|  |  |
| --- | --- |
| **Project Case** |  |
| COMP6115  Object Oriented Analysis & Design |
| **Computer Science** | **E212-COMP6115-CV02-00** |
| ***Valid on*** *Odd Semester Year 2021/2022* | **Revision 00** |

1. Seluruh mahasiswa tidak diperkenankan untuk:

*The whole student is not allowed to:*

* + 1. Melihat sebagian atau seluruh proyek mahasiswa lain,

*Seeing a part or the whole project from another students.*

* + 1. Menyadur sebagian maupun seluruh proyek dari buku,

*Adapted a part or the whole project from the book*

* + 1. Mendownload sebagian maupun seluruh proyek dari internet,

*Downloading a part or the whole project from the internet,*

* + 1. Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal proyek,

*Working with another theme which is not in accordance with the existing theme in the matter of the project,*

* + 1. Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + 1. Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika kelompok terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai kelompok** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the group is proved to the actions described in point 1 above, the score of the group which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan proyek, segala jenis pengumpulan proyek di luar jadwal tidak dilayani.

*Pay attention to the submission schedule for the project, all kinds of submission outside the project schedule will not be accepted*

1. 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*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| 40% | 60% | - |

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

|  |
| --- |
| **Software**  *Software* |
| Java 8  Eclipse 2020.6  XAMPP 7.4.7  MySQL Java Connection Library 5.1.49 |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri, dan proyek untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment and project collection for this subject are described as follows:*

|  |
| --- |
| **Proyek**  *Project* |
| JAVA, CLASS, SQL |

**Coffee Vibes**

Coffee Vibes is a local cafe located near Bina Nusantara. Coffee Vibe’s owner, Cha, asked you as a professional System Analyst, to help him develop a system to manage their business activity. Next is the business flow to be covered by the system.

* **Barista**

In Coffee Vibes, **every transaction** will be recorded after the barista **checked out the products** in the cafe’s database. Before checking out, all the products must be stored in a **temporary object called cart**. Barista is the only employee who can **add, view, and remove products from the cart**. The cart may **not contain** duplicate products, if the barista adds an already **existing product** in the cart, the **product quantity** **in the cart** will be updated instead of creating a new product in the cart.

When checking out, **display the total price** from the cart’s products. Barista can also apply a discount voucher and **vouchers can only be used once**. If the voucher is used, then **update the new total price** according to the applied discount. After checking out, a new transaction object is created and **will be recorded** in the database. Then **decrease the product’s stock** in the database according to the purchased item. Do not forget to **update voucher status** if used.

* **Product Admin**

Product Admin can do two types of things. First Product Admin can **insert, view, update, and delete products from database**. Other than that, Product Admin can also **generate, view, and delete voucher from database**.

|  |  |
| --- | --- |
| Product | Validation |
| Product ID | * Cannot be empty * Must be unique * Cannot be changed |
| Product Name | * Cannot be empty |
| Product Description | * Cannot be empty |
| Product Price | * Cannot be empty * Must be numeric * Cannot be less than one |
| Product Stock | * Cannot be empty * Must be numeric * Cannot be less than zero |
| Voucher | Validation |
| Voucher ID | * Cannot be null * Must be exists * Cannot be updated |
| Voucher Discount | * Cannot be empty * Must be numeric * Must be between 1 and 100 |
| Voucher Status | * Cannot be empty |

* **Manager**

As the manager of Coffee Vibes, the manager can **view all recorded transaction header in the databas**e. The manager can also **view the transaction details** that contain list of products that was added along with the transaction header when checking out. To view the transaction detail the manager can **click transaction header cell** and the system will then show all transaction details product in the transaction header. Other than viewing transaction, the manager can also **view and fire employee** that is stored in the database.

|  |  |
| --- | --- |
| Transaction Header | Validation |
| Transaction ID | * Cannot be empty * Must be unique * Cannot be changed |
| Transaction Purchase Date | * Cannot be empty |
| Transaction Voucher ID | * Cannot be empty * Must be numeric |
| Transaction Total Price | * Cannot be empty * Must be numeric * Cannot be less than one |
| Transaction List Transaction Item | * Cannot be empty |

|  |  |
| --- | --- |
| Transaction Detail | Validation |
| Transaction ID | * Foreign key to Transaction Header ID |
| Transaction Detail Product ID | * Cannot be empty * Must be numeric |
| Transaction Detail Quantity | * Cannot be empty * Must be numeric * Cannot be less than zero |

* **Human Resource Department**

Human Resource Employee is responsible to handle all the other employee’s data. They can **insert, view, update and delete employee’s data** that is stored in the database.

|  |  |
| --- | --- |
| Employee | Validation |
| Employee ID | * Cannot be empty * Must be unique |
| Employee Position ID | * Cannot be empty * Position ID must be existed |
| Employee Name | * Cannot be empty |
| Employee Status | * Cannot be empty |
| Employee Salary | * Cannot be empty * Must be numeric * Cannot lower than one |
| Employee Username | * Cannot be empty |
| Employee Password | * Cannot be empty |

|  |  |
| --- | --- |
| Position | Validation |
| Position ID | * Cannot be empty * Must be unique |
| Position Name | * Cannot be empty |

As a system analyst, to help you design the program Cha provides you with the design of the program based on the requirement and the business flow above. You can open “**ProjectOOAD.vpp**” provided to view the diagram.

As a programmer, Cha wants you to develop the application with the following requirement:

* The application must be build using Java-based Programming with **MVC** (**Model View Controller**) architecture for a better development process
* Model

The model layer is responsible for **representing concepts** in the business or information about the business situation. Besides that, model layer also responsible for **giving access to the database** via its public interfaces to acquiring and manipulating references to pre-existing domain objects.

* View

View layer, or Presentation Layer, is responsible for showing information to the user and interpreting the user's commands. This layer is the home for **all user interfaces** in the project.

* Controller

This layer is responsible to **validate** all input from the view layer and **all business logics** are implemented in the controller layer. It also responsible for **delegating request**s from the user to the lower layer for further processing.

* The database must be using **MySQL**
* The application must have an **authenticated user based on position**
* The application must minimalize human error with great user experiences
* The application must be made based on the analysis diagram that provided by Cha, but you may add additional components in the system based on your assumption
* Documentation for the application:
* The guide for using the application
* Any additional assumption that you make to develop the program