|  |  |
| --- | --- |
| Assignment Case | Description: LogoBINUS-University |
| JavaH1Special |
| **Periode Berlaku** Semester Ganjil 2023/2024  ***Valid on*** *Odd Year 2023/2024* | **Software Laboratory Center**  **Assistant Recruitment 24-1** |

## Materi

*Material*

* Object Oriented Programming (OOP)
* SOLID Principle
* MySQL

## Soal

*Case*

**VIennese Café**

**AX For AutoPets** is an engaging online auto battler game where players strategically assemble and manage teams of diverse animals, each with unique abilities, to compete in automated battles. This innovative game aims to provide a captivating experience, combining strategy and fun as players build their ultimate teams to outsmart and outlast their opponents. The development of **AX For AutoPets** will utilize the **Java Programming Language** with **Object Oriented Programming** concepts such as **Encapsulation**, **Inheritance**, and **Polymorphism**, ensuring a robust and flexible gaming experience. Your mission as a programmer is to create this exciting game, focusing on seamless gameplay, balanced mechanics, and an enjoyable user experience.

* **Database ERD** :

**Sebuah gambar berisi teks, cuplikan layar, Font, nomor

Description automatically generated**

Figure 1. Database ERD

* **Main Page**
* In the beginning, the program will show **2 menus**, which are :

1. **Login**
2. **Register**
3. **Exit**

Sebuah gambar berisi teks, Font, cuplikan layar, putih

Description automatically generated

Figure 2. Main Page

* If the user chooses **Login (Menu 1)**  then :
  + The program will ask the user to input the **username** and **password**.
  + Then, **validate** the **credentials** must **exist** in the **database**.
  + If the **credential** **does not exist**, then **show an error message**.
  + **Otherwise**, **redirect** the user **to menu based on their role**.



Figure 3. Login

* If the user choose **menu 2 (“Register”),** then :
  + Ask the user to input the **username**. Validate that the username **length** must be **between 5 and 15** **characters** **(inclusive)**.
  + Ask the user to input the **password**. Validate that the password **length** must be **more or equal to 6** **characters**.
  + Ask the user to input the **role**. Validate that the role **must be either “Head Chef” or “Waiter” (case sensitive)**.
  + **Generate** the **User ID**, based on this format :

**USXXX  
X ( random number 0 – 9 )**

* + If the **validation** is **successful**, then **insert the new user into the database**.

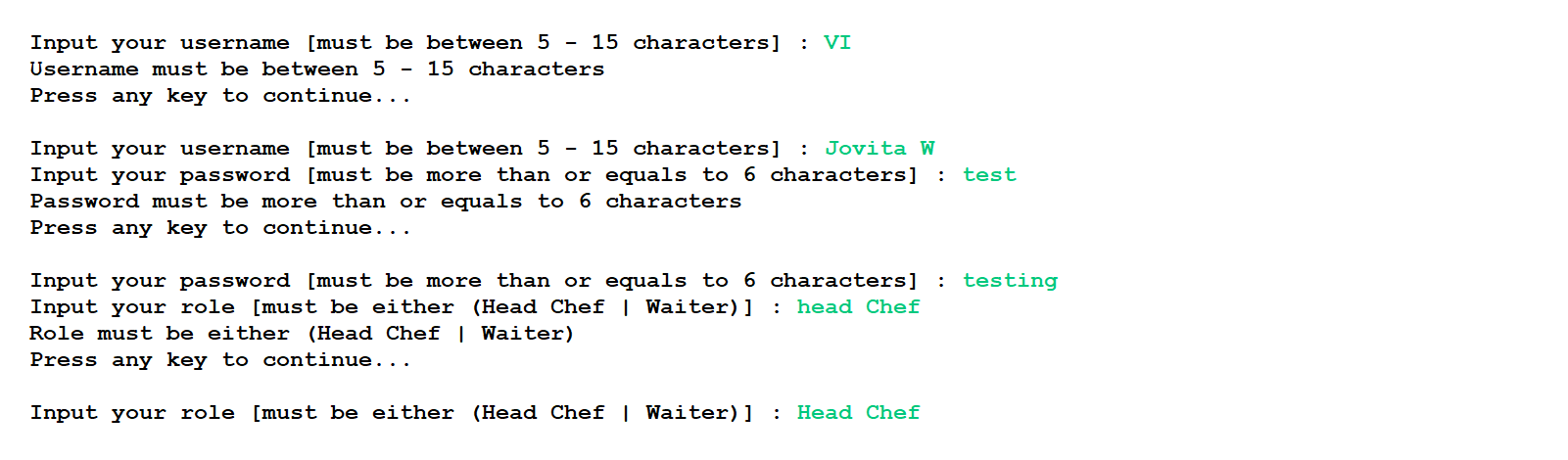


Figure 4. Register New User

* If the user choose **menu 3 (“Exit”)**, then the **program will be closed.**
* **Head Chef Menu**
* In this menu consist of **6 menus**, which are :

1. **Add Menu**
2. **View All Menu**
3. **View Ready Menu**
4. **Delete Menu**
5. **Prepare Food and Drinks**
6. **Exit**



Figure 5. Head Chef Menu

* If the user chooses **menu 1 (“Add Menu”)**, then :
  + Ask the user to input the **menu.** Validate that the menu **length must be more or equal to 3 characters**.
  + Ask the user to input **price.** Validate that the price **must be more than or equal to 1000**.
  + Ask the user to inpu**t the type of the menu.** Validate that the type **must be either of “Food”, “Drink”, or “Soup” (case sensitive)**.
  + **Generate** the **Menu ID,** based on this format :

**MNXXX  
X ( random number 0 – 9 )**

* + Validate that the **Menu ID must be unique**.
  + After the **validation** is **successful,** then **insert the new menu into the database**.

**Sebuah gambar berisi teks, Font, cuplikan layar

Description automatically generated**

Figure 6. Insert New Menu

* If the user chooses **menu 2 (“View All Menu”)**, then :
  + If the list of the **menu** from the databas**e is empty, then show an empty message**.
  + **Otherwise, display all menu** data that consist of **Menu ID, Menu, and Price**.

**Sebuah gambar berisi teks, cuplikan layar, Font, nomor

Description automatically generated**

Figure 7. Display All Menu

* If the user chooses menu 3 (“View Ready Menu”), then :
  + If the **Array List / Vector** **is empty**, **then show an empty message**.
  + **Otherwise**, **display all** the **ready menu** from the Array List / Vector. The ready menu data is **consisting of Menu ID, Menu, and Price**.

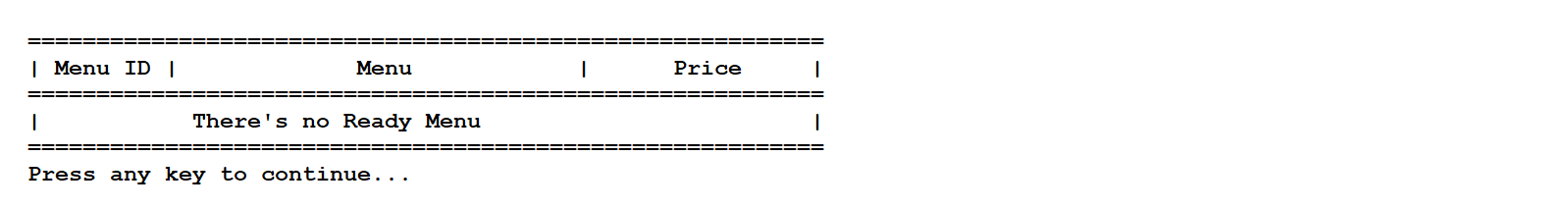


Figure 8. Empty Message

Sebuah gambar berisi teks, cuplikan layar, Font, garis

Description automatically generated

Figure 9. Display Ready Menu

* If the user chooses **menu 4 (“Delete Menu”)**, then :
  + **Display** **all the menu** from the database.
  + Ask the user to input the **Menu ID** that want to be deleted. Validate that the **Menu ID must be exists**.
  + If the **Menu ID doesn’t exist**, **then show an error message**.
  + **Otherwise**, **delete** the Menu ID **from the database**.

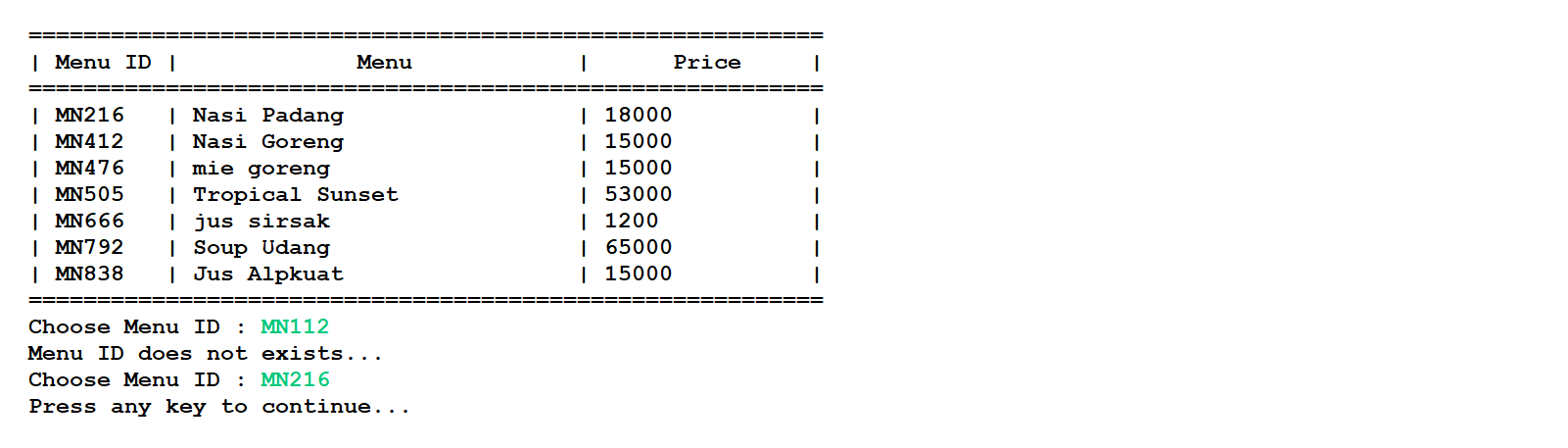


Figure 10. Delete Menu

* If the user **chooses menu 5 (“Prepare Food and Drinks”)**, then **redirect** user to **Prepare Menu**.
* If the user **chooses menu 6 (“Exit”)**, then **redirect** user to **Main Menu**.
* Prepare Menu
* In this menu consist of **7 menus**, which are :

1. **Prepare Food**
2. **Prepare Drink**
3. **Prepare Soup**
4. **Prepare All Food**
5. **Prepare All Drink**
6. **Prepare All Soup**
7. **Exit**

Sebuah gambar berisi teks, cuplikan layar, Font, putih

Description automatically generated

Figure 11. Prepare Menu

* If the user chooses **menu 1 (“Prepare Food”)**, then :
  + **Display** **All Order** data from the **Array List / Vector**. The order data **consists of Order ID, Menu ID, and Menu**. Validate the **Order** only for the **type ‘Food’**.
  + Ask the user to input the **Order ID**. Validate that the Order ID **must be exist**.
  + After the **validation** is **successful**, then :
    - If the **Menu ID is exist** from Array List / Vector of **Ready Menu**, then :
      * **Delete** the Menu ID **from** the **Array List / Vector of Need to Prepare**.
      * **Set** the **Order Status to “Done”**.
      * **Decrease** the Restaurant **Food Supply** by **20 points**.
      * Validate that the **Food Supply must be more or equal to 20 points**.
    - **Otherwise**, then **show an error message**.

Sebuah gambar berisi teks, cuplikan layar, Font, garis

Description automatically generated

Figure 12. Empty Message

Sebuah gambar berisi teks, Font, garis, cuplikan layar

Description automatically generated

Figure 13. Prepare Food

* If the user chooses **menu 2 (“Prepare Drink”)**, then :
  + **Display** **All Order** data from the **Array List / Vector**. The order data **consists of Order ID, Menu ID, and Menu**. Validate the **Order** only for the **type ‘Drink’**.
  + Ask the user to input the **Order ID**. Validate that the Order ID **must be exist**.
  + After the **validation** is **successful**, then :
    - If the **Menu ID is exist** from Array List / Vector of **Ready Menu**, then :
      * **Delete** the Menu ID **from** the **Array List / Vector of Need to Prepare**.
      * **Set** the **Order Status to “Done”**.
      * **Decrease** the Restaurant **Water Supply** by **20 points**.
      * Validate that the **Water Supply must be more or equal to 20 points**.
* **Otherwise**, then **show an error message.**

Sebuah gambar berisi teks, cuplikan layar, Font, garis

Description automatically generated

Figure 14. Empty Message

Sebuah gambar berisi teks, Font, garis, cuplikan layar

Description automatically generated

Figure 15. Prepare Drink

* If the user chooses menu 3 (“Prepare Soup”), then :
  + **Display** **All Order** data from the **Array List / Vector**. The order data **consists of Order ID, Menu ID, and Menu**. Validate the **Order** only for the **type ‘Soup’**.
  + Ask the user to input the **Order ID**. Validate that the Order ID **must be exist**.
  + After the **validation** is **successful**, then :
    - If the **Menu ID is exist** from Array List / Vector of **Ready Menu**, then :
      * **Delete** the Menu ID **from** the **Array List / Vector of Need to Prepare**.
      * **Set** the **Order Status to “Done”**.
      * **Decrease** the Restaurant **Water Supply** by **20 points**.
      * **Decrease** the Restaurant **Food Supply** by **20 points**.
      * Validate that the **Water Supply or Food Supply must be more or equal to 20 points**.
* **Otherwise**, then **show an error message.**

Sebuah gambar berisi teks, cuplikan layar, Font, garis

Description automatically generated

Figure 16. Empty Message

Sebuah gambar berisi teks, Font, cuplikan layar

Description automatically generated

Figure 17. Prepare Soup

* If the user chooses menu 4 (“Prepare All Food”), then :
  + **Display** **All Order** data from the **Array List / Vector**. The order data **consists of Order ID, Menu ID, and Menu**. Validate the **Order** only for the **type ‘Food’**.
  + Ask the user to input the **confirmation**. Validate the confirmation **must be either ‘Y’ or ‘N’ (case sensitive)**.
  + If the **confirmation** is equal to **‘Y’**, then :
    - If the **Menu ID is exist** from Array List / Vector of **Ready Menu**, then :
      * **Delete** **each** Menu ID **from** the **Array List / Vector** of items that **Need to Prepare**.
      * **Set** **each** of the **Order Status to “Done”**.
      * **Decrease** the Restaurant **Food Supply** by **10 points** for **each Order ID**.
      * Validate that the **Food Supply must be more or equal to 10 points**.
  + If the **confirmation** is equal to **‘N’**, then **redirect** the user tothe **previous menu.**
  + **Otherwise**, then **show an error message.**

Sebuah gambar berisi teks, cuplikan layar, Font, garis

Description automatically generated

Figure 18. Cancel Prepare All Food

Sebuah gambar berisi teks, Font, garis, nomor

Description automatically generated

Figure 19. Prepare All Food

* If the user chooses menu 5 (“Prepare All Drink”), then :
  + **Display** **All Order** data from the **Array List / Vector**. The order data **consists of Order ID, Menu ID, and Menu**. Validate the **Order** only for the **type ‘Drink’**.
  + Ask the user to input the **confirmation**. Validate the confirmation **must be either ‘Y’ or ‘N’ (case sensitive)**.
  + If the **confirmation** is equal to **‘Y’**, then :
    - If the **Menu ID is exist** from Array List / Vector of **Ready Menu**, then :
      * **Delete** **each** Menu ID **from** the **Array List / Vector** of items that **Needs to be Prepared**.
      * **Set** **each** of the **Order Status to “Done”**.
      * **Decrease** the Restaurant **Water Supply** by **10 points** for **each Order ID**.
      * Validate that the **Water Supply must be more or equal to 10 points**.
  + If the **confirmation** is equal to **‘N’**, then **redirect** the user tothe **previous menu.**
  + **Otherwise**, then **show an error message.**

Sebuah gambar berisi teks, cuplikan layar, Font, garis

Description automatically generated

Figure 20. Cancel Prepare All Drink

Sebuah gambar berisi teks, cuplikan layar, Font, nomor

Description automatically generated

Figure 21. Prepare All Drink

* If the user chooses menu 6 (“Prepare All Soup”), then :
  + **Display** **All Order** data from the **Array List / Vector**. The order data **consists of Order ID, Menu ID, and Menu**. Validate the **Order** only for the **type ‘Soup’**.
  + Ask the user to input the **confirmation**. Validate the confirmation **must be either ‘Y’ or ‘N’ (case sensitive)**.
  + If the **confirmation** is equal to **‘Y’**, then :
    - If the **Menu ID is exist** from Array List / Vector of **Ready Menu**, then :
      * **Delete** **each** Menu ID **from** the **Array List / Vector** of items that **Needs to be Prepared**.
      * **Set** **each** of the **Order Status to “Done”**.
      * **Decrease** the Restaurant **Food Supply** by **10 points** for **each Order ID**.
      * **Decrease** the Restaurant **Water Supply** by **10 points** for **each Order ID**.
      * Validate that the **Food Supply or Water Supply must be more or equal to 10 points**.
  + If the **confirmation** is equal to **‘N’**, then **redirect** the user tothe **previous menu.**
  + **Otherwise**, then **show an error message.**

Sebuah gambar berisi teks, Font, garis, cuplikan layar

Description automatically generated

Figure 22. Cancel Prepare All Soup

Sebuah gambar berisi teks, cuplikan layar, Font, garis

Description automatically generated

Figure 23. Prepare All Soup

* If the user chooses **menu 7 (“Exit”)**, then **redirect** the user to the **previous menu**.
* Waiter Menu
* Waiter menu consist of **4 menus**, which are :
  + - 1. **Add Order**
      2. **View Order**
      3. **View Ready Menu**
      4. **Exit**



Figure 24. Waiter Menu

* If the user **chooses menu 1 (“Add Order”)**, then :
  + Ask the user to input the **customer’s** **name**. Validate that the customer’s name **must be filled**.
  + Determine one **menu** by **random** from the **list menu**.
  + Ask the user to input the **quantity**. Validate that the quantity **must be between 1 – 3 (inclusive)**.
  + After the **validation** is **successfully**, then **add** the **new order into database**.

Sebuah gambar berisi teks, cuplikan layar, Font, putih

Description automatically generated

Figure 25. Insert New Order

* If the user **chooses menu 2 (“View Order”)**, then :
  + If the **Array List / Vector** of the Order is **empty**, then **show an empty message**.
  + **Otherwise**, **display all the order** data that **consist of Customer Name, Order ID, Menu Name, Quantity, Price, Total Price, Order Status**.
  + **Calculate** the **total price** by this formula :

**Total Price = Quantity \* Price**

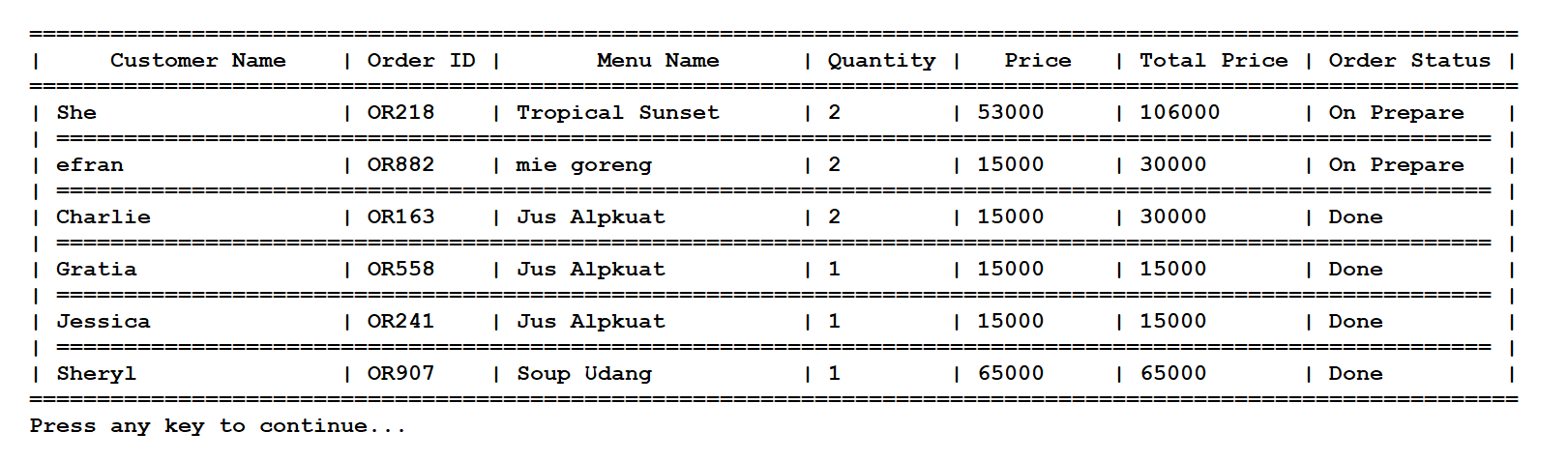


Figure 26. Display All Order

* If the user **chooses menu 3 (“View Ready Menu”)**, then :
  + If the **Array List / Vector** of the Ready Menu is **empty**, then **show an error message**.
  + Otherwise, **display** **all** **the ready menu** data that **consist of Menu ID, Menu, and Price**.

Sebuah gambar berisi teks, cuplikan layar, Font, garis

Description automatically generated

Figure 27. Empty Message

Sebuah gambar berisi teks, cuplikan layar, Font, garis

Description automatically generated

Figure 28. Display Ready Menu

Please run the BAT or JAR file to see the sample program.

## Komponen Penilaian

*Scoring Component*

|  |  |  |
| --- | --- | --- |
| No | Component | Weight |
| 1 | OOP Concept | 15 |
| 2 | Design Pattern | 5 |
| 3 | CRUD | 20 |
| 4 | SOLID Principle | 30 |
| 5 | Architectural Design | 8 |
| 6 | Actions | 12 |
| 7 | Validations | 10 |