|  |  |
| --- | --- |
| **Project Case** |  |
| ISYS6197 | ISYS6197003  Business Application Development |
| **Computer Science** | **O242-ISYS6197-SR02-00** |
| ***Valid on*** *Odd Semester Year 2023/2024* | **Revision 00** |

1. Kelompok tidak diperkenankan untuk:

*Members of the group are prohibited from:*

* + - Melihat sebagian atau seluruh jawaban kelompok lain,

*Seeing a part or the whole answer from other groups,*

* + - Menyadur sebagian atau seluruh jawaban dari buku, catatan, video, dan jenis referensi lainnya,

*Retell a part or the whole answer from books, notes, videos, and other references,*

* + - Menyadur sebagian atau seluruh jawaban dari internet,

*Retell a part or the whole answer from the internet,*

* + - Mengumpulkan jawaban yang tidak sesuai dengan tema soal,

*Submitting an answer with a different theme from the given case,*

* + - Melakukan tindakan yang menyebabkan jawaban dicontek oleh orang lain atau kelompok lain, baik disengaja maupun tidak disengaja,

*Doing action that could result the answer being copied by someone or other groups, intentionally or unintentionally,*

* + - Melakukan tindakan kecurangan lainnya.

*Committing other dishonest actions.*

1. Jika kelompok terbukti melakukan tindakan seperti yang dicantumkan pada butir ke-1, maka nilai mahasiswa dan/atau kelompok yang melakukan kecurangan, baik menyontek atau dicontek, akan dinolkan sesuai dengan peraturan yang berlaku.

*If it has been proven that a group has committed dishonest actions outlined in point 1 above, the whole groups related to the incident, regardless of which one copies or has their answer copied, will be issued a score of zero according to the regulation.*

1. Jawaban yang dapat diterima dan dinilai adalah jawaban yang dikumpulkan sebelum batas waktu yang telah ditentukan.

*The answer must be submitted before the designated deadline to be accepted and graded,*

1. Jawaban akan dinilai berdasarkan teknik atau metode yang diajarkan pada kelas praktikum dengan menggunakan software yang sudah ditentukan.

*The scoring will be based on the materials taught during the practicum classes using the designated software. Using different software than requested may result in your answer not being graded.*

1. Jika Anda tidak membaca peraturan ini, maka Anda dianggap sudah membaca dan menyetujuinya.

*By taking this exam, you agree to these regulations, regardless of whether you have read it or not.*

1. Persentase penilaian untuk matakuliah ini adalah sebagai berikut:

*The score will be distributed as follows:*

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

1. Perangkat lunak yang digunakan pada matakuliah ini adalah sebagai berikut:

*This course uses the following software:*

|  |
| --- |
| **Software**  *Software* |
| Eclipse 2020.6 R  Java 11.0.18  JavaFX 17.0.7  MySQL Java Connection Library 8.0.24  XAMPP 8.0.7 |

1. Ekstensi file yang harus dikumpulkan untuk matakuliah ini adalah sebagai berikut:

*Your answers must be in the following file extensions:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| JAVA. CLASS | JAVA. CLASS, SQL | - |

1. File yang harus dikumpulkan adalah keseluruhan jawaban beserta dengan aset yang digunakan (gambar, audio, video, dll) dan dokumentasi proyek yang berisikan link referensi aset dan penjelasan mengenai aplikasi yang dibuat (terlampir bersama dengan soal).

*Include 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.*

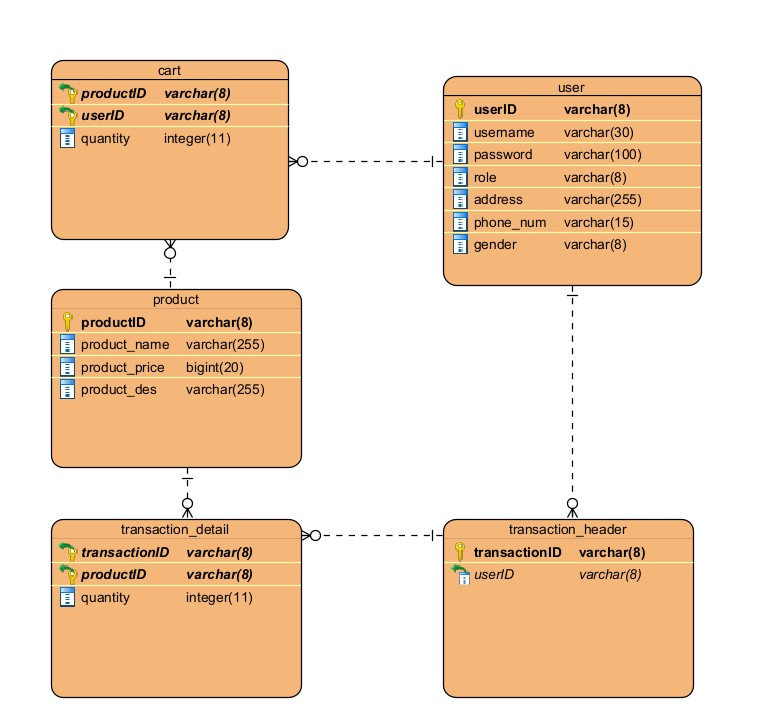
## Soal

*Case*

**SeRuput Teh**

SeRuput Teh is an expanding store specializing in premium tea products. SeRuput Teh aims to enhance the customer experience by launching a digital platform. The owner has given you their trust to develop a Java application to manage SeRuput Teh’s inventory and transactions. The program should be made **using Java Programming Language**, with **MySQL Database Engine**. **Ensure that you didn’t use regex, Java FXML, JavaFX Scene Builder, or any library used outside the class materials in your project or it will affect your score.**

There are some requirements to build this application:

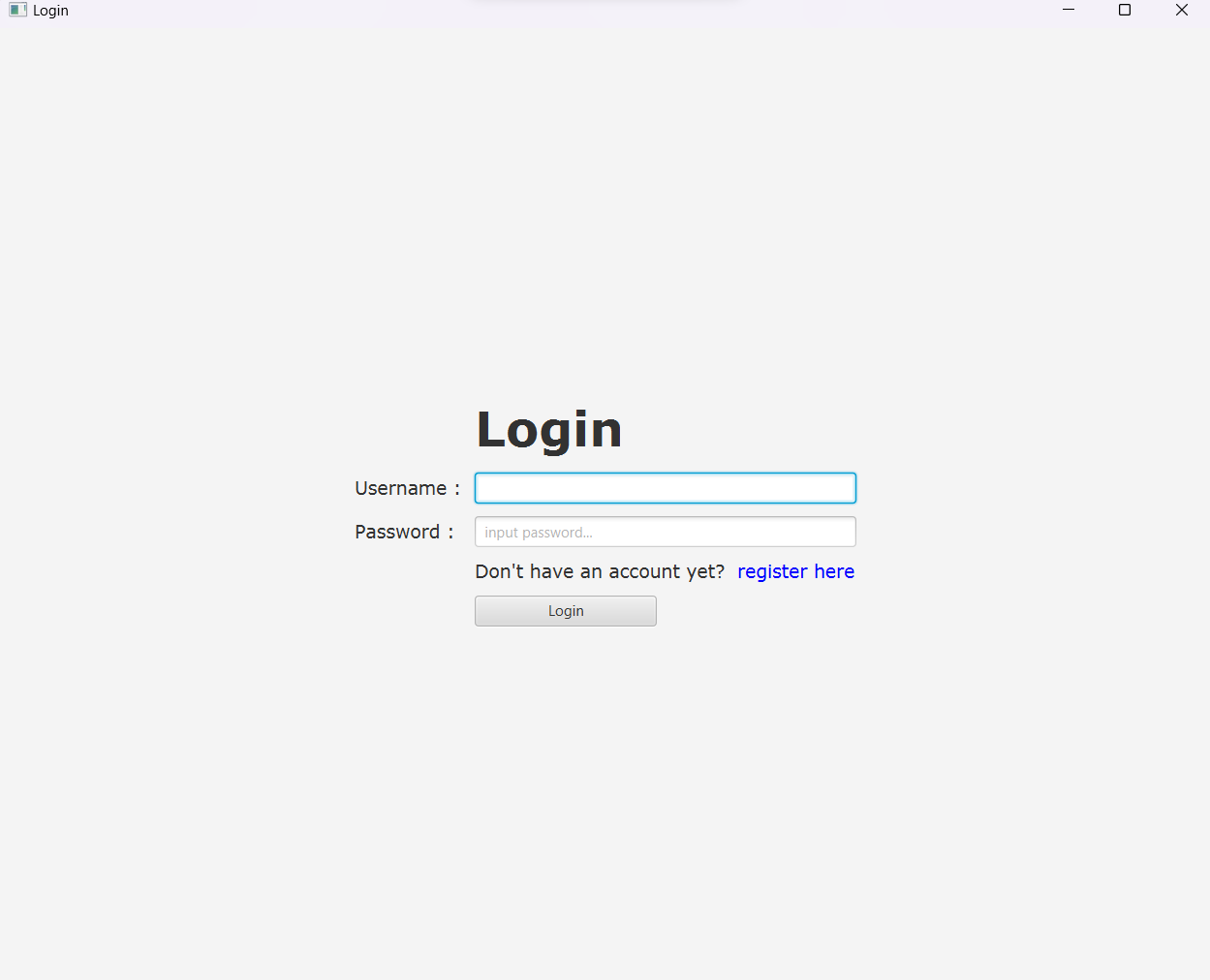


**Figure 1 ERD.**

Execute the provided create and insert query file on “seruput\_teh” database! (“create+insert.sql”).

* **Login Scene**

Login form is the first form that will appear when the program starts. The **Login** **Form** is used by users to **login**, in order to access the **Home Scene**.



**Figure 2 Login Form.**

***Components:***

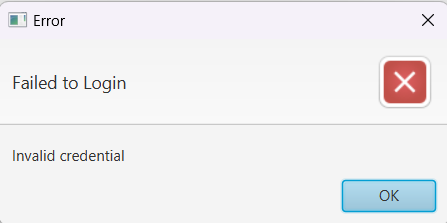
* + **Label** for Login, Username, Password, Register (“register here”), and Description (“Don’t have an account yet?”).
  + **TextField** for Username input.
  + **PasswordField** for Password input.
  + **Button** for Login Button.

**Description:**

* + **If user clicks Login Button**, then **validate**:
    - **Username and Password must be filled in**.
    - **Username and Password must exist and match** in database.
    - The program will **validate role (customer/admin).**
    - If the **credential is correct**:

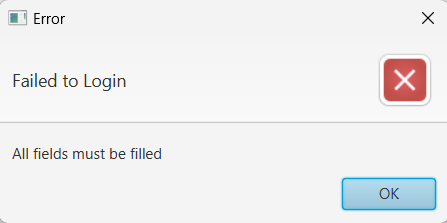
Direct the user to **Home Scene** if the **user’s role is “Customer”** or direct the user to **Admin Scene** if the **user’s role is an “Admin”**.

* + **If the user** **clicks Register Label**, redirect the user to the **Register Scene**.
  + If the user **failed to log in**, display an **Error Alert**.



**Figure 3 Error Alert if user failed to log in.**

* + If the user did not fill all the text field and password field, display an Error Alert



**Figure 4 Error Alert if user did not fill all the fields.**

* **Register Scene**

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

**A screenshot of a computer

Description automatically generated with medium confidence**

**Figure 5 Register Form.**

***Components:***

* + **Label** for Register, Username, Email, Password, Confirm Password, Phone Number, Address, Gender, Description (“Have an account?), and Login (“login here”).
  + **TextField** for Username input, Email input, Phone Number input, and Address input.
  + **PasswordField** for Password input and Confirm password input.
  + **RadioButton** for Gender input.
  + **TextArea** for Address input.
  + **CheckBox** for agreeing to Term & Conditions.
  + **Button** for Register button.

**Description:**

* + **If user clicks Register Button**, then validate:
    - **User ID** generated with **format**:

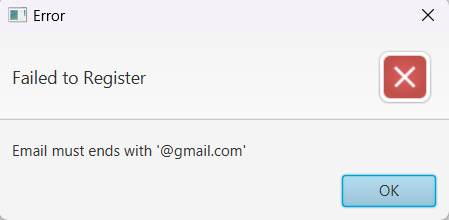
**CUXXX**

Where **X**: Digit (0-9) according to customer index (plus by one on insert)

Example: CU001, CU002, CU003

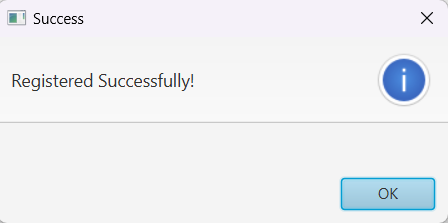
**R**

* + - Users who registered are **assigned as “Customer”**.
    - **Email** must **end with ‘@gmail.com’**.
    - **Username** must be **unique**.
    - **Username** must be **5-20 characters**.
    - **Password** must be **alphanumeric**.
    - **Password** must be **at least 5 characters**.
    - **Confirm password** must **equals to password**.
    - **Phone number** must be **numeric**.
    - **Phone number** must **start with ‘+62’**.
    - **All fields** must **be filled**.
  + If the user’s input does not meet the requirements, show an Error Alert. Every error message must be shown.



**Figure 6 Error Alert for email error.**

* + If the user **does mee**t the requirements and **successfully registered** themselves, **insert** the user **to the database** with the **assigned role and User ID**. Then, **show an Information Alert** and **redirect the user to the Login Scene**, to log in with the new existing account.

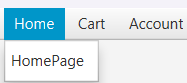


**Figure 7 Information Alert if user is registered successfully.**

* **Navigation Bar**
  + **Customer Navigation Bar**

**Customer Navigation Bar** is a **navigation bar** that is **available for all customers**. This navigation bar is **available on all scenes except Login Scene and Register Scene**. The navigation bar consists of **Home Menu, Cart Menu, and Account Menu**.

* + - **Home Menu** consists of **Home Page Menu Item**.
    - **Cart Menu** consists of **My Cart Menu Item**.
    - **Account Menu** consists of **Log out Menu Item** and **Purchase History Menu Item**.



**Figure 8 Customer Navigation Bar.**

***Components:***

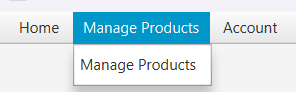
* + - * **MenuBar** for containing Home, Cart, and Account Menu
      * **Menu** for Home, Cart, and Account Menu
      * **MenuItem** for Home Page, My Cart, Purchase History, and Log out Menu Item

**Description:**

* + - * **Home Page** is used to **navigate to Home Scene** when clicked or selected.
      * **My Cart** is used to **navigate to Cart Scene** when clicked or selected.
      * **Purchase History** is used to **navigate to Transaction Scene** when clicked or selected.
      * **Log Out** is used to **logout of the current account** then user will be **redirected** to the **Login Scene**.
  + **Admin Navigation Bar**

**Admin Navigation Bar** is a **navigation bar** that is **available for all admins**. This navigation bar is **available on all scenes except Login Scene and Register Scene**. The navigation bar consists of **Home Menu, Manage Products Menu, and Account Menu**.

* + - **Home Menu** consists of **Home Page Menu Item**.
    - **Manage Products Menu** consists of **Manage Products Menu Item**.
    - **Account Menu** consists of **Log out Menu Item**.



**Figure 9 Admin Navigation Bar.**

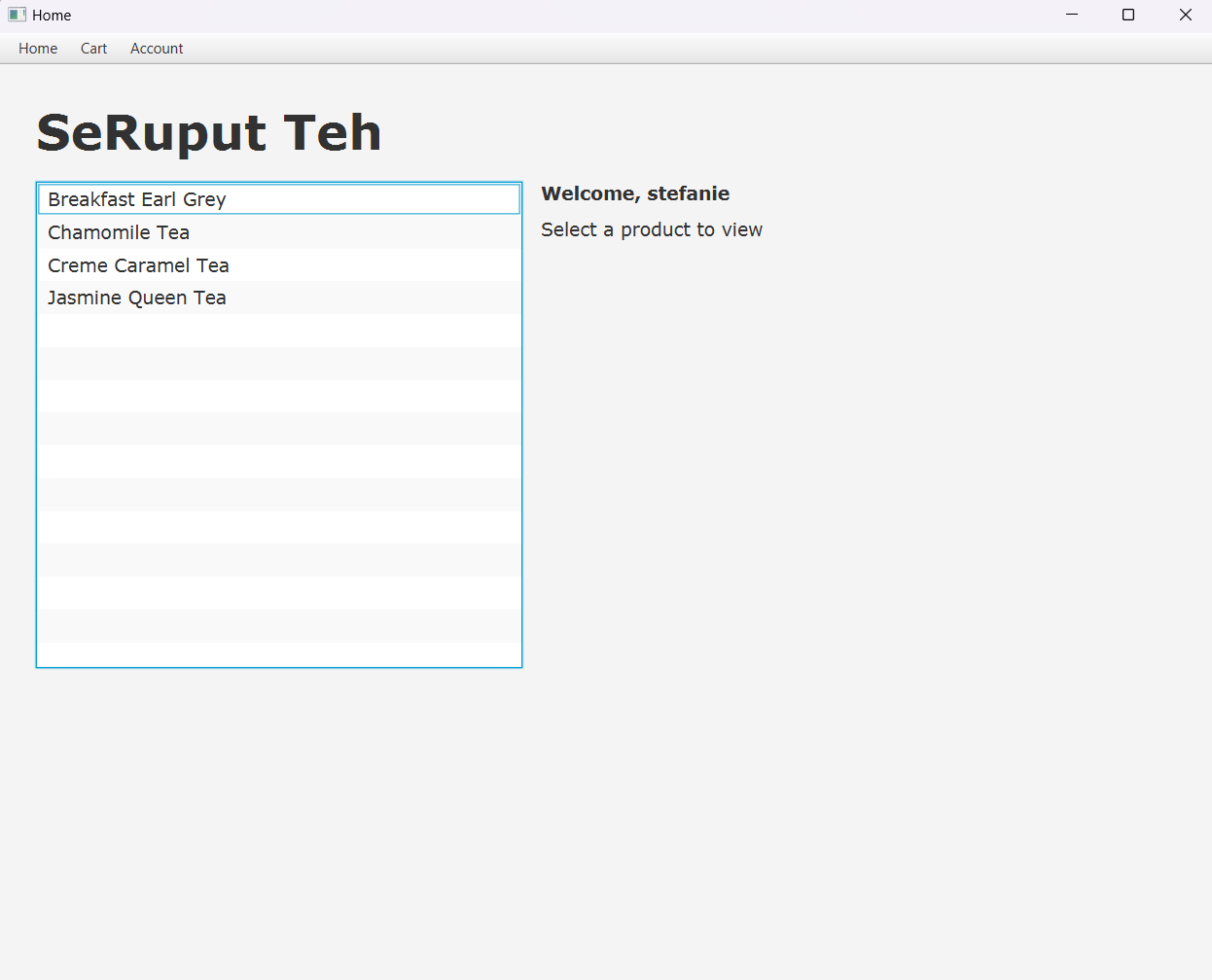
***Components:***

* + - * **MenuBar** for containing Home, Manage Products, and Account Menu
      * **Menu** for Home, Manage Products, and Account Menu
      * **MenuItem** for Home Page, Manage Products, and Log out Menu Item

**Description:**

* + - * **Home Page** is used to **navigate to Home Scene** when clicked or selected.
      * **Manage Products** is used to **navigate to Edit Product Scene** when clicked or selected.
      * **Log Out** is used to **logout of the current account** then user will be **redirected** to the **Login Scene**.
* **Home Scene**

**Home Scene** is **available for all users**. In this scene, the user can **view all products** that are **available in the database** on a **ListView**. Users can **click on one of the products** on the list **to view the details of the selected product** **including a button to add the product to the cart**.



**Figure 10 Home Form (no product is selected) from customer’s view.**

A screenshot of a computer

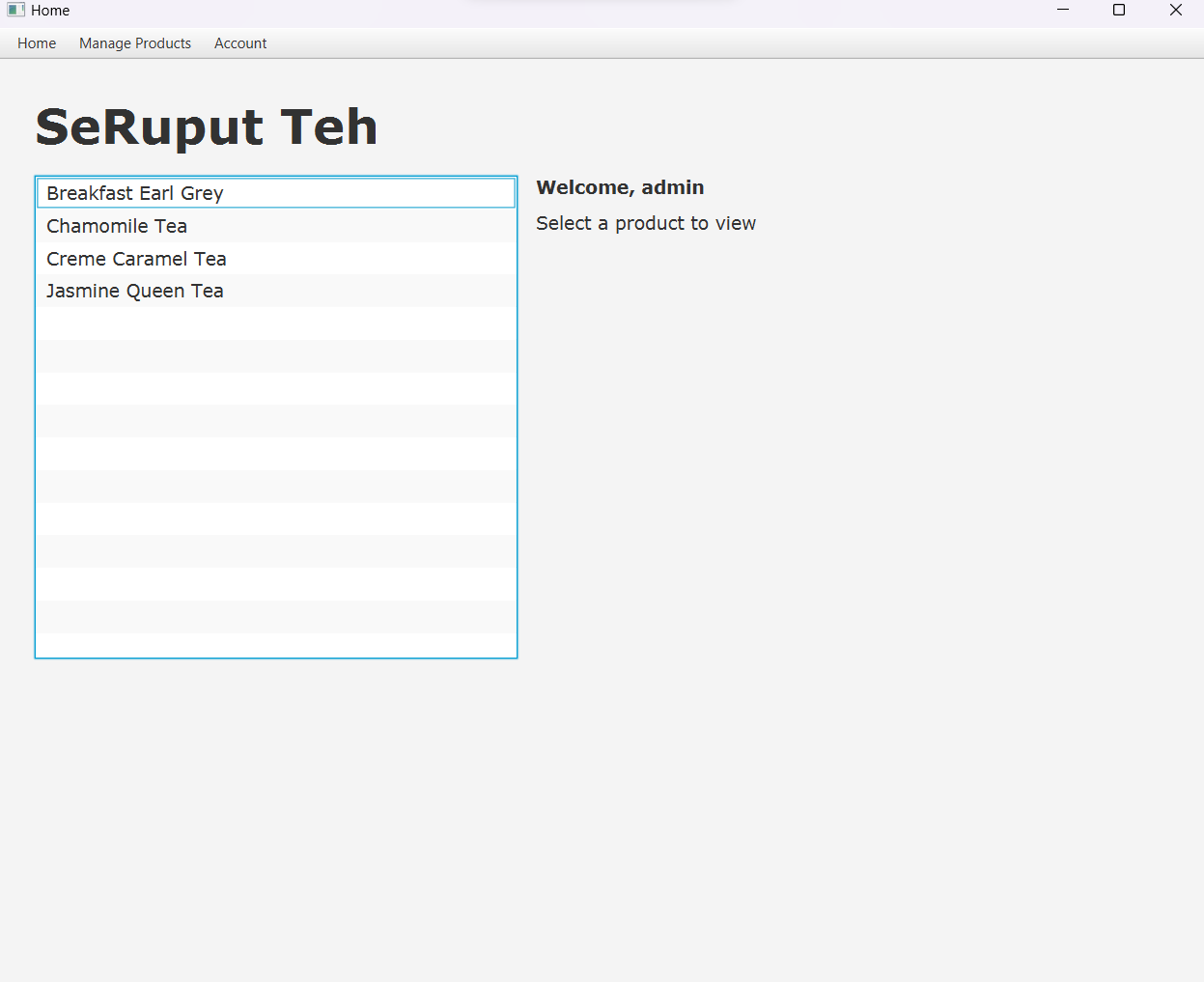
Description automatically generated

**Figure 11 Home Form (a product is selected) 1.0 from customer’s view.**

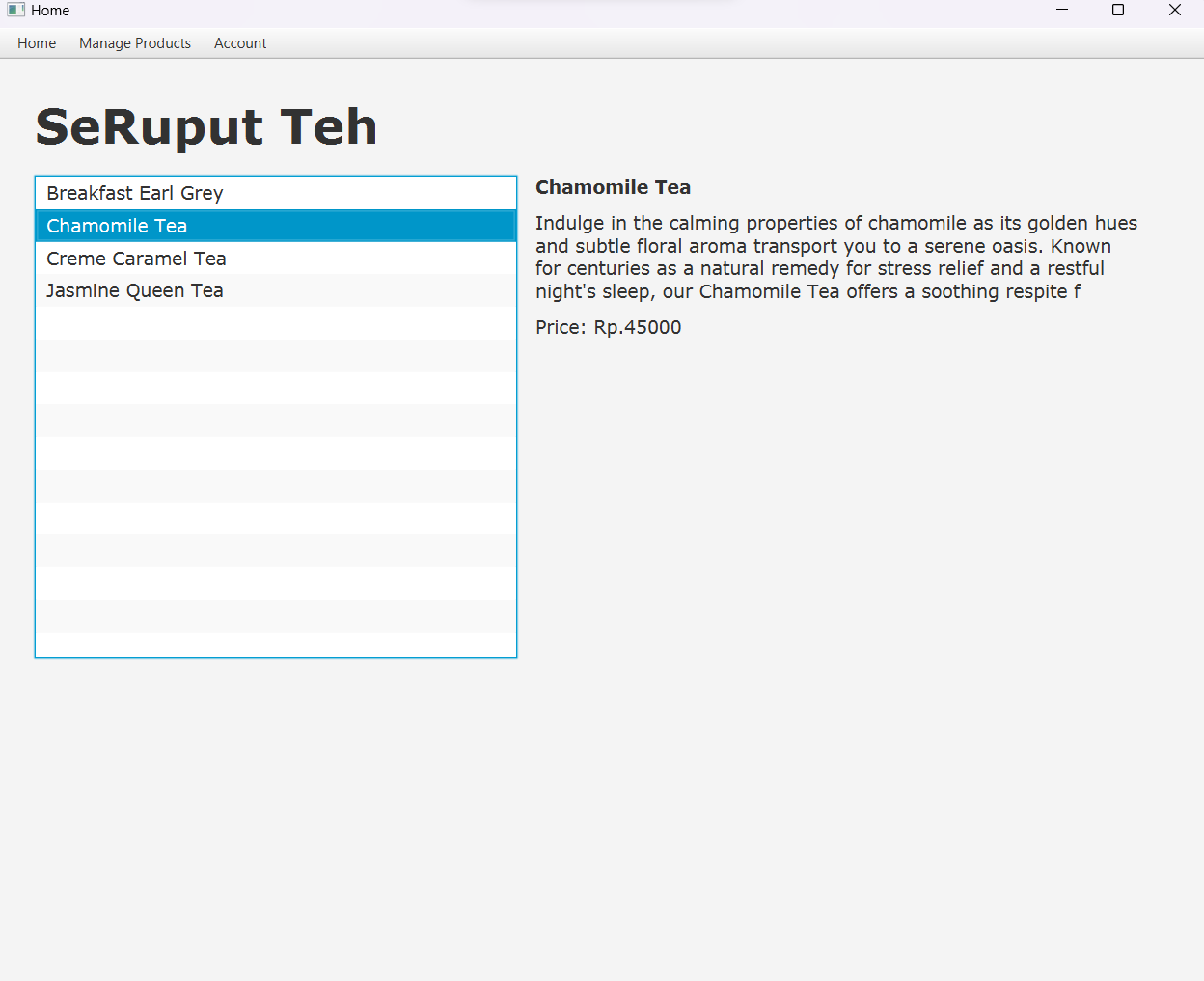
A screenshot of a computer

Description automatically generated

**Figure 12 Home Form (an product is selected) 2.0 from customer’s view.**



**Figure 13 Home Form (no product is selected) from admin's view.**



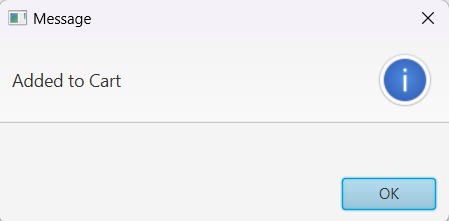
**Figure 14 Home Form (a product is selected) from admin's view.**

***Components:***

* + **Label** for:
    - SeRuput Teh
    - Product Name
    - Product Detail
    - Product Price
    - Total Price
    - Quantity
    - Welcoming Message
  + **ListView** for displaying all the products available from database.
  + **Spinner** for inputting the preferred quantity of the product.
  + **Button** for Add to Cart.

**Description:**

* + Display the product’s name for each product in the ListView.
  + When the user first enters the Home Scene, **no product is selected** and **a Welcoming Message is displayed**. The welcome message is followed by the username of the current user and an instruction as shown in figure 10.
  + When a product is **clicked**, the welcoming message will be replaced by the **details of the product**.
    - If the user is **a customer**, display (as shown in figure 11 and 12):
      * Product Name
      * Product Detail
      * Product Price
      * Quantity Label
      * Quantity Spinner
      * Total Label
      * Add to Cart Button
    - If the user is **an admin**, display (as shown in figure 13 and 14):
      * Product Name
      * Product Detail
      * Product Price
    - All product’s information **should be obtained from the database**.
  + Total label **will only be displayed** if **the value in the spinner is more than 1**.
  + Total label will **display the sum of the current value in the spinner multiplied with the selected product’s price** as shown in figure 12. Therefore, **if the value of the spinner changes, the total label will have to update its displayed text**.
  + When the **Add to Cart button is clicked**,
    - If the **selected product is not in the current user’s cart**, the selected product with the preferred quantity from the spinner **will be added to the current user’s cart**. **The data will also be stored on the Cart table in the database**.
    - If the **selected product is already in the current user’s cart**, the data of the product in the **Cart table will be updated**. **The quantity** of the product **will be updated** to the **sum of the quantity in the cart plus the value of the spinner**.
  + **Display an Information Alert** if the product is **successfully added to the cart** and an **Error Alert** if **failed**.



**Figure 15 Information Alert if a product is successfully added to cart.**

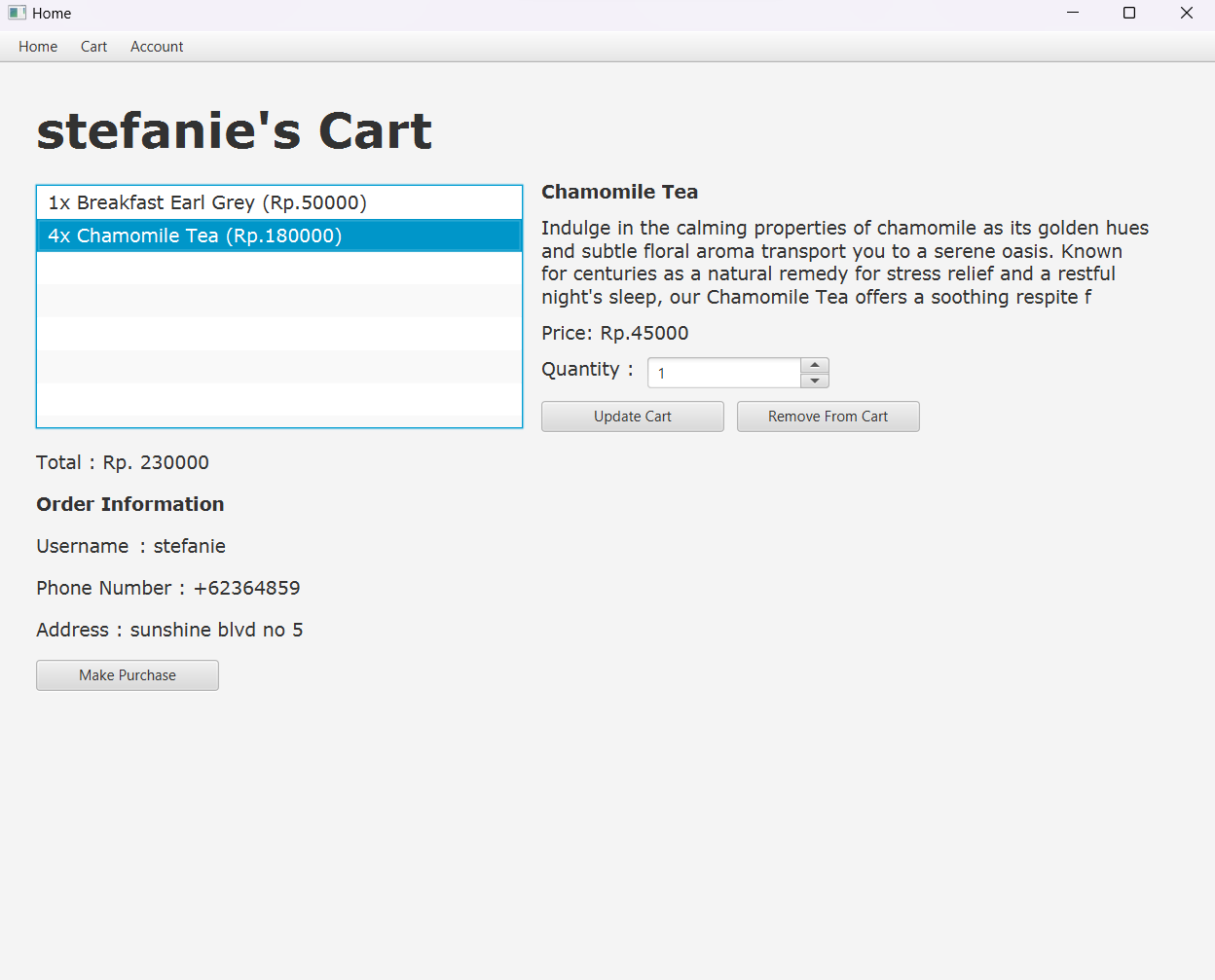
* + A product **can only be added to cart** if the **quantity is at least one product**.
* **Cart Scene**

**Cart Scene** is **only available** to users that is a **“Customer”**. In this scene, **all the products in the current user’s cart will be displayed** using a **ListView**. **The data** that will be displayed **can only be obtained from the Cart table in the database**. Users can **interact with the ListView** to **view the details** of the selected product, **updating its quantity** and **remove** the product from cart.

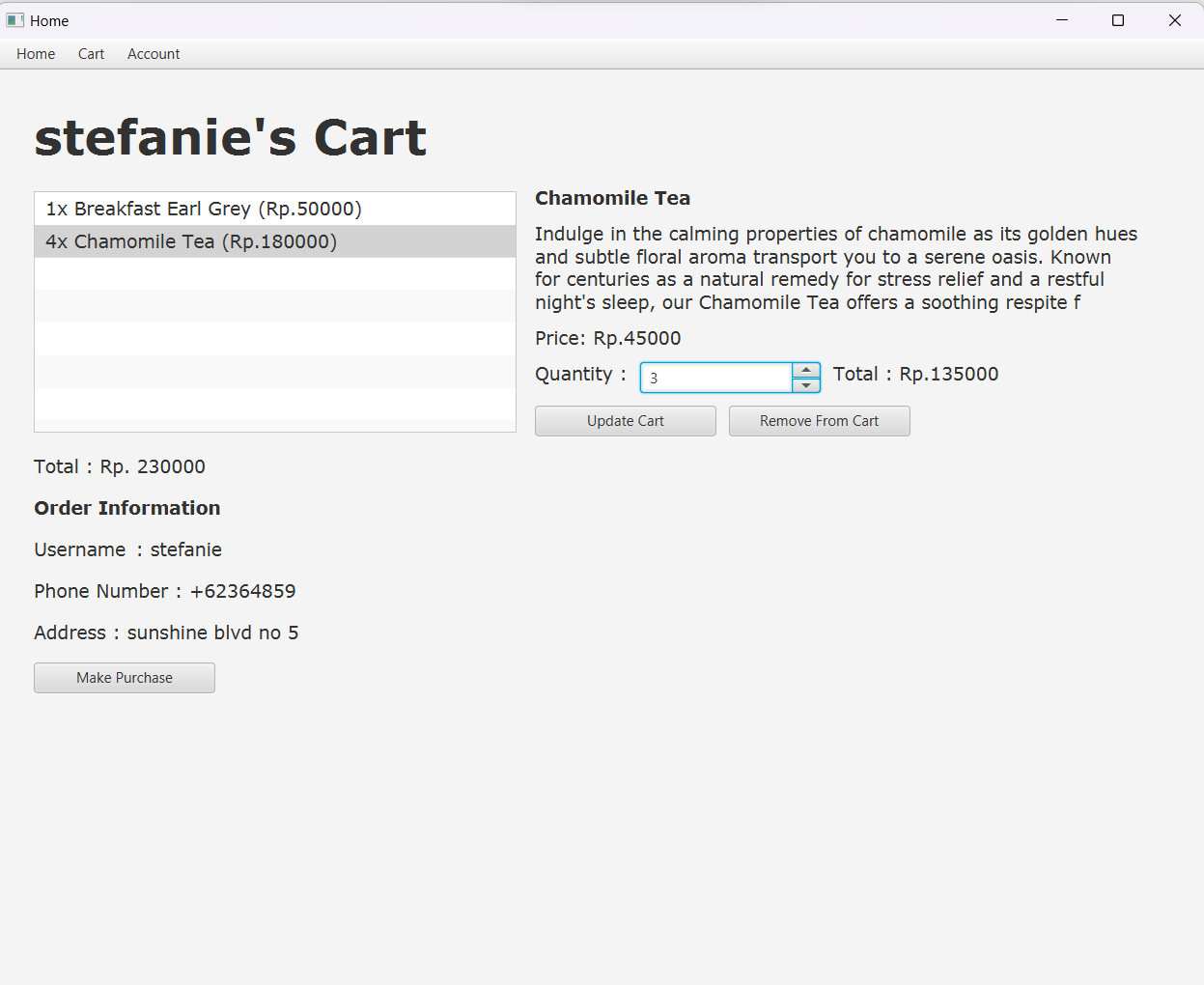
A screenshot of a computer

Description automatically generated with medium confidence

**Figure 16 Cart Form (no product is selected).**



**Figure 17 Cart Form (a product is selected) 1.0.**



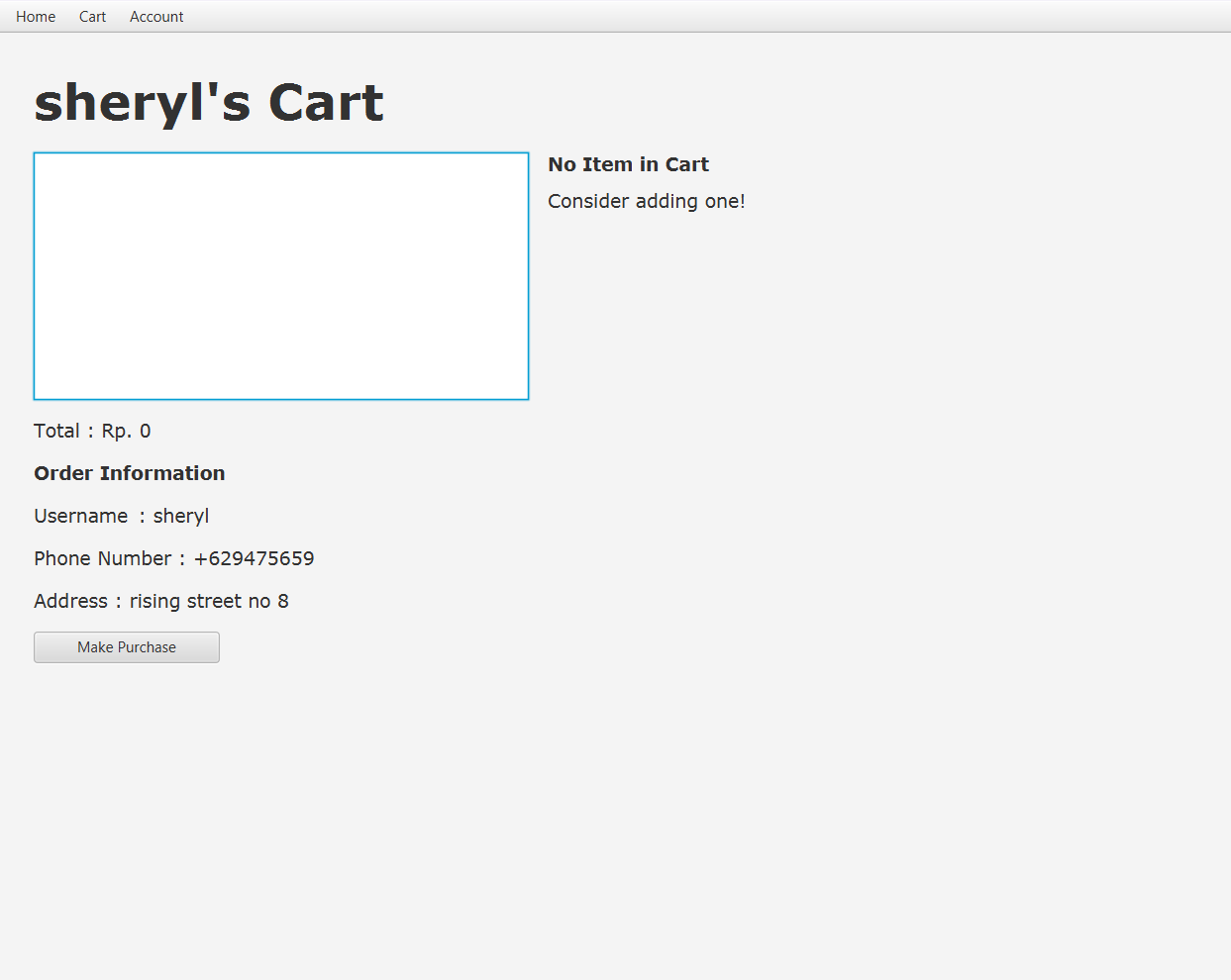
**Figure 18 Cart Form (a product is selected) 2.0.**

***Components:***

* + **Label** for:
    - [Username]’s Cart
    - Product Name
    - Product Detail
    - Product Price
    - Quantity Label
    - Total Label
    - Total Price of the Cart
    - Order Information
    - Username Information
    - Phone Number Information
    - Address Information
  + **Spinner** for product quantity.
  + **Button** for:
    - Update Cart
    - Remove From Cart
    - Make Purchase
  + **ListView** for viewing all the products in cart.

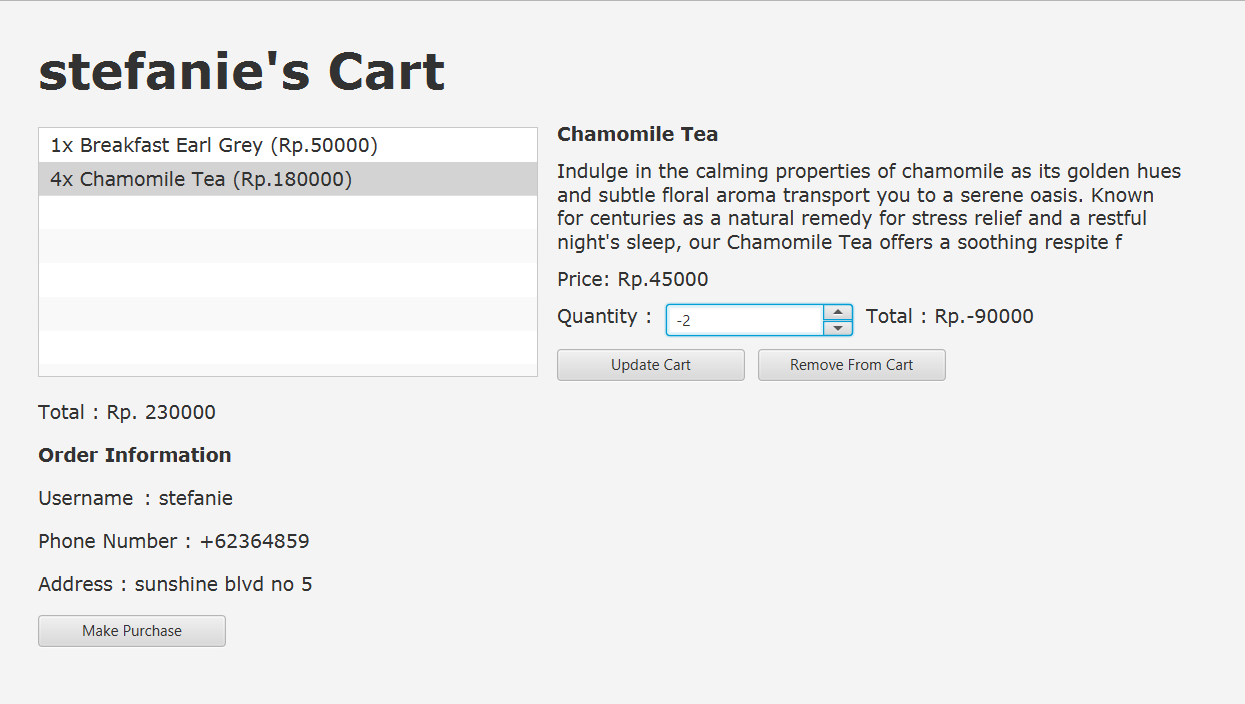
**Description:**

* + Display the **product’s name, quantity, and total price** for each product in the **Cart Table from database**. Total price is obtained from **multiplying the product price with the quantity**. The data should be **displayed in ‘[quantity]x [product name] ([total price])’ format**.
  + When the user first enters **Cart Scene**, **no product is selected,** and a welcoming message is displayed as shown in figure 16. The message should consist of a welcome, username of the current user and instructions.
  + If the user has **no added products to their cart**, an **empty message** should be displayed.

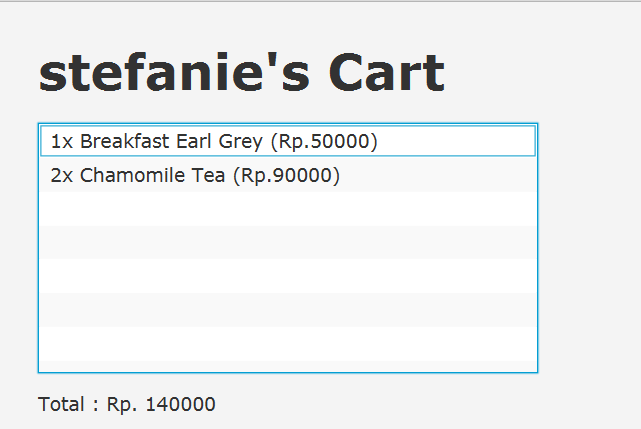


**Figure 19 Cart Form if the user has no product in cart.**

* + When a product is **clicked**, the welcoming message will be replaced by the **details of the product** as shown in figure 17.
    - The details that need to be displayed are:
      * Product Name
      * Product Detail
      * Product Price
      * Quantity Label
      * Spinner Quantity
      * Total Price
      * Update Cart Button
      * Remove From Cart Button
    - All the products information **must be obtained** from Cart Table and Product Table in the database.
  + User can **update the quantity** of a product by **selecting** the product and by using the quantity spinner,
    - If the value of the spinner is **negative**,
      * If the sum of the quantity in cart plus value of the spinner is equals to or **more than one**, the quantity of the selected product in the cart will be **decreased** according to the spinner’s value.

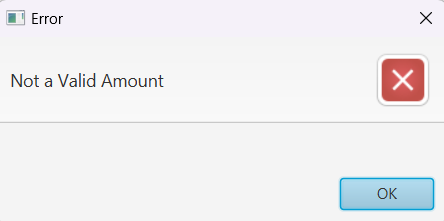


**Figure 20 Product Form before decreasing the quantity.**



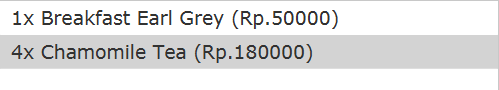
**Figure 21 Product Form after decreasing the quantity.**

* + - * If the sum of the quantity in cart plus the value of the spinner **equals to zero**, the product will be **removed**.
      * Other than that, show an Error Message displaying that the quantity is not valid.



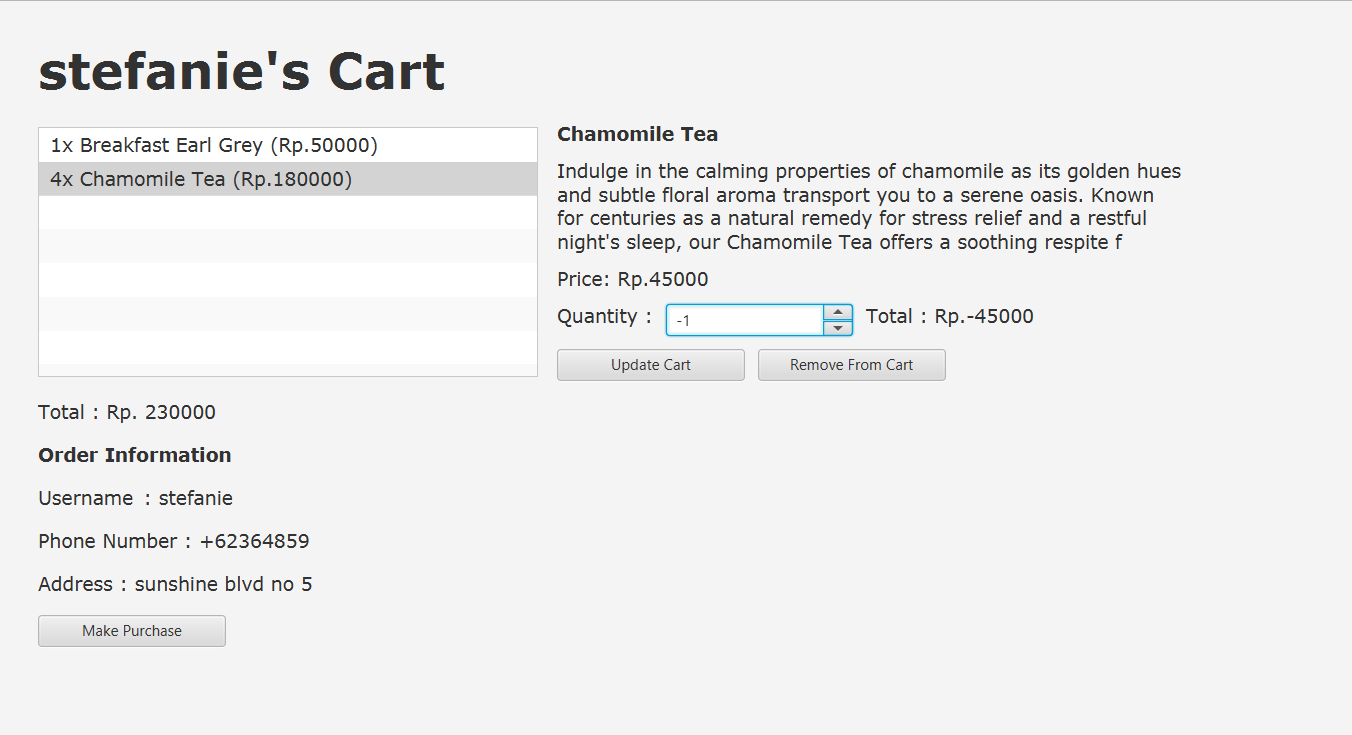
**Figure 22 Error Label Quantity.**

* + - If the **value of the spinner is zero**, show an Error Alert displaying a message that it is not a valid amount as shown in figure 22.
    - If the value of the spinner is **positive**, update the quantity of the selected product to the sum of the product’s quantity in cart plus the value of the spinner.
  + Users can **remove** a product by **selecting** the preferred product and **clicking** the Remove Product button.
  + **Total Price of Cart** is obtained by **adding all the product’s prices**. The product prices that this statement is referring to is displayed in figure 23.



**Figure 23 Product Price that should be calculated to get total price of the cart.**

* + For **every update** that is made to the cart, the ListView, Total Price of the Cart and Label **should all be updated**.
    - **Total Price of Cart** should be updated to the **current total price of all the product prices.** The product price that this statement is referring to is displayed in figure 22.
    - **ListView** should be updated to **the newest data** in the Cart Table in the database.
    - **Product details** of the previously selected product should **no longer be displayed**.



**Figure 24 Cart Form before decrease quantity update.**

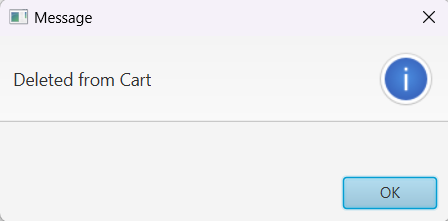


**Figure 25 Cart Form after quantity update.**

* + If an update that is made to the cart **is successfully executed**, an Information Alert should be displayed.

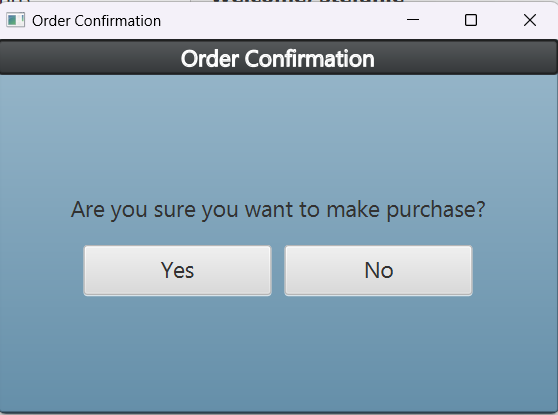


**Figure 26 Information Alert if successfully update quantity.**



**Figure 27 Information Alert if successfully update quantity.**

* + If an update that is made to the cart **failed**, an Error Alert should be displayed.
  + Order Information that consists of username, phone number, and address should all be **adjusted** according to the user’s data that can be obtained from the User Table in the database.
  + Users can **make purchases** by **clicking** the Make Purchase button. When the button is clicked, **a confirmation pop up should appear**.
* **Confirmation Pop Up**



**Figure 28 Order Confirmation Pop Up.**

***Components:***

* + **Label** for confirmation question.
  + **Button** for yes and no button.

**Description:**

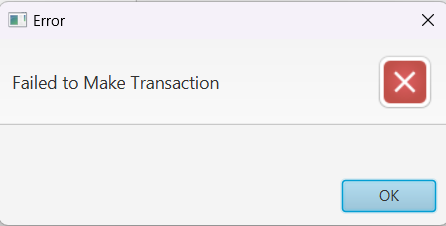
* + If the no button is **clicked**, the pop up **will close**.
  + If the yes button is **clicked**,
    - If the cart **is empty**, an Error Alert should be displayed as shown in figure 29, and the pop **will close**.
    - If the cart **is not empty**, a purchase will be made. The data in the Cart Table should **be moved** to the Transaction Header and Transaction Detail Table and an Information Alert should be displayed as shown in figure 30. **Clear up** the user’s cart in the database. **Refresh and reset** the form into default layout as shown in figure 19. Then the pop up **will close**.
    - Transaction ID **is generated** with the following format.

**TRXXX**

Where **X**: Digit (0-9) according to transaction index (plus by one on insert)

Example: TR001, TR002, TR003

**R**



**Figure 29 Transaction Error Alert.**

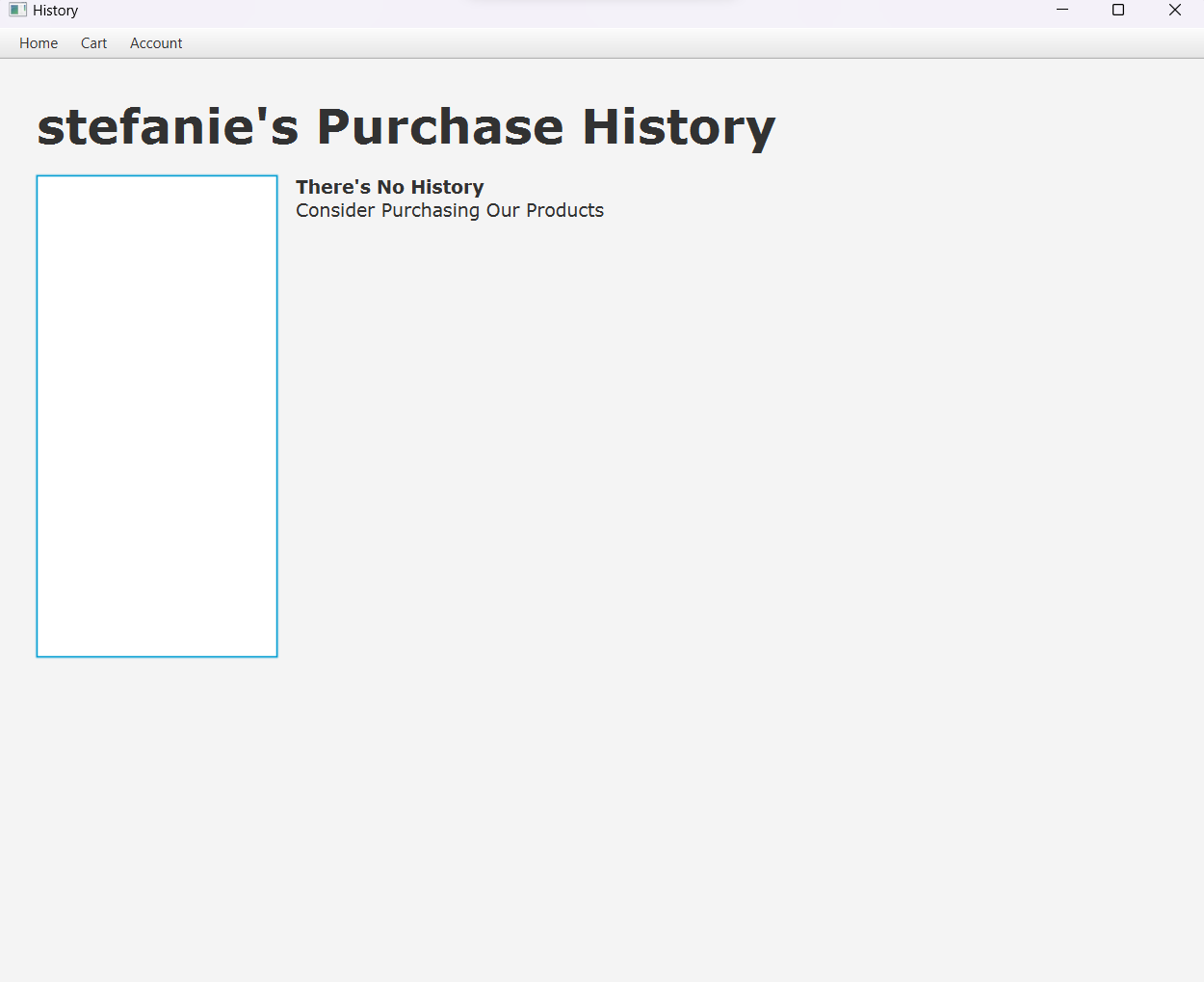
A screenshot of a computer

Description automatically generated with medium confidence

**Figure 30 Transaction Information Alert**

* **Transaction Scene**

**Transaction Scene** is **only available** to users that is a **“Customer”**. This scene’s main objective is to display all the transactions that are done by the current user. All the transactions will be **displayed on a ListView**. Users can interact with the ListView to view details of the selected transaction.



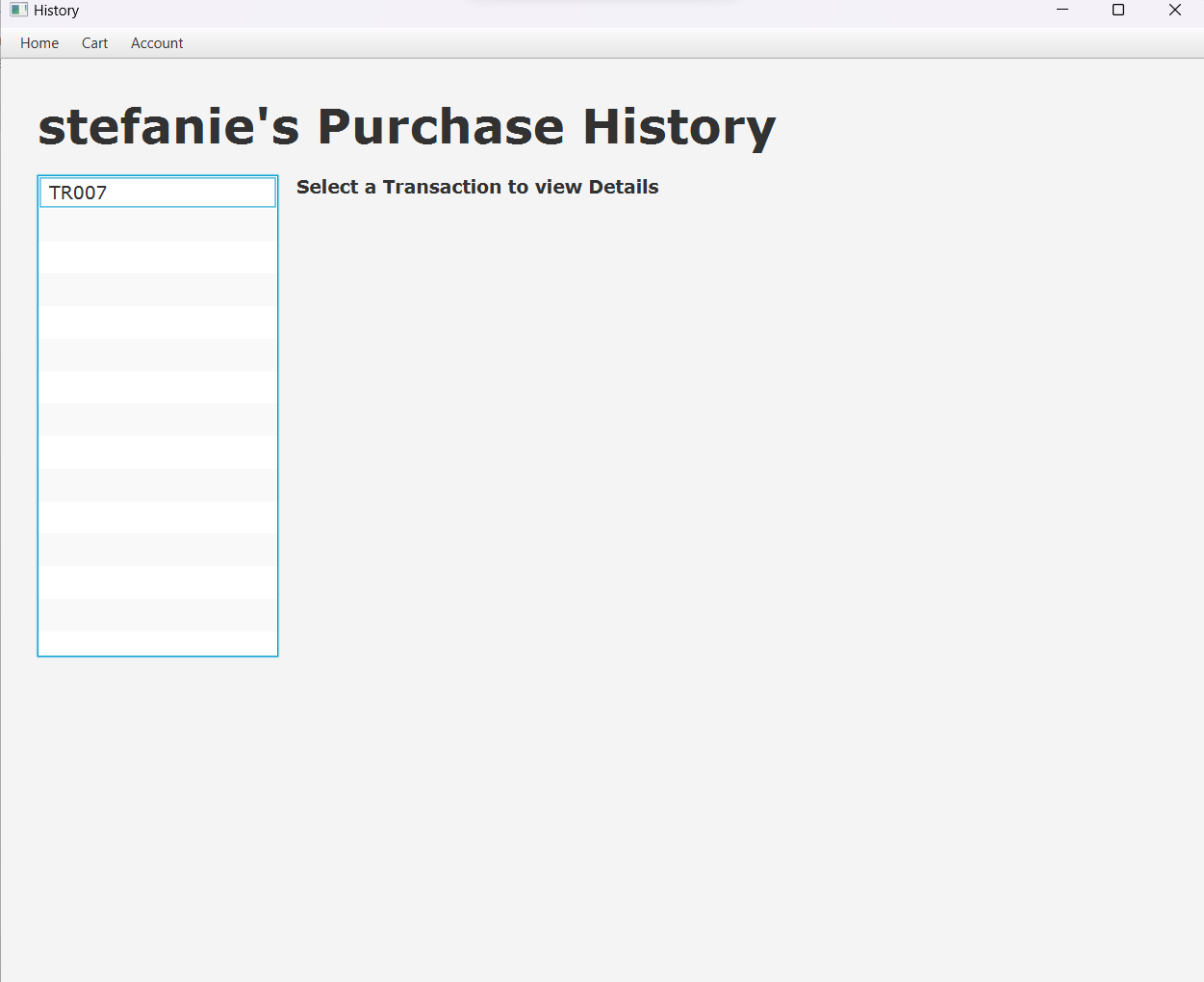
**Figure 31 Purchase History Form if there is no purchase made.**

***Components:***

* + **Label** for:
    - [username]’s Purchase History
    - Transaction ID
    - Username
    - Phone Number
    - Address
    - Welcome Message
    - Total Price
  + **ListView** for list of transactions and transactions details

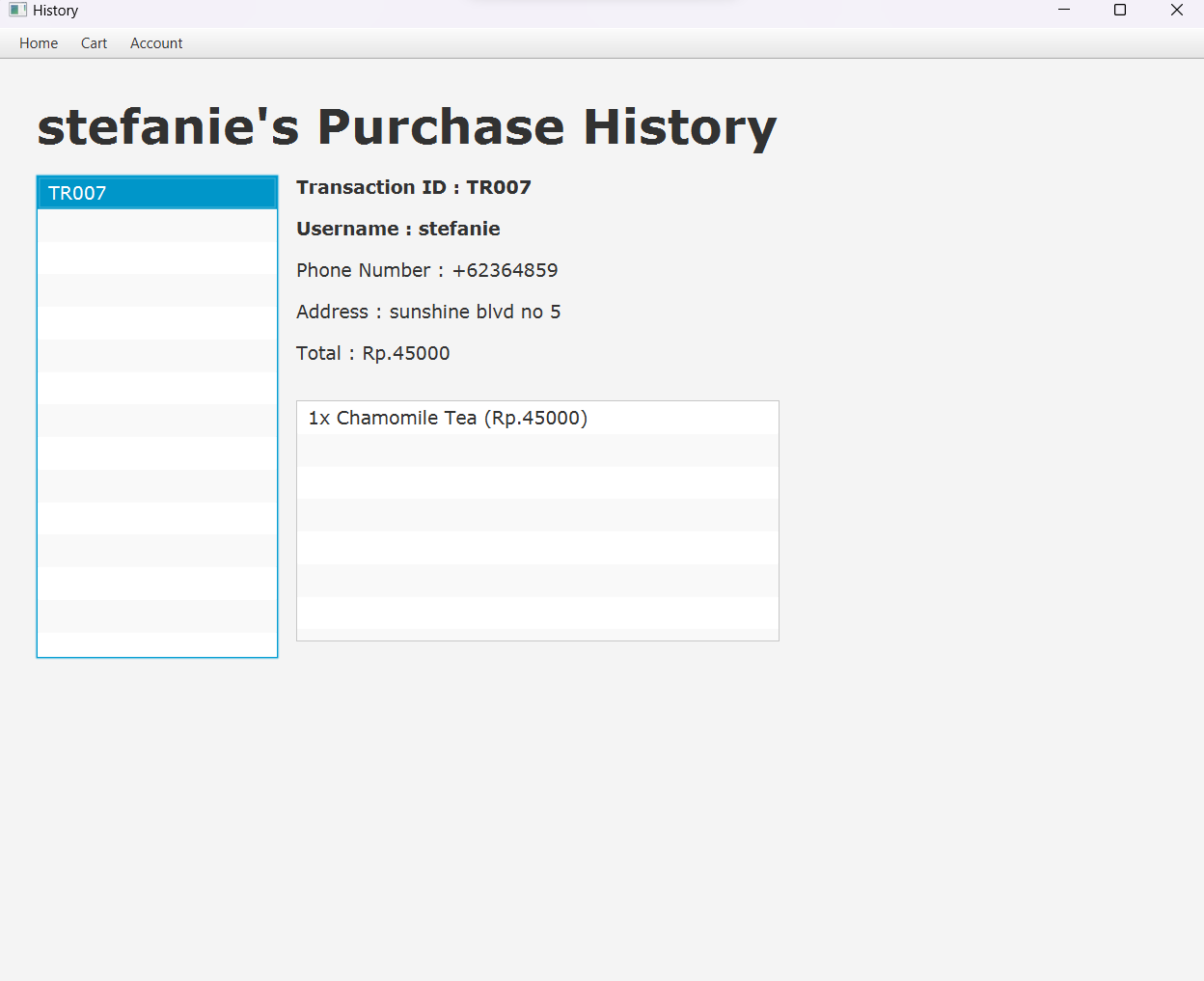
**Description:**

* + When the user first enters purchase history, **display an empty message** as shown in figure 31 **if the user has not made any transactions or purchases** yet. If the user **has made any transactions or purchases**, **display an instruction** as shown in figure 32.
  + Display the current user’s username combined with purchase history on the left top of the form.



**Figure 32 Purchase History Form if there is purchase made.**

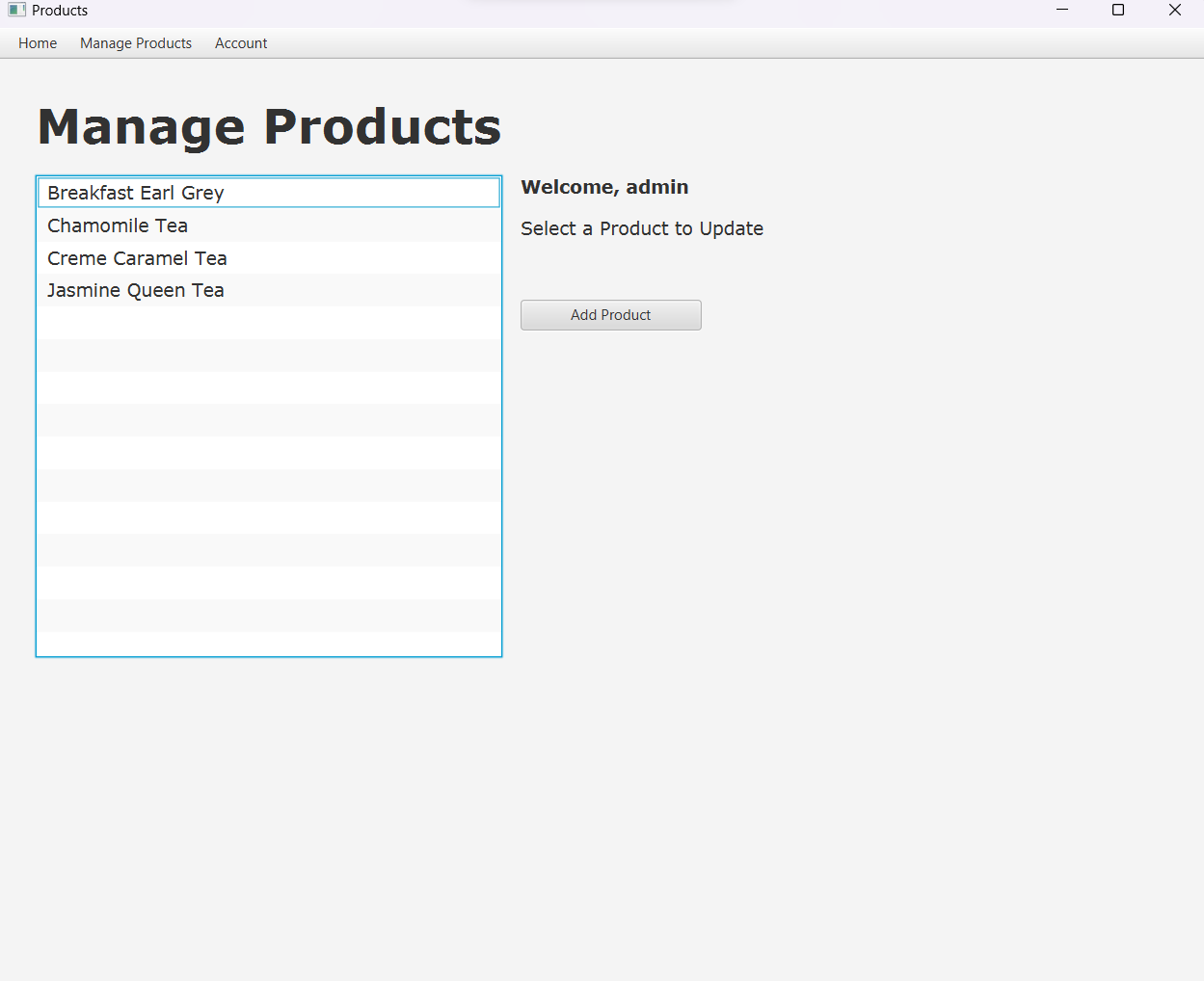
* + The starting ListView must display **all the of Transaction ID** from all the transactions that has been made by the current user.
  + When one of the Transaction ID **is clicked**, the empty or instruction message **will be replaced by the details of the selected transaction**.



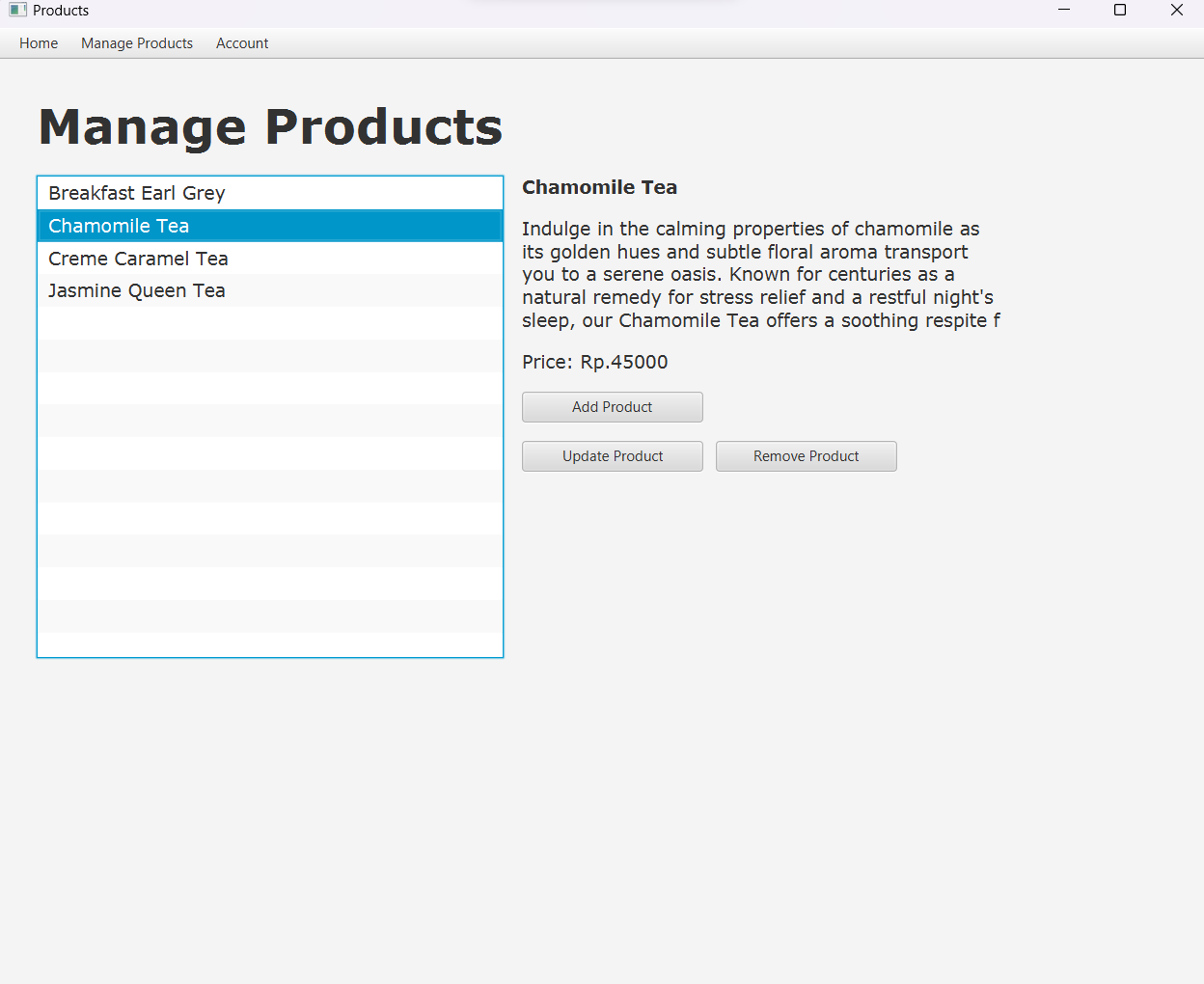
**Figure 33 Purchase History Form (a transaction is selected)**

* + Display the transaction information:
    - Display Transaction ID **according to the selected transaction**.
    - Display the username **according to the current user’s username**.
    - Display the phone number **according to the current user's phone number**.
    - Display the address **according to the current user's address**.
    - Display the total price **according to the total price of the transaction**.
  + Display all the transaction details in a ListView with **‘[quantity]x [product name] ([total product price])’ format** as shown in figure 33.
  + **All the data must be obtained from the database**.
* **Edit Product Scene**

**Edit Product Scene** is **only available** for users that is an **“Admin”**. In this scene, users can view all of the products in the database and have the authority to insert a new product into the database, update a product’s price, and delete a product from the database.



**Figure 34 Manage Products Form (no selected product)**



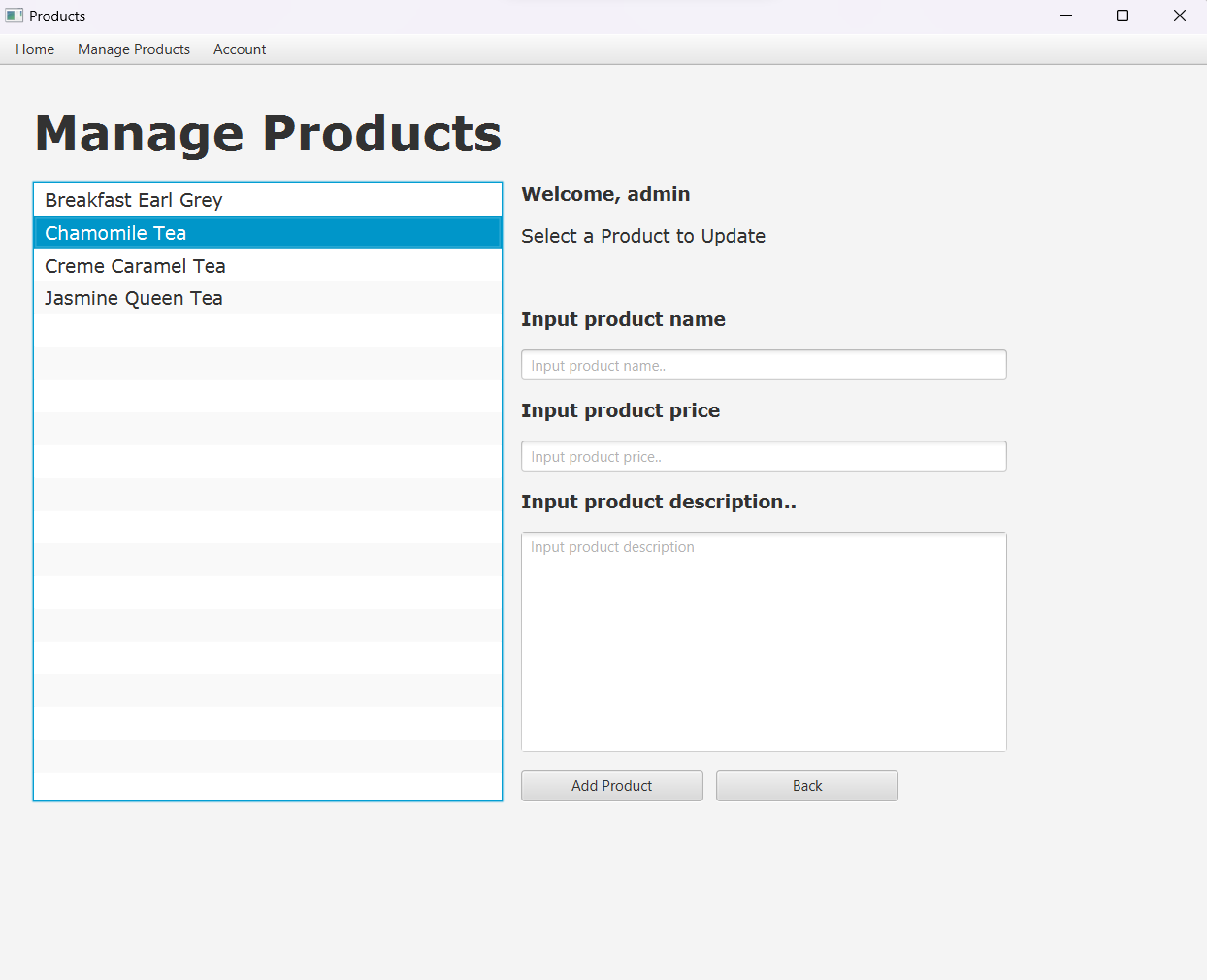
**Figure 35 Manage Product Form (a product is selected)**

***Components:***

* + **Label** for:
    - Manage Products
    - Product Name
    - Product Details
    - Product Price
    - Input Product Price
    - Input Product Name
    - Input Product Description
    - Update Product
    - Remove Confirmation
  + **Button** for:
    - Add Product Button
    - Back Button
    - Update Product Button
    - Remove Product Button
  + **TextField** for product’s name input, product’s price input and product’s price description.
  + **ListView** for viewing all the products.

**Description:**

* + When the user first enters Manage Products Form, display welcome message that consists of a welcome, the current user’s username, an instruction, and an Add Product button.
  + When the Add Product button is **clicked** display as shown in figure 36 and 37:
    - Input Product Name Label
    - Input Product Price Label
    - Input Product Description Label
    - Product Name TextField
    - Product Price TextField
    - Product Description TextField
    - Add Product Button
    - Back Button
  + If the **Add Product Menu is displayed and the add product button is clicked**, the product **will be added to the database if it meets the requirements** below, (the Add Product Menu that this statement is referring to is shown in figure 38):
    - All fields **must be filled out**.
    - Product name **must be unique**.
    - Product price **must be more than 0**.
  + If one of the inputs **does not meet** the requirements, show an Error Alert.
  + If a product is **successfully added to the database**, show an Information Alert and update the Product Table in the database. If it’s **not successful**, show an Error Alert.
  + If the back button is **clicked**, the Form **will be reset** to the default layout that is shown in figure 34.



**Figure 36 Manage Product Form (no product is seleceted and add product button clicked)**

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 37 Manage Product Form (a product is seleceted and add product button clicked)**

A screenshot of a computer

Description automatically generated with low confidence

**Figure 38 Add Product Menu**

* + If a product is **selected** display (as shown in figure 35):
    - Product Name
    - Product Detail
    - Product Price
    - Add Product Button
    - Update Product Button
    - Remove Product Button
  + If the Update Button is **clicked**, display:
    - Product Price TextField
    - Input Product Price Label
    - Update Product Button
    - Back Button

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 39 Manage Product Form (a product is selected and update product button is clicked)**

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 40 Update Product Menu**

* + If the **Update Product Menu is displayed and update product button is clicked**, **update the selected product’s price** **in the database** with the price that has been inputted in the TextField only if **the price is more than 0 and is not empty** (the Update Product Menu that this statement is referring to is shown in figure 40).
  + If the product is **updated successfully**, show an Information Alert. If **the update request is not successful**, show an Error Alert.
  + If the Remove Product is **clicked**, display:
    - Remove Confirmation Label
    - Remove Product Button
    - Back Button

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 41 Manage Product Form (a product is selected and remove button is clicked)**

* + If the **Remove Product Menu is displayed and Remove Product Button is clicked**, the selected product **will be removed from the database** (the Remove Product Menu that this statement is referring to is shown in figure 42).

A screenshot of a computer

Description automatically generated with low confidence

**Figure 42 Remove Product Menu**

* + If the product is **removed successfully**, show an Information Alert. If **the remove request is not successful**, show an Error Alert.
  + For **every update** that **has been made in the Product Table in the database**, the Form **will be reset** to the default layout that is shown in figure 34 and the data in the ListView **will be updated** to the newest data from the product table.
  + **All information about the product must be obtained from the product table in the database**.
* **Here are the credentials** that you can use for this case:
  1. **Admin Role**

Username: admin

Password: admin123

* 1. **User Role**

Username: stefanie

Password: 12345qwer