|  |  |
| --- | --- |
| **Project Case** |  |
| COMP7094 | COMP7094001  Multimedia Programming Foundation |
| **Computer Science** | **E242-COMP7094-DT01-00** |
| ***Valid on*** *Even 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 |

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, Asset File | - |

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*

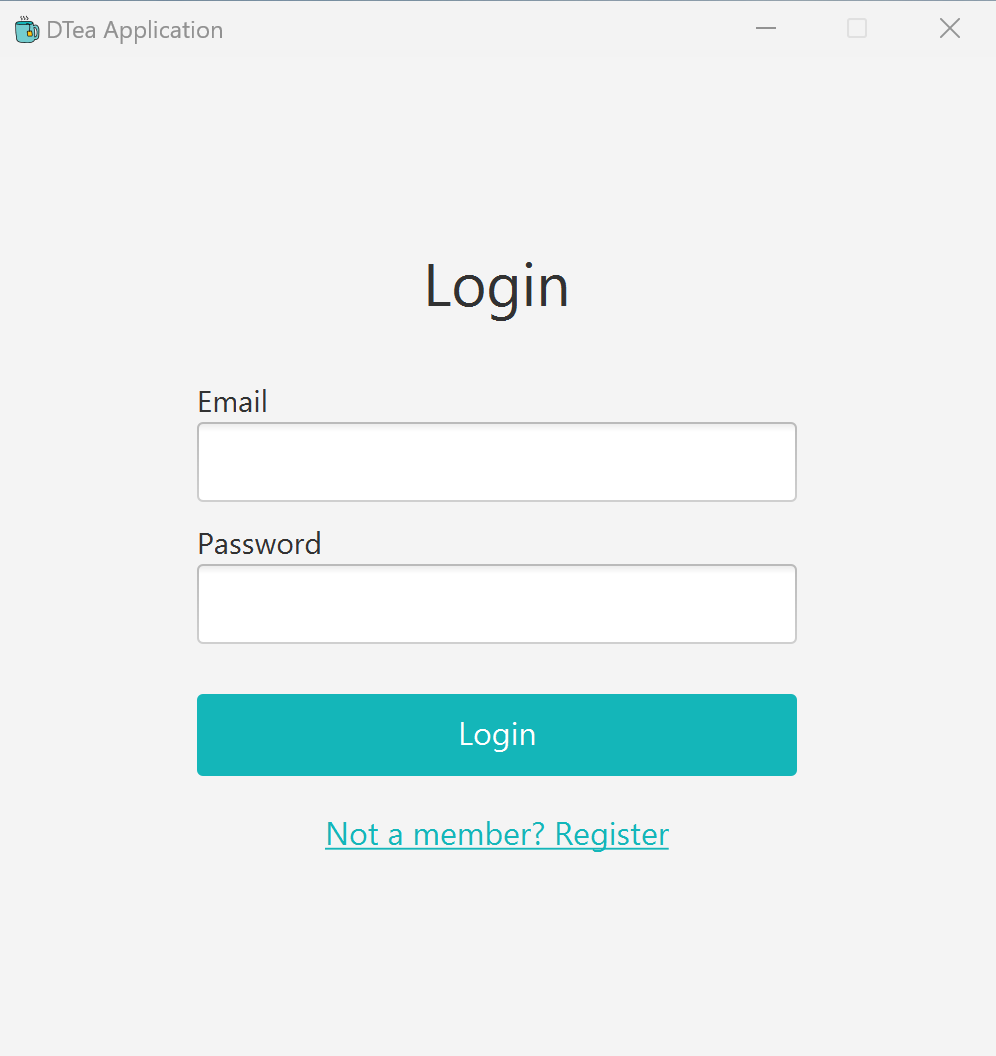
**DTea**

**DTea** is a comfy place to enjoy many kinds of tea in a calm and chill setting, making it the perfect spot to unwind and sip on your favorite brews. DTea is a newly opened coffee shop so they only sell **five kinds of tea**, which are shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Image File** | **Price** | **Stock** | **Description** |
| Lemon Tea | lemon\_tea.jpg | 20000 | 10 | A refreshing infusion of black tea subtly brightened by the zesty essence of fresh lemons, offering a tangy and invigorating flavor profile |
| Black Tea | black\_tea.jpg | 10000 | 20 | A robust and full-bodied brew, characterized by its deep, malty notes and a bold, satisfying taste that makes it a classic choice for tea enthusiasts |
| Apple Tea | apple\_tea.jpg | 25000 | 20 | A delightful infusion blending the sweetness of ripe apples with the comforting warmth of tea, creating a fruity and aromatic beverage that evokes the essence of a crisp autumn day |
| Honey Tea | honey\_tea.jpg | 30000 | 20 | A soothing concoction that combines the natural sweetness of honey with the mellow tones of tea, resulting in a comforting and mildly sweetened drink that is perfect for relaxation |
| Milk Tea | milk\_tea.jpg | 35000 | 0 | A rich and creamy fusion of tea and milk, offering a harmonious balance of bold tea flavors and the velvety smoothness of milk, creating a comforting and indulgent beverage enjoyed worldwide |

You are appointed to make an application using Java Programming to manage the user’s transaction and manage the products details.

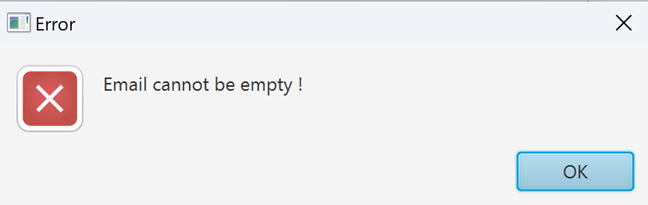
1. **Login Window**



**Figure 1. Login Window**

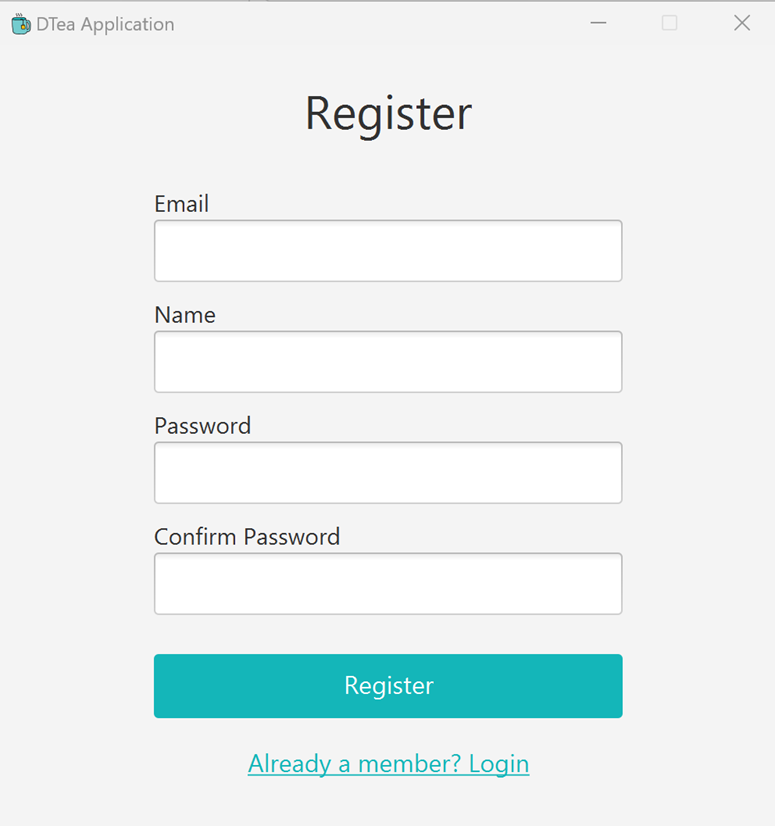
The application will **display** a **Login Window** with the following **components**:

* **Label** for the title, email and password.
* **TextField** to input the email.
* **PasswordField** to input the password.
* **Button** for the login button and register button.
  + **Login button**. The application will **validate** the form based on following requirements:
    - Validate **email** must be **filled**.
    - Validate **password** must be **filled**.
    - Validate **credential** (email & password) that user input is **valid**.
    - If the user input “**admin**” for the email and “**admin123**” for the password, the user will be logged in as admin. The application will **close the Login Window** and **open Admin Window**.
    - Other than that, the user will be logged in as **customer**. The application will **close the Login Window** and **open Welcome Customer Window**.
  + **Register button**. The application will **open** the **Register Window**.
* **Alert** for every invalid validation as shown below:



**Figure 2. Alert**

1. **Register Window**

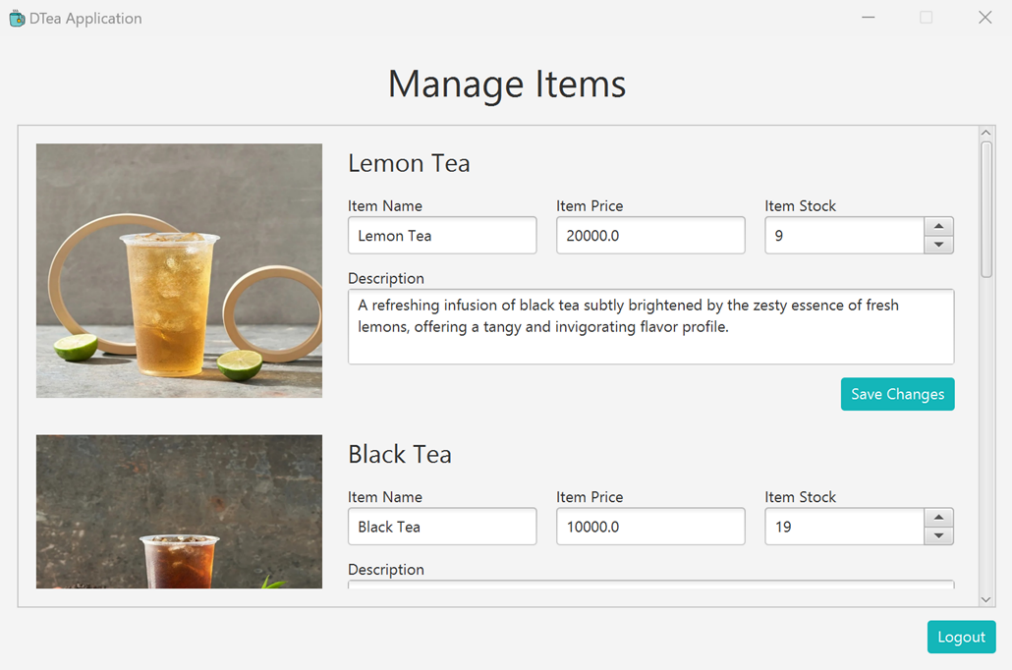


**Figure 3. Register Window**

The application will **display** a **Register Window** for register new user data with the following **components**:

* **Label** for the title, email, name, password, and confirm password.
* **TextField** to input the email and name.
* **PasswordField** to input the password and confirm password.
* **Button** for login button, and register button.
  + **Login button**. The application will **open** the **Login Window**.
  + **Register button**. The application will **validate** the form based on following requirements:
    - Validate **email** must be **filled**.
    - Validate **email** must **only have** **one** ‘**@**’.
    - Validate **email** must **end with** ‘**.com**’.
    - Validate **email** must be **unique**, have not been used for any registration before.
    - Validate **name** must be **filled**.
    - Validate **password** must be **filled**.
    - Validate **confirm password** must be **filled**.
    - Validate **confirm password** must be **matched** with the password.
    - After all validation **succeed**, the application will **save** the data in the system and **open** the **Login Window**.
* **Alert** for every invalid validation.

1. **Admin Window**



**Figure 4. Admin Window**

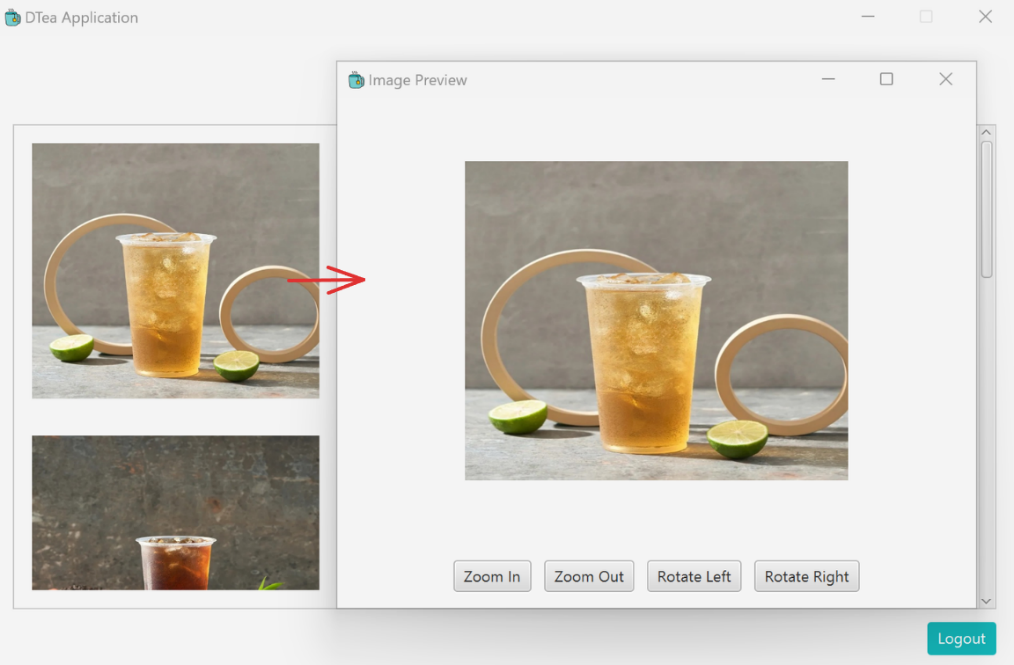
The application will **display** the **Admin Window** for the admin to update the item’s name, item’s price, item’s stock, and item’s description. The list of **component** are listed below:

* **Label** for the title, item name, item price, item stock, and item description.
* **ImageView** to display item’s image.
* **TextField** to input item’s name and item’s price.
* **TextArea** to input item’s description.
* **Spinner** to input item stock.
* **Button** for logout button and save changes button.

The **descriptions** for this window are listed below:

* **Display** the **images** for each item using **Image View**.
* **Display** all of the **current item’s name, item’s price, item’s stock, and item’s description** when the **Admin Window** is opened.
* Set the **Spinner** for each **item’s stock** with **minimum** value is **0**, **maximum** value is **1000**, and the **step** is **1**.
* If the user **clicks** the **save changes** **button**, the application will:
  + Validate **item’s name** must **be filled**.
  + Validate **item’s price** must **be numeric** (**int or float**).
  + Validate **item’s price** must **be at least 10000**.
  + Validate **item’s stock** must **be at least 1**.
  + Validate **item’s description** must **be at least 10 characters**.
  + After all validation **succeed**, the application will **update** the item data.
* **Alert** for every invalid validation.
* If the user **clicks** the **item’s** **image**, the application will **open** the **View Image Window** to show the selected item image.
* If the user **clicks** the **logout** **button**, the application will closethe Admin Window and **open** the **Login Window**.

1. **View Image Window**



**Figure 5. View Image Window**

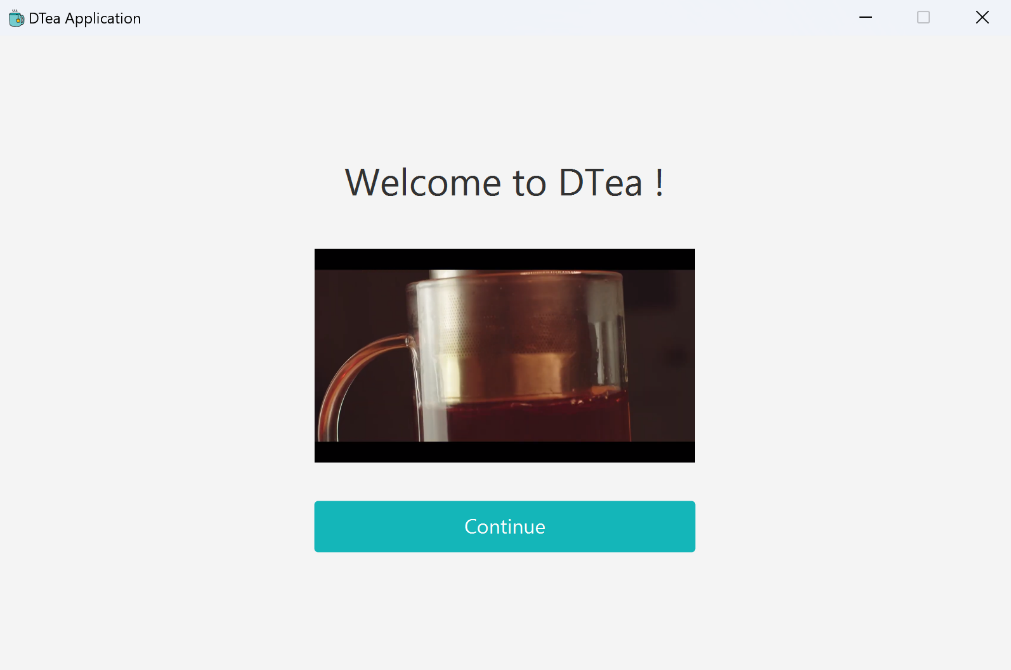
The application will **display** the **View Image Window** when admin clicked view image button from Master Window. The list of **component** are listed below:

* **Button** for zoom in button, zoom out button, rotate left button, and rotate right button.

The **description** for this window are listed below:

* **Display** the **image** of the item depends on what button clicked from Master Window.
* If the user **clicks** the **zoom in button**, the image will be **zoomed in** to **1.2 times** **larger** from its original scale.
* If the user **clicks** the **zoom out button**, the image will be **zoomed out** to **0.8 times** **smaller** than its original scale.
* If the user **clicks** the **rotate** **left button**, the image will be **rotated left** to **-90 degrees**.
* If the user **clicks** the **rotate right button**, the image will be **rotated right** to **90 degrees**.

1. **Welcome Customer Window**

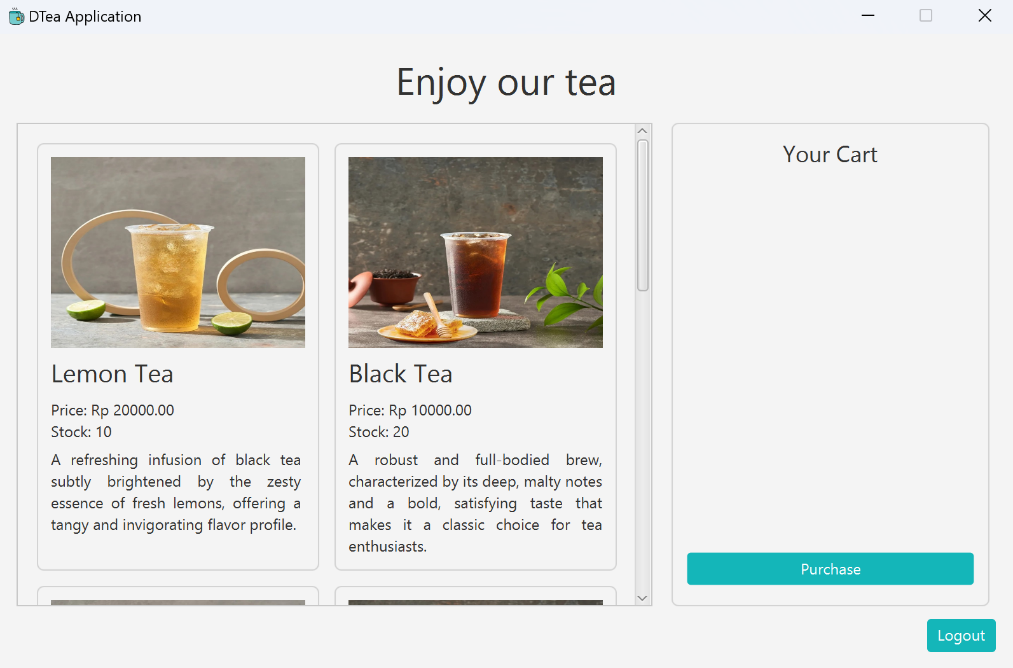


**Figure 6. Welcome Customer Window**

The application will **display** the **Welcome** **Customer Window** when users log in from **Login Window**. The list of **components** are listed below:

* **Label** for the title.
* **MediaView** to display promotional videousing **video.mp4** in the assets folder. The video must be **repeated** after the video ends.
* **Continue button**. The application will open the **Catalogue and Cart Window** and **stops the video from playing**.

1. **Catalogue and Cart Window**



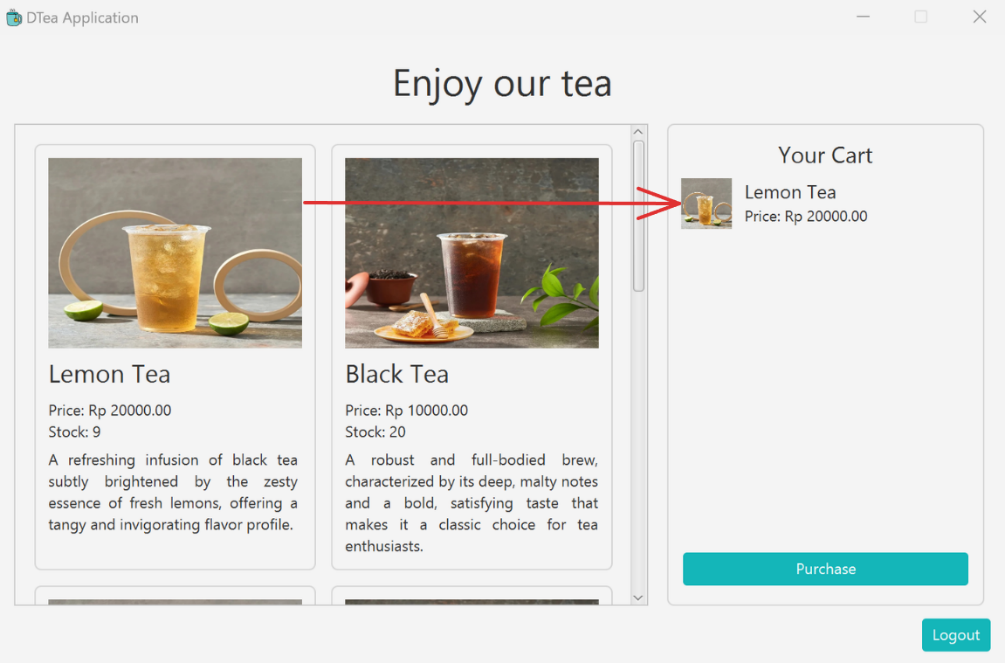
**Figure 7. Catalogue and Cart Window**

The application will **display** the **Catalogue and Cart Window** when users **clicked** **continue button** from **Welcome Customer Window**. The list of **component** are listed below:

* **Label** for the title, item’s name, item’s price, item’s stock, and item’s description.
* **Button** for purchase button, logout cart button.
* **ImageView** to display item’s image.
* **MediaPlayer** to play piano music.

The **descriptions** for this window are listed below:

* **Display** the logged on **user’s cart**. The user’s cart is saved in the system, so when the user re-login, the user will have the same cart.
* **Display** the **images** for each item using **Image View**.
* **Display** the **item’s name, item’s price**, **item’s stock**, **item’s description** based on the item list in the system.
* **Play** the **audio** using **piano.mp3** musicin the assets folder while users are on **Catalogue** and **Cart** window.
* Each item’s image on the catalogue section has the following effect when **dragged** by mouse:
  + - If the user **drags** an item inside the **cart**, then:
      * Validate if the **item’s stock** must be **greater than 0** and **not in the user’s cart**. Otherwise, the item cannot be added to the cart.
    - If the **item’s** **has** **stock and not in the user’s cart**, the application will:
      * **Add** the item to the **user’s cart**, r**educe** the item’s **quantity by 1** and **save** it to the system.



**Figure 8. Drag and drop illustration**

* If the user clicks the **Purchase** button, the application will **clear** the user’s cart.
* If the user **clicks** the **item’s** **image**, the application will **open** the **View Image Window** to show the selected item’s image.
* If the user **clicks** the **logout** **menu**, the application will closethe **Catalogue and Cart Window** and **open** the **Login Window**.
* **Alert** for every invalid validation.

**References:**

* + - https://food.grab.com/id/id/restaurant/teazzi-mall-taman-anggrek-delivery/6-C4AUV4BCREVBSA?
    - https://www.youtube.com/watch?v=KGvbeK\_A3QU
    - https://w1.pngwing.com/pngs/332/896/png-transparent-tea-yellow-tea-bag-teacup-japanese-tea-ceremony-drawing-drink-food-instant-tea.png