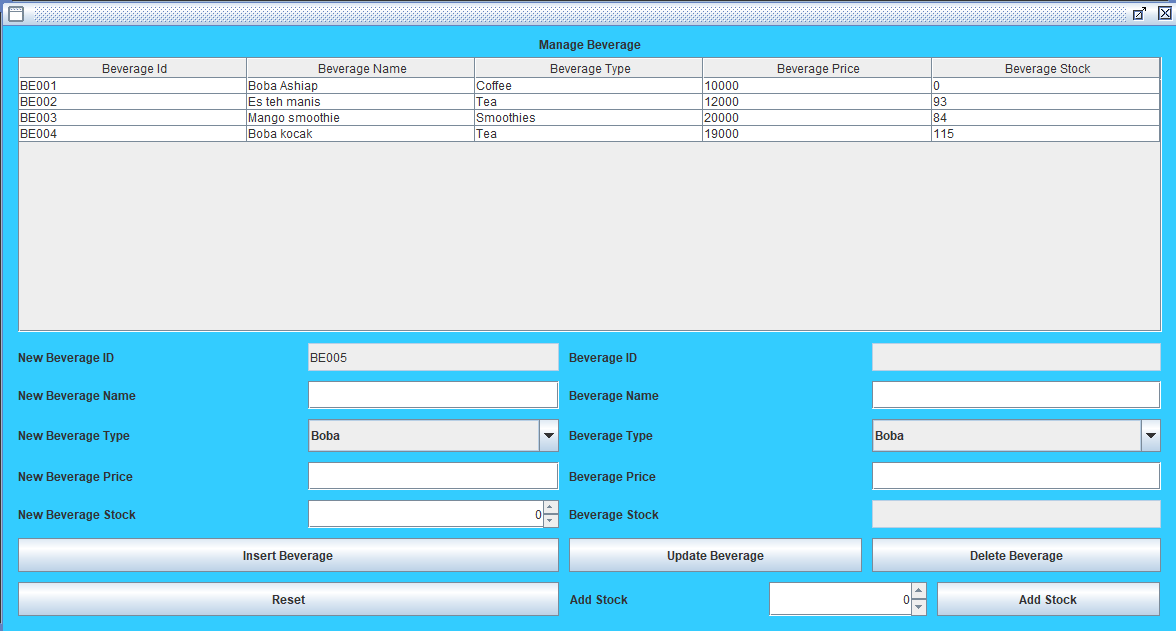


Figure 23. Manage Beverage Form

**Components**:

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

**Constraint**:

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

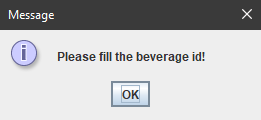


Figure 24. Choose Beverage Message

* + - **Beverage** **Name** must consist of **5 - 30 characters**.
    - **Beverage** **Type** must be **chosen** either **Boba**, **Coffee**, **Tea** or **Smoothies**.
    - **Beverage Price** must more than **0 and numeric**.
  + If all conditions for **Update** **Beverage** are fulfilled, then program will **ask** **confirmation** first and if use choose **yes**, program will **update** **beverage’s** **data** **in** **database**.

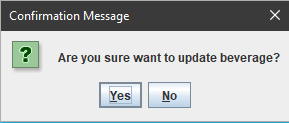


Figure 25. Update Beverage Confirmation

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

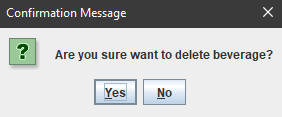


Figure 26. Remove Beverage Confirmation

* + If admin clicks **Add** **Stock** button:
    - **Beverage** **ID** is not **empty** (by click a row in beverage table).
    - **Add** **Stock** must be more than **0**.
  + If all conditions for **Add** **Stock** are fulfilled, then program will **ask** **confirmation** first and if use choose **yes**, program will **update beverage’s stock in database with the addition of Added Stock**.

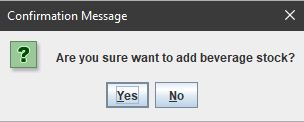


Figure 27. Add Beverage Stock Confirmation

* + If admin clicks **Insert** **Beverage** button:
    - **New Beverage ID** is generated when **Manage** **Beverage** **Form** open with format:

**BEXXX**

Where **X**: Increment digit

Example: BE001, BE002, BE003

* + - **New Beverage Name** must consist of **5 - 30 characters**.
    - **New Beverage Type** must be **chosen** either **Boba**, **Coffee**, **Tea** or **Smoothies**.
    - **New Beverage Price** must be more than **0 and numeric**.
    - **New Beverage Stock** must be more than **0 and numeric**.
  + If all conditions for **Insert** **Beverage** are fulfilled, then program will **ask** **confirmation** first and if use choose **yes**, program will **insert** **new** **beverage** **to** **database**.

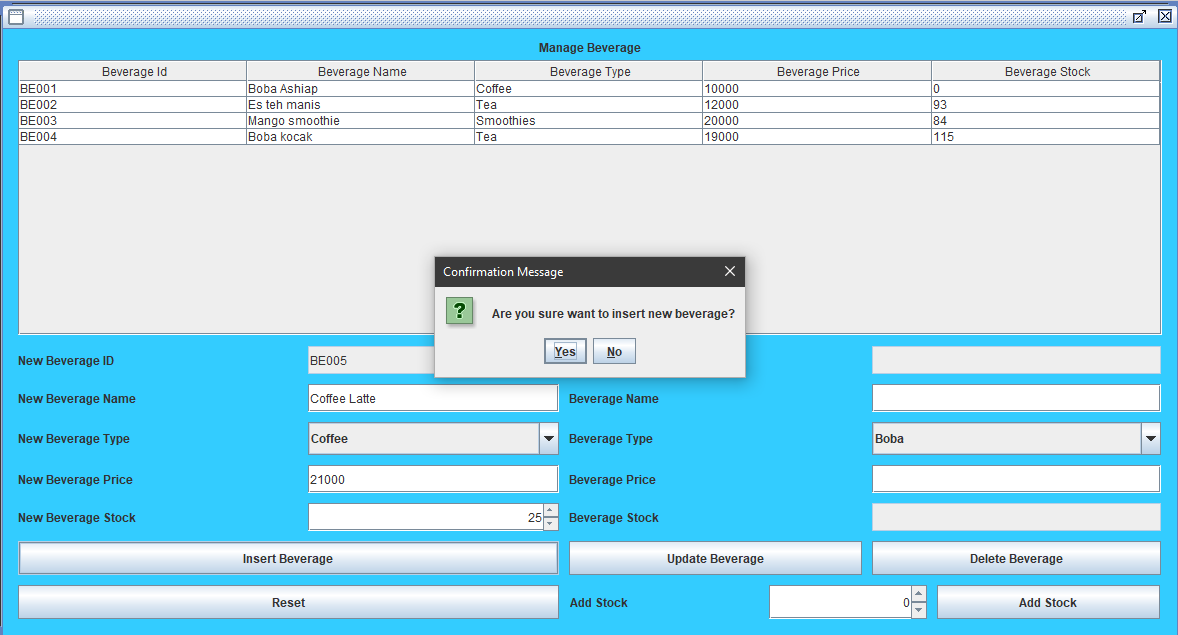


Figure 28. Insert New Beverage Confirmation

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

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

1. **Edit Profile Form**

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

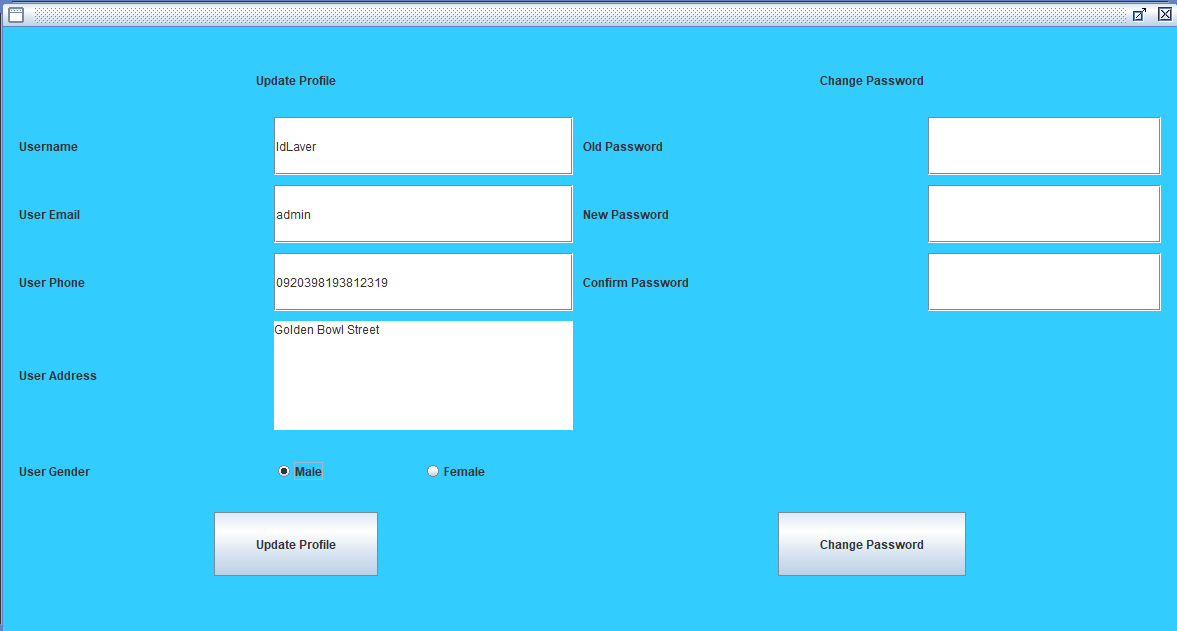


Figure 29. Edit Profile Form

**Components**:

* **JLabel** for Update Profile Title, Username, User Email, User Phone, User Address, User Gender, Change Password Title, Old Password, New Password, and Confirmation Password.
* **JTextField** for input Username, User Email, and User Phone.
* **JTextArea** for input User Address.
* **JRadioButton** for choose User 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**:
  + **Username** must be between **5 - 30 characters**.
  + **Email** must 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** number must be **numeric** and more than equals **12 digits**.
  + **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**.

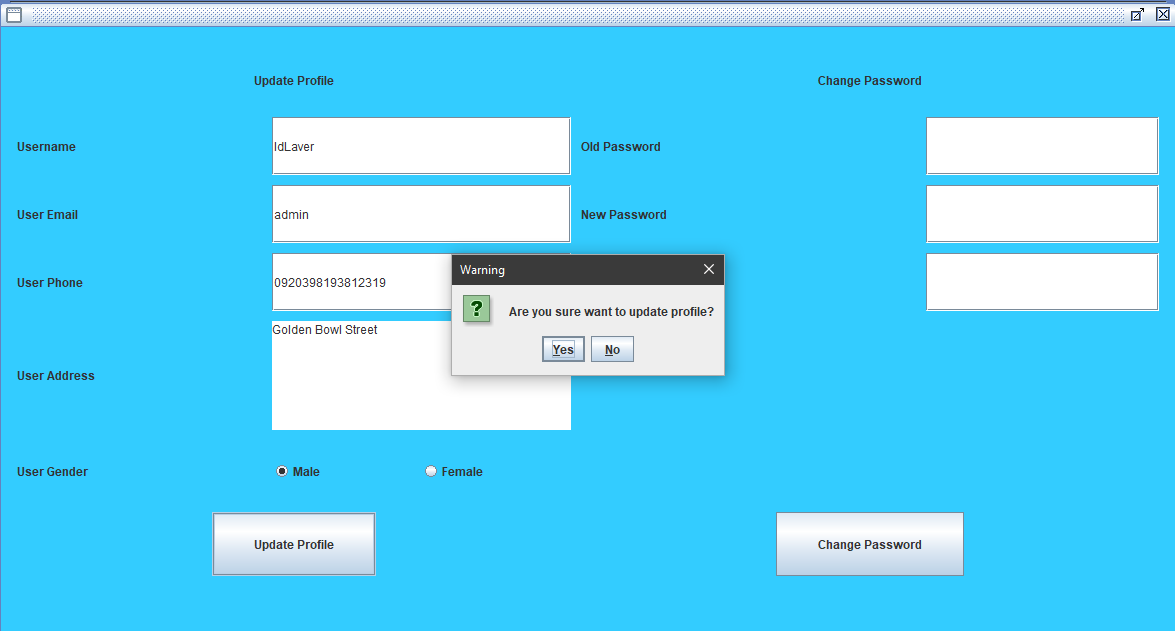


Figure 30. Update Profile Confirmation

* If user clicks **Change** **Password** button, program will **validate**:
  + **Old** **Password** must **match** with **user** **current** **password**.
  + **New** **Password** must **5 - 30 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**.

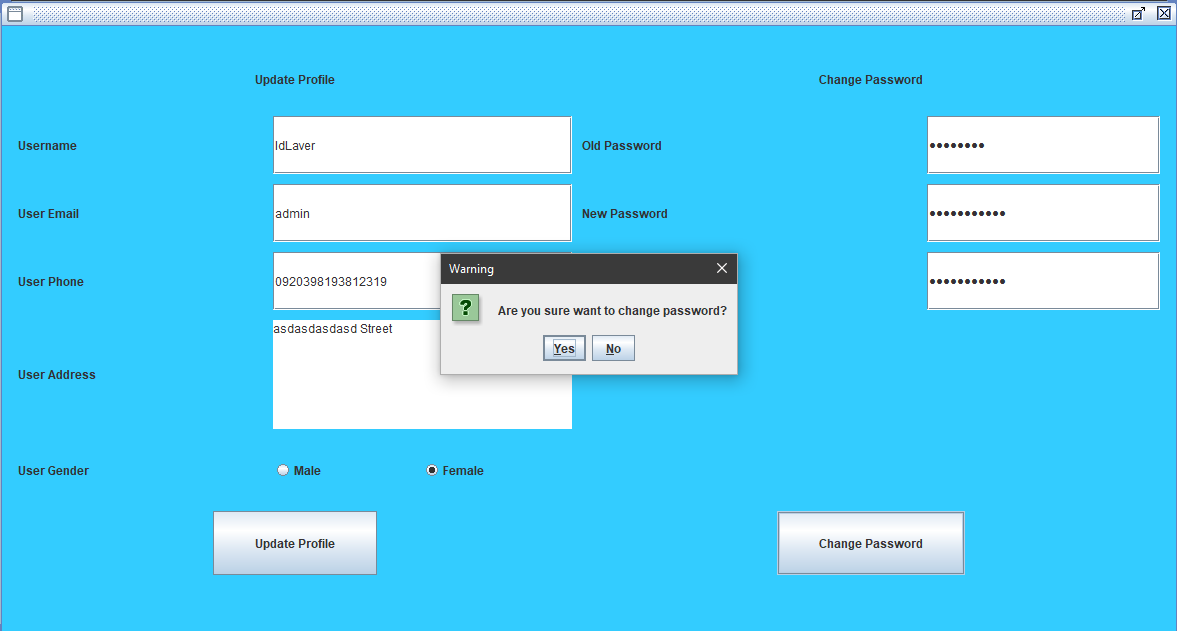


Figure 31. Change Password Confirmation