|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Practicum Final Exam – Even Semester Year 2021/2022** | | | | | | |
| **Subject** | | | **COMP6708001 – Object Oriented Programming** | | |  |
| **Class** | **:** | **B201 / B301 / BC02** | | **Start Date** | **: 17 June 2022** |
| **Lecturer** | **:** | **D3442 - Herru Darmadi, S.Kom., M.TI**  **D4689 - Rulyna, S.Kom., M.M.** | | **Start Time** | **: 07:20** |
| **End Date** | **: 17 June 2022** |
| **End Time** | **: 09:20** |

**PERATURAN UJIAN:**

*Exam Regulations:*

* Mahasiswa tidak diperbolehkan berdiskusi dan/atau bekerja sama dengan peserta ujian lainnya

*Student is not allowed to discuss and/or work together with other exam participants*

* Mahasiswa tidak diperbolehkan untuk membuka dan menyalin dari **BUKU** atau **CATATAN**, **VIDEO** dari pengajar (recording kelas, VBL, Youtube, dsb) dan **REFERENSI** lainnya

*Student isn't allowed to open and copy from any resources such as notes, videos (class recording, VBL, Youtube, etc) and other references*

* Mahasiswa tidak diperbolehkan membuka dan menyalin jawaban dari internet (google, stackoverflow, dsb)

*Student isn't allowed to open and copy answer from the internet (google, stackoverflow, etc)*

* Asisten **BERHAK** memberi nilai 0 **(NOL)** bagi peserta ujian yang melakukan segala bentuk kecurangan

*Assistant is able to give 0 (ZERO) score for exam participant who does any cheating actions*

* Kumpulkan jawaban tepat pada waktunya, apabila terlambat mengumpulkan maka jawaban tidak akan dikoreksi dan nilai mahasiswa adalah 0

*Submit the answer on time, if not, then the answer will not be checked, and the students will receive 0 (ZERO)*

* Bila Anda tidak membaca peraturan ini, maka Anda dianggap telah membaca dan menyetujuinya

*If you have missed to read these regulations, so you are considered to have read and agreed on it*



**SOFTWARE YANG DIGUNAKAN:**

*Software will be used:*

* Eclipse 2020.6
* Java 8
* MySQL Java Connection Library 8.0.24
* XAMPP 8.0.7

**FILE YANG DIKUMPULKAN:**

*File must be collected:*

* JAVA
* CLASS
* SQL

**PERHATIAN!**

*Attention!*

* Bagi yang mengerjakan tidak sesuai dengan soal, maka akan diberikan nilai **NOL (0)**

*For those who do not work in accordance with the exam case will be marked as* ***ZERO (0)***

* Bagi yang mengerjakan tidak sesuai dengan software dan versi yang telah ditetapkan, maka akan tetap dikoreksi dengan software dan versi yang telah ditetapkan

*For those who do not work in accordance with the software and specific version will be corrected by the predefined software and version*

* Kompres semua jawaban yang akan diunggah. Pastikan format pengumpulan nama file dan ekstensi sesuai dengan format berikut: **[NIM]-[NAMA].zip**

*Compress all file that will be uploaded. Make sure the format for collecting file name and extension according to the following format:* ***[NIM]-[NAME].zip***

**Soal**

*Case*

**Bakeroo**

**Criteria** :

1. **Database Operation**

Perform database operation from given **sql file** (“**bakeroo.sql**”) in Java application

1. **Abstract Class**

Design at least **three** classes, consisting of **one abstract** class and **two concrete** class. **Abstract** class consists of all common attributes and behaviour that both of the concrete class had. **Concrete** class consist of specific attribute and behaviour that is not common between the concrete classes

1. **Encapsulation**

Attributes of the class must be **encapsulated** in order to prevent **illegal** direct access, and can only be accessed using **accessor** and **mutator** that may perform validation before accessing the encapsulated attribute

1. **Inheritance**

Concrete classes **must inherit all** attributes and behaviours from the abstract class

1. **Polymorphism**

Concrete class may **override** behaviour inherited from abstract class according to their **specific implementation**

**Bakeroo** is a well-known bakery that established by Mr. BakBak. Given massive growth development, he asked his acquaintance Mr. WillBoy to develop a program using **Java** as the programming language and **MySQL** for its storage data. Because of your expertise, Mr. WillBoy assigned the project to you. To help you with the programming, Mr. WillBoy has given the detail and requirements for the program, which is shown below :

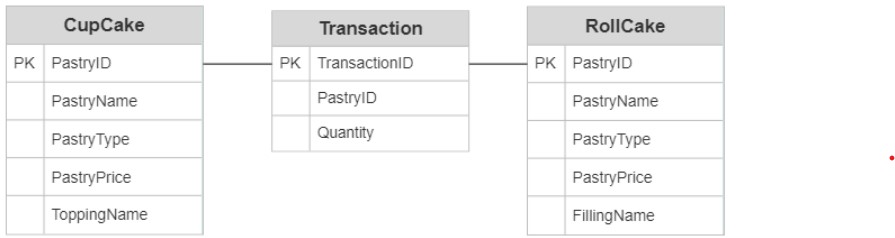


Figure 1. Entity Relationship Diagram (ERD)

|  |  |
| --- | --- |
| **Connection String** | **jdbc:mysql://localhost:3306/bakeroo** |
| **Username** | **root** |
| **Password** |  |
| **Driver Class** | **com.mysql.cj.jdbc.Driver** |

The **default port number is 3306**. If you have **changed the port number for MySQL in your XAMPP** then feel free to **change the 3306 into your port number**.

* At the start of program, there will be 4 menus :

1. **Insert Transaction**
2. **View Transaction**
3. **Delete Transaction**
4. **Exit**

Background pattern

Description automatically generated with low confidence

Figure 2. Main Menu

* If user choose menu 1 (“**Insert Transaction**”), then the program will :
* Show **all items data from the database, whether it’s a “cupcake”** or **“rollcake”**. After that ask the user to **choose the item from the list**.

Table

Description automatically generated

Figure 3. Display Item

* Askthe user to input the **number of list they want to buy**. **Validate** the input must be **between 1** – **total existing cake(s).**



Figure 4. Input Index

* Ask the user to input the **quantity**. **Validate** the quantity **must be bigger than 0.**



Figure 5. Input quantity

* Define **pastry’s discount** based on this following formula

|  |  |
| --- | --- |
| **Cup Cake** | **Roll Cake** |
| 15000 | 25000 |

* **Show the details** of the user’s transaction with the **total price** obtained from the following formula.

|  |  |
| --- | --- |
| **Cup Cake** | **Roll Cake** |
| Total Price = (Pastry Price \* Quantity) - Discount | Total Price = (Pastry Price \* Quantity) - Discount |

Graphical user interface

Description automatically generated with low confidence

Figure 6. Detail Transaction

* **Then insert the transaction data into the database** with the **following id format** and **redirect the user back to the main menu.**

|  |
| --- |
| **PIXXX**  **Where:**  **X:** **Number (0-9) that is increment from the last transaction**  **Example: PI001, PI002, PI003** |

* If user choose menu 2 (“**View Transaction**”), then the program will :
* **Retrieve** transaction data from the database
* If data is **empty**, **display** the following message



Figure 7. View Transaction (Empty)

* Otherwise, **display** all transaction details that was obtained from the database.

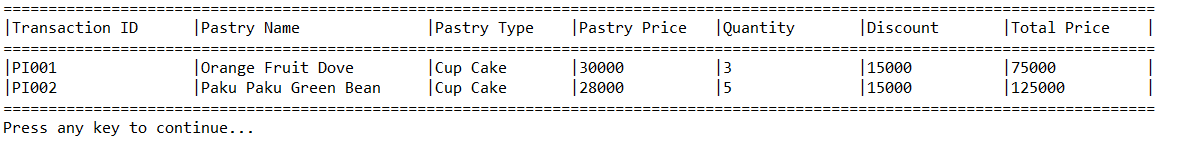


Figure 8. View Transaction

* If user choose menu 3 (“**Delete Transaction**”), then the program will :
* **Retrieve** transaction data from the database.
* If data is **empty**, **display** the following message:



Figure 9. View Transaction for delete (empty)

* Otherwise, **display** all transaction details that was obtained from the database.

Table

Description automatically generated with medium confidence

Figure 10. View Transaction for delete

* Ask the user to input the **transaction ID**. **Validate** the transaction ID must be **exist on the database (case insensitive)**



Figure 11. Input Transaction ID

* **Finally**, **delete the transaction** that was chosen from the database. Then **show a success message**.



Figure 12. Success Message

* If user choose menu 4 (“**Exit**”), then the program will close and display an exit message



Figure 13. Exit Message