|  |  |
| --- | --- |
| **Project Case** |  |
| ISYS6197 | ISYS6197003  Business Application Development |
| **Computer Science** | **O242-ISYS6197-NJ05-00** |
| ***Valid on*** *Odd Semester Year 2023/2024* | **Revision 00** |

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 | - |

## Soal

*Case*

**NJuice**

NJuice is an innovative juice store powered by the cutting-edge JavaFX technology. To accommodate new customers, you are chosen to create an application for NJuice’s **customer** and **admin.** The program should be made using **JavaFX 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 picture containing text, screenshot, font

Description automatically generated

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

* **Landing Page Navigation Bar**

Landing Page Navigation Bar is a **Menu Bar**, with 1 **Menu** “Dashboard” and 2 **Menu Item** **“**Login” and “Register”. Landing Page Navigation Bar is used in Login Scene and Register Scene.

A picture containing screenshot, text, software, design

Description automatically generated

**Figure 1. Landing Page Navigation Bar**

* **Login Scene**

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

A screenshot of a computer

Description automatically generated

**Figure 2. Login Scene**

***Components***:

* + **Label** for Username, Password, and Error Label
  + **Text Field** for Username input
  + **Password Field** for Password input
  + **Button** for Login Button

**Description:**

* + If the 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** (user / admin)
    - If password and email matches:

Direct user to **Customer Scene** if the user is a customer or

Direct user to **Admin Scene** if the user is an admin.

* + If the user failed to login, display an **Error Label**.

**A screenshot of a computer

Description automatically generated**

**Figure 3. Error Label for Failed Credential**

* **Register Scene**

User can access register scene from the navigation bar (dashboard) **Menu Bar**. The **Register Form** is used by users to **Register**.

A screenshot of a computer

Description automatically generated

***Figure 4. Register Scene***

***Components***:

* + **Menu Bar** for Landing Pages Navigation Bar
  + **Label** for Username and Password
  + **Text Field** for Username input
  + **Password Field** for Password input
  + **Checkbox** for Agreeing terms and conditions.
  + **Button** for Register Button

**Description:**

* + If the user clicks **Login** button, then validate:
    - **Username**, **Password**,and **Terms and Conditions** must be filled in.
    - **Username** is **not already taken** in the database.
* If the user successfully registers, the program will **insert a new user account to the database** as a customer. Then direct the user to **Login Scene**.

**CUXXX**

Where **X**: Last inserted number plus by one

Example: CU001, CU002, CU003

* + If the user failed to register, display an **Error Label**.

**A screenshot of a computer

Description automatically generated**

***Figure 5. Error for input field***

**A screenshot of a computer

Description automatically generated**

***Figure 6. Error Label for not Unique Username***

* **Customer Navigation Bar**

Customer navigation bar is a toolbar that is used in **Customer Home Scene** and **Customer Checkout** **Scene**.



***Figure 7. Error Label for not Unique Username***

***Components:***

* + **Toolbar** to hold the navigation bar.
  + **Button** for logout
  + **Label** for customer greeting (“hi, [username]”)
* **Customer Home**

User is redirected to customer home scene if the user role is a **Customer**. In this scene, user can **access** their cart, **delete** items in their cart, **add new item** to their cart, and **checkout all items** in their cart.

A screenshot of a computer

Description automatically generated

***Figure 8. Customer Home Scene***

***Components***:

* + **List** for user’s Cart
  + **Label** for Title and Price
  + **Text Field** for Username input
  + **Button** for Add new item, Delete item, and Checkout.

**Description:**

* + Initially the list is empty, so show the empty message.
  + If the user clicks **Add new Item** button, then pops up the **Add New Item Scene.**
  + If the user clicks **Delete Item,** then validate:
* **An item** in list is **selected**.

If the validation succeeds, proceed to **Delete that item** from the user’s cart.

* + If the user failed to delete an item, display an **Error Alert**.
  + If the user clicks **Checkout cart**, then validate:
* **User’s** **cart** is **not empty.**

If the validation succeeds, direct the user to the **Checkout Scene**.

* + If the user failed to delete an item, display an **Error Alert**.
  + Don’t forget to **refresh the scene** when changes happen so the result is shown immediately.

**A screenshot of a computer

Description automatically generated with medium confidence  
*Figure 9. Customer Home Scene after the user filled their cart***

A screenshot of a computer

Description automatically generated with medium confidence  
***Figure 10. Error Alert if no juice is selected for delete.***

*A screenshot of a computer

Description automatically generated with medium confidence*

***Figure 11. Error Alert if no juice in cart for checkout.***

* **Customer Add New Item Scene**

In this scene users can **add a new item to their cart**. This scene is a new **pop-up window** that is separated from the main window. Customer add new item scene doesn’t have the customer navigation bar.

A screenshot of a computer

Description automatically generated

***Figure 12. Customer add new item pop-up***

***Components***:

* + **Window** for the pop up with the title “Add new item”.
  + **Combo Box** for juices option, Combo box options are all **Product Name** that exists in the database.
  + **Label** for Option, Price, Quantity, and Subtotal
  + **Spinner** for Quantity input
  + **Button** for Add New Item Button.

**Description:**

* + If the user **chooses** **an** **item** from the combo box, the **product** **information** will be shown.
  + If the user **changes** the **quantity** **spinner**, the **total** **price** will also change.
  + If the user clicks **Add New Item,** then validate:
* **An item** in combo box is **selected**.
* **The quantity** spinneris not below one.
* **If the product** is already in the user’s cart.
  + If **the product is not in the user’s cart**, then **insert a new item** to their cart.
  + If **the product is in user's cart,** then **add the quantity** to that item in their cart.

**A screenshot of a computer menu

Description automatically generated with medium confidence**

***Figure 13. Item Combo Box Options depends on the product in the database***

**A screenshot of a computer

Description automatically generated**

***Figure 14. After user choose an option, price, description and total price will be updated***

**A screenshot of a computer

Description automatically generated**

***Figure 15. If the quantity is changed, the total price will also change.***

* **Customer Checkout Scene**

In this scene users can **view** items, subtotal, and total price in their cart before they check out all items in their cart. User must also specify which payment type is used in this transaction.

A screenshot of a checkout

Description automatically generated

***Figure 16 Checkout scene.***

***Components***:

* + **Label** for Title, Items and Subtotal, and Total Price.
  + **Radio Button** for Payment Type option
  + **Button** for Cancel Button and Checkout Button.

**Description:**

* + If the user clicks **Checkout Button,** then validate:
* **A payment type** radio button is selected.
  + If the validation succeeds:
* **Insert** new **transaction** to the database by generating Id:

**TRXXX**

Where **X**: Last inserted number plus by one

Example:TR001, TR002, TR003

* **Clear** **user’s cart**.
* Direct user back to customer’s home page
  + If the user clicks **Cancel Button,** then direct user back to customer's home

A screenshot of a computer

Description automatically generated with low confidence

***Figure 17. Error message if payment is not selected.***

A screenshot of a computer

Description automatically generated with medium confidence

***Figure 18. Information messsage after successfully checkout all item.***

* **Admin Navigation Bar**

Admin navigation bar is a menu bar that is used in **Admin View Transactions Scene** and **Manage Juice** **Scene.**

A picture containing text, screenshot, font, software

Description automatically generated

***Figure 19. Admin Navigation Bar option if the menu dashboard is selected.***

A picture containing text, screenshot

Description automatically generated

***Figure 20. Admin Navigation Bar option if the menu logout is selected.***

***Components:***

* + **Menu Bar** to hold the navigation bar.
  + **Menu Admin’s Dashboard** that consists of **View Transaction** and **Manage Juice**
  + **Menu Logout** that consists of **Logout from admin,** If clicked direct user to login scene.
* **Admin View Transaction Scene**

Admin View Transaction Scene appears when admin logged in and can be accessed from the **Admin’s Navigation Bar**. In this scene, admin can view past transactions made when **user checked out their cart**. Admin can see the **transaction details** by **clicking through various transactions**.

A screenshot of a computer

Description automatically generated with medium confidence

***Figure 21. Admin View Transaction Scene.***

***Components***:

* + **Table View** for Transaction Table (TransactionId, JuiceId, PaymentType) and Transaction Detail Table (TransactionId, JuiceName, JuiceId, Quantity)
  + **Label** for Title.

**Description:**

* If the user clicks **a row in** the transaction table, then **show** **details (TransactionId, JuiceName, JuiceId, Quantity)** from that transaction in the **transaction** **detail** table.

A screenshot of a computer

Description automatically generated with medium confidence

***Figure 21. If the transaction is clicked, the detail appears in the detail table.***

* **Admin Manage Products Scene**

Admin Manage Juice Scene can be accessed through the **Admin’s Navigation Bar**. In this scene, admin can **view**, **insert**, **update**, and **delete** products in NJuice’s catalog.

A screenshot of a computer

Description automatically generated with medium confidence

***Figure 22. Manage Products Scene.***

***Components***:

* + **Table View** for Product Table (ProductId, Product Name, Product Price, Product Description)
  + **Label** for Title, Product Id, Product Name, Product Price, and Product Description.
  + **Combo Box** for Product Id input.
  + **Text Field** for Product Name input.
  + **Spinner** for Product Price input.
  + **Text Area** for Product Description input.
  + **Buttons** for Add Product Button, Update Product Button, and Delete Product Button.

**Description:**

* + If the user clicks **Insert Product,** then validate:
* **Product Name and Product Description** must be filled in.
* **The price** input is not below 10000.
* **Description** input length is not below 10 and over 100.
  + After the validation successfully passed, **insert** the new juice into the database with generate id.

**JUXXX**

Where **X**: Last inserted number plus by one

Example: JU001, JU002, JU003

* + If the ser clicks **Update Product,** then validate:
* **The price** input is not below 10000.
* **A juice id** in the combo box is selected.
  + After the validation successfully passed, **update** the price of that juice in the database.
  + If the user clicks **Delete Product,** then validate:
* **A juice id** in the combo box is selected.
  + After the validation successfully passed, **delete** the juice from the database.
  + Don’t forget to **refresh the scene** so that the effect will be shown immediately after updating, deleting, and inserting new data.

**A screenshot of a computer

Description automatically generated with medium confidence**

***Figure 23. Error message if the data needed is not sufficient.***

**A screenshot of a computer

Description automatically generated with medium confidence**

***Figure 24. Product Id in the combo box must matches with juices id in the database.***

**A screenshot of a computer

Description automatically generated with medium confidence**

***Figure 24. Table example after a new juice is inserted.***

1. Here are the credentials that you can use for this case:
   1. Role admin

Username: admin

Password: admin123

* 1. Role User

Email: [customer](mailto:boodi@gmail.com)

Password: user123