|  |  |
| --- | --- |
| **Project Case** | A logo for a software laboratory center  Description automatically generated with low confidence |
| ISYS6197 | ISYS6197003  Business Application Development |
| **Computer Science** | **O242-ISYS6197-CL06-00** |
| ***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*

**CLminton**

CLminton is a shop that sells various types of badminton rackets. CLminton wants to improve customer feelings by providing various types of quality rackets with various brands. To run this shop, the shop owner asks you to create a program to manage the course of transactions and products in the store. 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:

A screenshot of a computer

Description automatically generated with low confidence

**Figure 1. ERD**

Execute the provided create and insert query file on “**hO-Ohdie**” database! (“create+insert.sql”).

* + - 1. **Login Scene**

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

A screenshot of a computer

Description automatically generated

***Components:***

* **Label** for Login, Username and Password.
* **TextField** for Username input.
* **PasswordField** for Password input.
* **Button** for Login button.

**Description:**

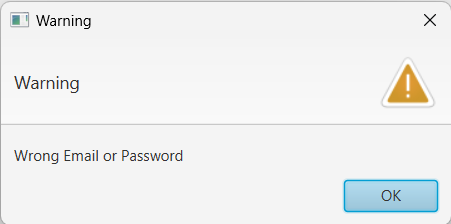
* **If user clicks Login button**, then **validate**:
  + **Username and Password must be filled**.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 3 If Email or Password is Empty**

* **Username and Password must exist and match** in database.



**Figure 4 If Email or Password is Wrong**

* The program will **validate role (user / admin)**.
* **If credential is correct**, direct the user to **Home Scene (if user role is “User”) or to Edit Product Scene (if user role is “Admin”).**
  + If user **failed to log in**, display an **Warning Alert.**

1. **Register Scene**

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

A screenshot of a computer

Description automatically generated

**Figure 5 Register Form**

***Components:***

* **Label** for Register, Email, Password, Confirm Password, Age, Gender, Nationality.
* **TextField** for Email input.
* **PasswordField** for Password, Confirm Password input.
* **RadioButton** for Gender (Male or Female) input.
* **ComboBox** for Nationality input.
* **Button** for Register button.
* **Menubar, Menu** for showing current scene (Register Scene)
* **MenuItem** for navigation to Login Scene
* **MenuItem** for navigation to Login Scene

**Description:**

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

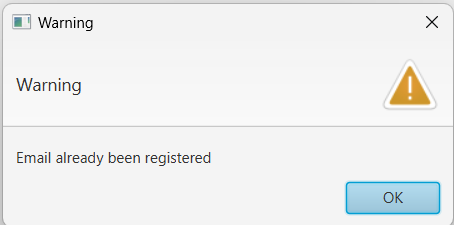
**UAXXX**

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

Example: UA001, UA002, UA003

**R**

* **Email must ends with ‘@gmail.com’**.
* **Username must be unique**
* **Password must contain minimum 6 characters**
* **Confirm Password must be the same as Password**.’
* **Age must be greater than 0**
* **Gender must be selected**
* **Nationality must be selected**
* **Every error message must be shown as Warning Alert**

****

**Figure 6 Email has already been registered**

* **If user successfully register**, insert a new user to the database with **“User” role** and the generated User ID. **Then direct the user to Login Scene**, to login with the new or existing user account.

1. **Login Register Navigation Bar**

This navigation bar is **only available on Login Scene and Register Scene**. It is **used to navigate between Login Scene and Register Scene**.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 7 Navigation Bar**

***Components:***

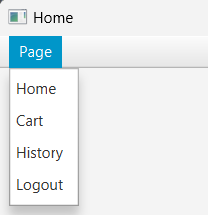
* **Menubar, Menu** for showing page to navigate
* **MenuItem** for navigation between Login and Register Scenes.

**Description:**

* **In the Login Scene**, the Register menu will direct the user to the Register Scene
* **In the Login Scene**, selecting the Login menu will not lead the user anywhere.
* **In the Register Scene**, the Login menu will direct the user to the Login Scene
* **In the Register Scene**, selecting the Register menu will not lead the user anywhere.

1. **Main Navigation Bar**

This navigation bar is available for all users. **Page menu for users with “User” Role**, and **Admin menu for users with “Admin” Role**. If user’s role is “User”, **Admin menu will be hidden** and cannot be accessed. If user’s role is “Admin”, **Page menu will be hidden** and cannot be accessed.



**Figure 8 User Menu**

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 9 Admin Menu**

***Components:***

* **MenuBar** for containing Page and Admin Menu.
* **“User” role:**
  + Menu for Page Menu.
* “**Admin” role:**
  + Menu for Admin Menu.
* **MenuItem** for Page and Admin Menu:
  + Logout (Account Menu)
  + Home (User Menu)
  + Cart (User Menu)
  + History (User Menu)
  + Manage Product (Admin Menu)
  + View History (Admin Menu)

**Description:**

* Logout is **used to logout** the current logged-in user’s account. **Direct the user to Login Scene** to login on a new or existing account.
* Home is **used to navigate to Home Scene**.
* Cart is **used to navigate to Cart Scene**,with **the current cart property** of the **current logged-in user.**
* History is **used to navigate to History Scene**, with **the transaction history of the current logged-in user.**
* Manage Product is **used to navigate to Manage Product Scene**.
* View History is **used to navigate to View History Scene**, with **the transaction history off all user**.

1. **Home Scene**

Home Scene is **only available for users with “User” Role**. In this scene, **user can view all products from the database** on a TableView. User can **click on one of the products** on the list to **show the details of the selected product**.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 10 Home Form (no selected item)**

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 11 Home Form (item selected)**

***Components:***

* **Label** for:
* Product List
* Product Detail :
* Product Name
* Product Brand
* Product Stock
* Product Price
* Total Price
* **TableView**, to display all products from database
* **Spinner** input Quantity
* **Button,** add to cart button.

**Description:**

* **Display ProductName**, **ProductBrand, ProductStock and ProductPrice** value for each of the products in the TableView.
* **Hide the product from user** if product stock is 0
* If the **user click a produc**t from the product list, **update the Home Scene to display details of the selected product.** The details consist of:
  + Product ID
  + Product Name
  + Product Price
  + Quantity Spinner (**Minimum Value: 1** and **Maximum Value : Product Stock**)
  + Total Price (**Product Price \* Quantity**)
  + Add to Cart Button
* If the **user did not select any of the product** from the product list and user try adding to cart show Warning Message.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 12 Product not selected warning alert**

* **Add to Cart** button:
  + **Add the selected product with the quantity slider value** to the current user’s cart. Data will be stored on the Cart table in the database and Reduce the selected product stock quantity equal to quantity slider value.
  + **Refresh the page** if the selected product is successfully added to cart in the database.

1. **Cart Scene**

Cart Scene is **only available to users with “User” Role**. In this scene, user will be able to **view all the products in current user’s cart** from database on a TableView. User **can click on one of the products** on the list **to show the details of the selected product**.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 13 Cart Scene (no selected item)**

A screenshot of a computer

Description automatically generated with medium confidence

***Figure 14 Cart Scene (item selected)***

***Components:***

* **Label** for:
  + Your Cart List
  + Cart’s Total Price
  + Product Detail:
    - Name
    - Brand
    - Price
* **TableView** for displaying all the products in the current user’s cart.
* **Button** for Remove from Cart and Checkout buttons.

**Description:**

* **Display ProductName, ProductBrand, ProductPrice, Quantity and Total Price Columns** on TableView.
* If the user **click a product** from the product list, update the Cart Scene to **display details of the selected product**. The details consist of:
  + Product Name
  + Product Brand
  + Product Price
* If the **user did not select any of the products** from the product list and **click delete product button** display Warning Message.

**A screenshot of a computer

Description automatically generated with medium confidence**

***Figure 15* Product not selected warning alert**

* **Remove from Cart button** will **remove the selected product from the current user’s cart database**, update the table to **display the remaining products in the current user's cart and update product stock to current stock + deleted product cart quantity.**

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 16 Cart Table (before removal)**

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 17 Cart Table (after removal)**

* **If user click Checkout** button:
  + **Validate**:
    - **If the cart is empty**, display an Warning Alert and cancel the operation.
    - **If the cart is not empty**, navigate to Checkout Cart Scene.

A screenshot of a computer

Description automatically generated with low confidence

**Figure 18 Cart is empty warning alert**

1. **Transaction Card Pop-up**

This pop-up window is **used determine Courier and Insurance**. This window **will pop-up when user want to checkout** current cart.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 19 Transaction Card Window**

***Components***:

* **Label** for :
* List Label
* List of Product name that user Checked out
* Courier Label
* Total Price from total product price
* **ComboBox** to determine which Courier user will choose.
* **CheckBox** to determine wether user want to use Insurance or not
* **JFXtras Window** for Transaction Card window.

**Description:**

* **If user selected Insurance Checkbox** add Total price by 90000
* **If user click checkout Button :**
* Show **Confrimation Alert**

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 20 Transaction Confirmation Alert**

* If user click ok :
* **TransactionID** is generated with format:

**THXXX**

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

Example: TH001, TH002, TH003

**R**

* **Clear the current user’s cart**.
* **Navigate user to Cart Scene.**
* If user click cancel :
* Navigate user to Cart Scene

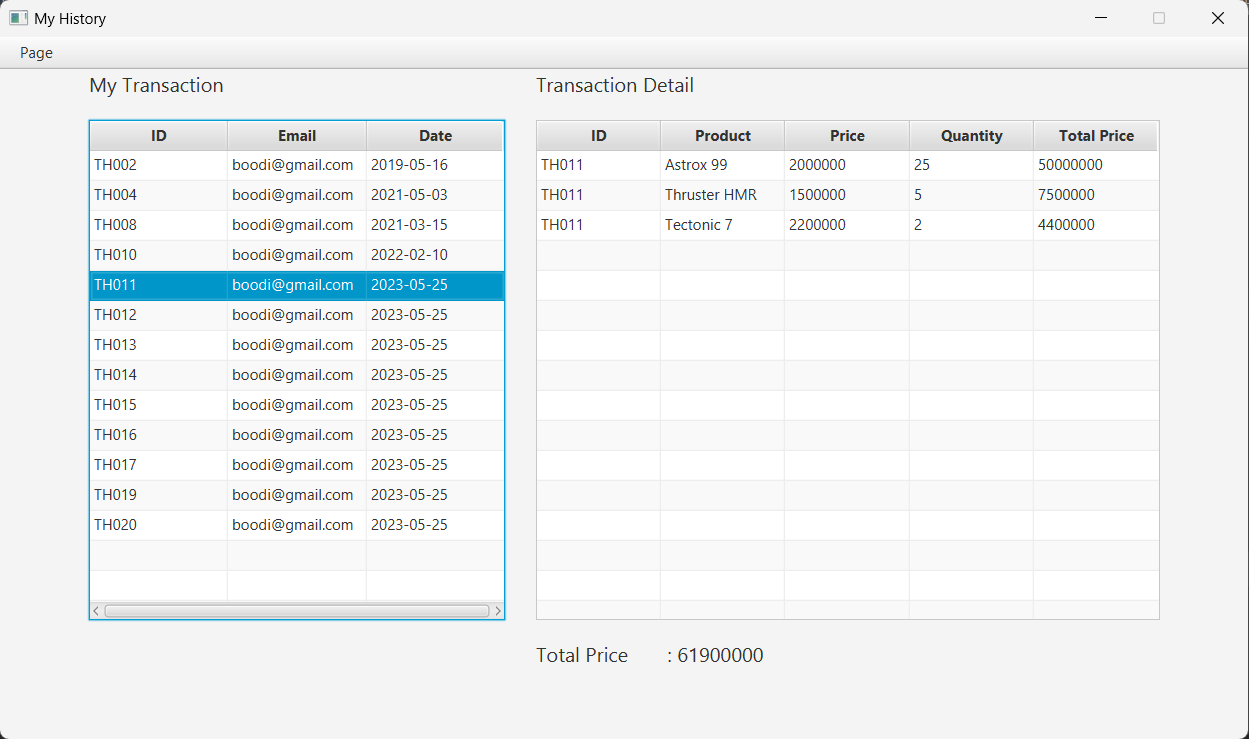
1. **History Scene**

History Scene is **used by users to see all transactions** that are done by the current user. All transactions will be displayed on a TableView. User can **click one of the transactions** in the table to **view the selected transaction’s details**.

A screenshot of a computer

Description automatically generated

**Figure 21 History Scene (no transaction selected)**



**Figure 22 History Scene (transaction selected)**

***Components:***

* **Label** for:
  + My Transaction
  + Transaction Detail
  + Total Price
* **TableView** for Transaction(s) and Transaction Detail(s) tables.

**Description:**

* Display **Transaction ID, User Email and Transaction Date columns** on Transaction table.
* If the user **click a transaction** from the Transaction table, update the History scene to **display details of the selected transaction** and **Total Price of the selected transaction**. **The details** consist of 5 columns:
  + Transaction ID
  + Product Name
  + Product Price
  + Quantity
  + Total Price (Product **Price \* Quantity**)
* **Total Price** is the **sum of all product’s total price in the selected transaction**.

1. **Manage Product Scene**

ManageProduct Scene **is only available for users with “Admin” Role**. In this scene, users with “Admin” Role can **view all products in the database** and have the authority to **insert a new product into the database, update a product stock, and delete a product from the database**.

A screenshot of a computer

Description automatically generated

**Figure 23 Manage Product Scene (no selected product)**

A screenshot of a computer

Description automatically generated with medium confidence**Figure 24 Manage Product Scene ( selected product)**

***Components:***

* **Label** for:
  + Product List
  + Add Product :
* Product Name
* Product Brand
* Product Price
  + Update & Delete Product:
    - Product Name
* **TextField** for:
  + Update Product
    - Stock input
  + Add Product:
    - Name and Price inputs
* **ComboBox** for selecting product brand.
* **Spinner** for Quantity product to add to the stock.
* **Button** for:
  + Update & Delete Product(s):
    - Update Stock button
    - Delete Product button
  + Add Product:
    - Add Product button
* **TableView** for showing all the products in the database.

**Description:**

* If the user **click a product** from the product list, update the Manage Product Scene to **display product name of the selected product**.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 25 Product Name Label**

* **Add Stock button** will **update the selected product’s stock** based on **Add Stock Spinner + Current Product Stock,** then update the product table.
* **Delete Product Button** will **delete the selected product’s**, then update the product table.
* **Add Product button:**
  + **add a new product** with a generated Product ID, Product Name based on Name input, Product Brand based by Brand selection and Product Price based on Price input.
  + **Product ID** is generated with format:

**PDXXX**

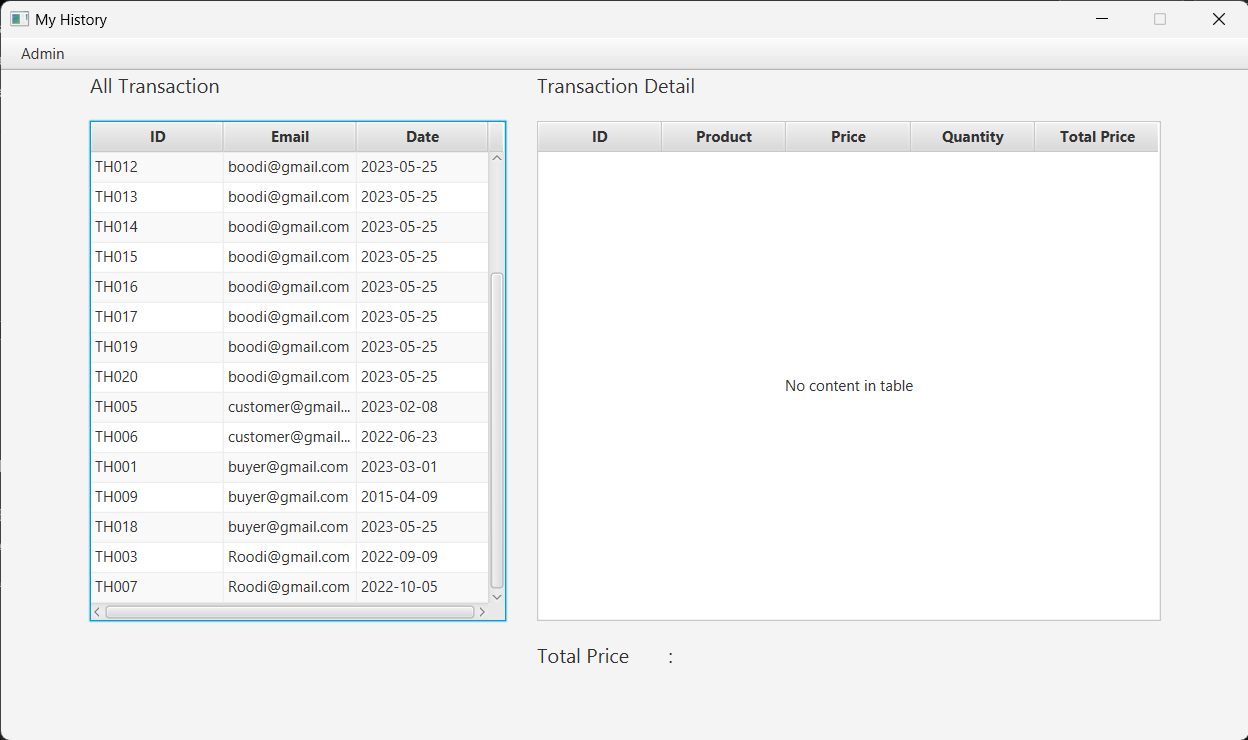
Where **X**: Digit (0-9) according to product index (plus by one on insert)

Example: PD001, PD002, PD003

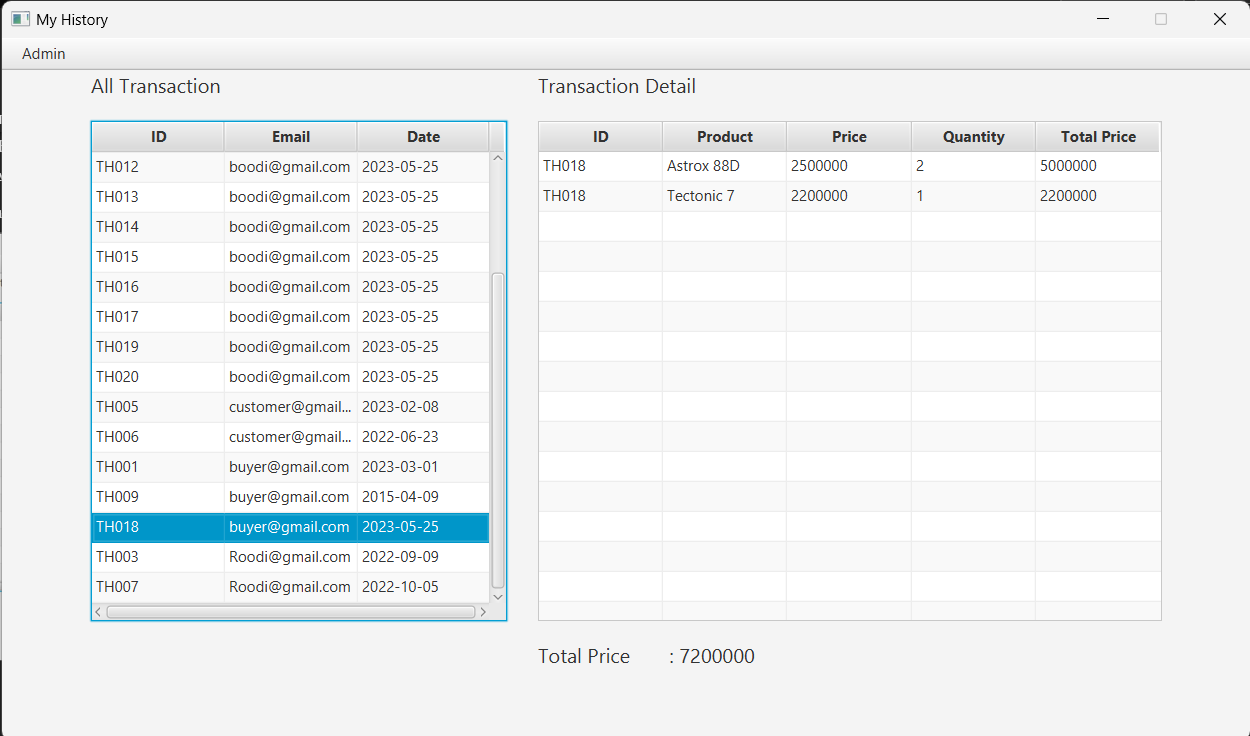
**R**

1. **View History Scene**

View History Scene is **used by admin to see all transactions** that are done by the all user. All transactions will be displayed on a TableView. admin can **click one of the transactions** in the table to **view the selected transaction’s details**.



**Figure 26 View History Scene (no transaction selected)**



**Figure 27 View History Scene (no transaction selected)**

***Components:***

* **Label** for:
  + All Transaction
  + Transaction Detail
  + Total Price
* **TableView** for Transaction(s) and Transaction Detail(s) tables.

**Description:**

* Display **Transaction ID, User Email and Transaction Date columns** on Transaction table.
* If the user **click a transaction** from the Transaction table, update the History scene to **display details of the selected transaction** and **Total Price of the selected transaction**. **The details** consist of 5 columns:
  + Transaction ID
  + Product Name
  + Product Price
  + Quantity
  + Total Price (Product **Price \* Quantity**)
* **Total Price** is the **sum of all product’s total price in the selected transaction**.
* **Here are the credentials** that you can use for this case:
  1. **Admin Role**

Username: admin@gmail.com

Password: admin1234

* 1. **User Role**

Username: boodi@gmail.com

Password: user1234