|  |  |
| --- | --- |
| **Project Case** |  |
| ISYS6197 | ISYS6197003  Business Application Development |
| **Computer Science** | **O242-ISYS6197-OO01-00** |
| ***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 8u291  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*

**hO-Ohdie**

hO-Ohdie is a growing store that sells various high-quality hoodies. hO-Ohdie want to provide a better customer experience by making hO-Ohdie available digitally. To make this happen, the owner asked you to create an application to manage hO-Ohdie’s products 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:

A diagram of a server

Description automatically generated

**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

***Figure 2 Login Form***

***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 exist and match** in database.

A screenshot of a computer

Description automatically generated with low confidence

**Figure 3 If credential doesn't exist or 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 **Error Alert.**

1. **Register Scene**

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

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 4 Register Form**

***Components:***

* **Label** for Register, Email, Username, Password, Confirm Password, Phone Number, Gender, Address.
* **TextField** for Email, Username, Phone Number input.
* **PasswordField** for Password, Confirm Password input.
* **RadioButton** for Gender (Male or Female) input.
* **TextArea** for Address input.
* **CheckBox** for agreeing to Term & Conditions
* **Button** for Register button.
* **Menubar, Menu** for showing current scene (Register Scene)
* **MenuItem** for navigation to Login Scene

**Description:**

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

**USXXX**

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

Example: US001, US002, US003

**R**

* **Email must ends with ‘@hoohdie.com’**.
* **Username must be unique**
* **Password must contain minimal 5 characters**
* **Confirm Password must be the same as Password**.
* **Phone Number length must be 14 characters and starts with ‘+62’.**
* **Gender must be selected**
* **Address must be filled**
* **CheckBox must be checked**
* **Every error message must be shown as Error Alert**

A screenshot of a computer

Description automatically generated with low confidence

**Figure 5 Username already taken error**

* **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 low confidence

**Figure 6 Login Scene Navigation Bar**

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 7 Register Scene Navigation Bar**

***Components:***

* **Menubar, Menu** for showing current scene (Login or Register)
* **MenuItem** for navigation between Login and Register Scenes.

**Description:**

* **In Login Scene**, Register menu will **direct the user to Register Scene.**
* **In Register Scene**, Login menu will **direct the user to Login Scene.**

1. **Main Navigation Bar**

This navigation bar is available for all users. It contains of Account menu, **User 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”, **User menu will be hidden** and cannot be accessed.

A screenshot of a computer

Description automatically generated

**Figure 8 Account Menu (User)**

***Components:***

* **MenuBar** for containing Account, User, and Admin Menu.
* **“User” role:**
  + Menu for Account and User Menus.
* “**Admin” role:**
  + Menu for Account and Admin Menus
* **MenuItem** for Account, User, and Admin Menu:
  + Logout (Account Menu)
  + Home (User Menu)
  + Cart (User Menu)
  + History (User Menu)
  + Edit Product (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.**
* Edit Product is **used to navigate to Edit Product Scene**.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 9 User Menu**

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 10 Admin Menu**

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 ListView. 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 11 Home Form (no selected item)**

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 12 Home Form (item selected)**

***Components:***

* **Label** for:
  + hO-Ohdie
  + Hoodie’s Detail:
    - Hoodie ID
    - Name
    - Price
    - Quantity
    - Total Price.
* **ListView** for displaying all products from database.
* **Spinner** for Quantity input.
* **Buttton** for Add to Cart button.

**Description:**

* **Display HoodieID and HoodieName** value for each of the products in the ListView.
* 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:
  + Hoodie ID
  + Hoodie Name
  + Hoodie Price
  + Quantity Slider (**Minimal Value: 1**)
  + Total Price (**Hoodie Price \* Quantity**)
  + Add to Cart Button
* If the **user did not select any of the product** from the product list, update the Home Scene to **hide the details of product**.
* **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.
  + **Display a Success Alert** if the selected product is **successfully added to the cart** in database.

A screenshot of a computer

Description automatically generated with low confidence

**Figure 13 Add to Cart Success Alert**

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 14 Cart Scene (no selected item)**

A screenshot of a computer

Description automatically generated with medium confidence

***Figure 15 Cart Scene (item selected)***

***Components:***

* **Label** for:
  + [‘Username’]’s Cart
  + Hoodie’s Detail:
    - Hoodie ID
    - Name
    - Price
    - Quantity
    - Total Price
  + Contact Information:
    - Email
    - Phone
    - Address
  + Cart’s Total Price
* **TableView** for displaying all the products in the current user’s cart.
* **Button** for Remove from Cart and Checkout buttons.

**Description:**

* **Display Hoodie ID, Hoodie Name, 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:
  + Hoodie ID
  + Hoodie Name
  + Hoodie Price
  + Quantity
  + Total Price (Hoodie Price \* Quantity)
  + Remove from Cart Button
* If the **user did not select any of the products** from the product list, update the Cart scene **to hide the details of product**.
* **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.**

A screenshot of a computer

Description automatically generated

**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 Error Alert and cancel the operation.
    - **If the cart is not empty**, display a Payment Confirmation Pop-up window.

A picture containing text, screenshot, font, number

Description automatically generated

**Figure 18 Cart is empty error alert**

1. **Payment Confirmation Pop-up**

This pop-up window is **used to confirm payment by users**. This window **will pop-up when user want to checkout** current cart.

A screenshot of a computer screen

Description automatically generated with low confidence

**Figure 19 Payment Confirmation Window**

***Components***:

* **Label** for confirmation text.
* **Button** for Make Payment and Cancel buttons.
* **JFXtras Window** for Payment Confirmation window.

**Description:**

* **Make Payment button**:
  + Complete the checkout operation by **inserting a new Transaction into the database.**
  + **TransactionID** is generated with format:

**TRXXX**

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

Example: TR001, TR002, TR003

**R**

* **Clear the current user’s cart**.
* **If successful**, display a Success Alert, then direct the user to History Scene.
* **Cancel button** will cancel the checkout operation.

A screenshot of a computer

Description automatically generated with low confidence

**Figure 20 Transaction Success Alert**

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)**

Screens screenshot of a computer screen

Description automatically generated with low confidence

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

***Components:***

* **Label** for:
  + [‘Username’]’s Transaction(s)
  + [‘Transaction ID’]’s Transaction Detail(s)
  + Total Price
* **TableView** for Transaction(s) and Transaction Detail(s) tables.

**Description:**

* Display **Transaction ID and User ID columns** on Transaction(s) 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
  + Hoodie ID
  + Hoodie Name
  + Quantity
  + Total Price (**Hoodie Price \* Quantity**)
* **If the user did not select any of the transactions** from the Transaction table, update the History scene to hide the details and Total Price of the transaction.
* **Total Price** is the **sum of all hoodie’s total price in the selected transaction**.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 23 History Scene (empty)**

1. **Edit Product Scene**

Edit Product 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 price, and delete a product from the database**.

A screenshot of a computer

Description automatically generated

**Figure 24 Edit Product Scene (no selected product)**

A screenshot of a computer

Description automatically generated with low confidence

**Figure 25 Edit Product Scene (product selected)**

***Components:***

* **Label** for:
  + Edit Product
  + Update & Delete Hoodie(s):
    - Hoodie ID
    - Name
    - Price
  + Insert Hoodie:
    - Name
    - Price
* **TextField** for:
  + Update & Delete Hoodie(s):
    - Price input
  + Insert Hoodie:
    - Name and Price inputs
* **Button** for:
  + Update & Delete Hoodie(s):
    - Update Price button
    - Delete Hoodie button
  + Insert Hoodie:
    - Insert button
* **TableView** for showing all the products in the database.

**Description:**

* If the user **click a product** from the product list, update the Edit Product Scene to **display details of the selected product**. **The details** consist of:
  + Hoodie ID
  + Hoodie Name
  + Price TextField
  + Update Price and Delete Hoodie Buttons

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 26 Price TextField**

* **Price TextField**:
  + Price TextField value is based on selected hoodie’s price.
  + Price TextField value can be changed to update hoodie’s price with the Update Price button.
* **Update Price button** will **update the selected product’s price** based on **Price TextField value,** then update the product table.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 27 Test Hoodie’s Price (after update)**

* **Delete Hoodie button** will **delete the selected hoodie from the database**, then update the product table.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 28 Edit Product Scene (after deletion)**

* **If the user did not select any of the products from the product list**, update the Edit Product Scene to **hide the details of product**.
* **Insert button:**
  + **Insert a new product** with a generated Hoodie ID, Hoodie Name based on Name input, and Hoodie Price based on Price input.
  + **Hoodie ID** is generated with format:

**HOXXX**

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

Example: HO001, HO002, HO003

**R**

* Name TextField and Price TextField **will be cleared** and update the product table.

A screenshot of a computer

Description automatically generated with low confidence

**Figure 29 Insert Hoodie example**

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 30 Edit Product Scene (after insertion)**

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

Username: admin

Password: admin

* 1. **User Role**

Username: dummy

Password: dummy