Versions

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | In charge | Description |
| 25-October 22 | V1.0 | Ngoc Tran, A Nguyen | Update Section I, II and Appendix A |
|  |  |  |  |

Abbreviation

DMS Database Management System

… …

List of Figures

Figure 1 The accuracy between X and Y 10

List of Tables

Table 1: A comparison between X and Y 10

Table of Contents

I…. 3

II… 4

III…. 5

*(Removing this italic red paragraph below after you read it)*

*Student please use font* ***Times New Roman****, size* ***13****,* ***Justify Text Alignment*** *for the report content.*

*If you insert the code into the report, please keep its original format with font, size and colors.*

*Please insert only the code useful for the report in clarification, and limit lengthening the report by pasting unnecessary code. You can insert the path of the code files, and I will search it in your attached project code to check them.*

1. **INTRODUCTION**

*In this project, we investigate the business functions of the hotel management system, and provide a desktop application supporting users.*

*This project provides the basic functions for the stakeholders/users as follows:*

* *Input the customer information.*
* *Input the booking.*
* *Search customers*
* *Display the booking details.*
* *…*

*<Students should select around 10 functions for the projects including 3 input, 3 search, 3 output functions, …. Certainly, students can code more functions if students want and get some bonus on those.>*

*In this project, we use Java OOP for designing classes and applying the OOP such as encapsulation, inheritance, … [You can update this section during the regular request and submission, and make it complete for the final submissions ]*

1. **CLASS ANALYSIS**
2. *Objects*

|  |  |  |  |
| --- | --- | --- | --- |
| *No* | *Object Name* | *State* | *Behaviours* |
| *1* | *Manager 1* | *Anna, 30, ID: M001, Salary: 2000$, bonus:500$* | *isReceiving(),*  *isChecking()* |
| *2* | *Manager 1* | *Frank, 50, ID: M002, Salary: 2000$, bonus:500$* | *isReceiving(),*  *isChecking()* |
| *3* | *Employee 1* | *John, 35, ID: E001, Salary: 1000$, bonus:300$* | *isPicking(),*  *isShipping(),*  *isStoring()* |
| *4* | *Employee 2* | *Varance, 40, ID: E002, Salary: 1000$, bonus:300$* | *isPicking(),*  *isShipping(),*  *isStoring()* |
| *5* | *Employee 3* | *Jack, 27, ID: E003, Salary: 1000$, bonus:300$* | *isPicking(),*  *isShipping(),*  *isStoring()* |
| *6* | *Inventory 1* | *Knife, ID: I001, Datein: 26/10/2022, Kitchen Items, 1000 items, new quality, Japan, Price: 27.59$, located in warehouse 1* |  |
| *7* | *Inventory 2* | *Body Soap, ID: I002, Datein: 26/10/2022, Toilettories, 3000 items, new quality, Korea, Price: 28.49$, located in warehouse 2* |  |
| *8* | *Inventory 3* | *Vacumm Cleaner, ID: I003, Datein: 26/10/2022, Kitchen Items, 500 items, new quality, Japan, Price: 139.99$, located in warehouse 1* |  |
|  |  |  |  |
|  |  |  |  |

*Table1. List of Objects*

*1-Nurse*

* 1. *NurseA, States (Nguyen Thi A, 35, General Diagnosis Department, Full Time), Behaviors (isWorking, isPaid, ….)*

*2-Doctor*

*3-Patient*

1. *Classes*

* *Create classes (name, method, functions) for each group. List the classes accordingly.*
* *You can use the table containing objects and their class.*
* *Analyse the inheritance among classes, Abstract classes*
* *Draw a diagram to show the inheritance (Note: at the analysis stage, please do not provide the details of the classes such as data types, variable names, return datatype of methods and the param lists of the methods)*

1. **CLASS DESIGN**
2. *Classes*

* *Add a Class diagram: relationship among classes*

*Figure 2. Class Diagram of Project …..*

* *Add a table (provides more details)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *No* | *Class* | *Instance Variable* | *Methods* | *Description* |
| *1* | *Staff* | 1. *Private String fullname;*  * *It is public because the other classes or in the other packages cannot read this personal information*   *Private Int age*  *Public String emailaddr;* | 1. *Boolean getFullname(string staffID)*  * *This function is used to get the fullname of the staff and return the success (true) or failure (false).*  1. *Boolean setFullname(string sID)*  * *This function is used to …* | *This class is used for managing the staff information and behaviors.* |
| *2* |  |  |  |  |

*Table 2. Details of Classes*

* *For each class, design the detailed members. Students please choose the method/class/variable type for each class: public, private, default, final, static,… with explanation.*
* *Abstract classes*

*Create a table of abstract classes*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *No* | *Abstract Class* | *Abstract Methods* | *Concrete Methods* | *Decription* |
| *1* | *PersonInformation* | *setInfo()* | *Util()* | *This class is an abstract class that is used by subclasses A, B, C, X, Y, Z, …. (See Figure 2.)* |
| *2* |  |  |  |  |

1. *Some OOP techniques*

*2.1.Overloading method:*

* *methods 1, 2,3 in class 1*

*Attached the class code if it is possible.*

*(Add some code of some significant classes you think there is a need)*

*abstract class 1{//abstract class*

*…. //overloading methods*

*// call-by-value*

*}*

*….*

* *methods 4, x, y in class 2*
* *..*

*2.2. Overriding method:*

* *methods 11, 12, 13 in class 1*
* *..*

1. *Inheritance*

* Present which inheritance-related techniques you use in those classes
* Representative code
* …

1. **Package Design**

* Analyse the package hierarchy used in your project.
* …

1. **Interface Design**

* Design interface you use in your project.
* …

1. **Access Control**

* Analyse and discuss on the access control relating variables (data), method, classes and packages, interface.

Table X1: Data Access Control Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Data | Class | Modifier | Description |
| 1 | Name, Department | Student | Public | * The name of students can be accessed by the other classes inside or outside the package of Student. * Name can be public as it is not private information. It does not cause the serious consequence if the others know the name of student. * Classes A1, A2, A3,…can access these data. |
| 2 | Age, Mailing address | Student | Private |  |
| … |  |  |  |  |

Table X2: Method Access Control Table

|  |  |  |  |
| --- | --- | --- | --- |
| No |  |  |  |
| 1 |  |  |  |
| 2 |  |  |  |
| … |  |  |  |

….

Note: students can compare or clarify using the tables or figures, whenever you insert a table or figure, you have to have at least one paragraph to discuss on their content.

…

1. **Encapsulation vs Inheritance vs Polymorphism**

[In this section, students will discuss on the three OOP features: Encapsulation, Polymorphism, Inheritance by in each part, you give the sample from your project code and explain why do you think those sample relates to the above three features.]

1. **Encapsulation**

…

1. **Inheritance**

…

1. **Polymorphism**

…

1. **Experiment**
2. **Environment and Tools**
3. **Environment:** Describe the physical resources (numbers of PCs, CPU, RAM, …)
4. **Tools**: List tools, libraries you use for your project in here. If that tool is so new, please add a section Appendix to instruct how to install the tools.
5. …
6. **Project functions**

* List and describe a bit the functions of the project
  + Search the students by their Department, Total Grade, ….
  + Input the Exam registration: student ID, Module, Exam date, ….
  + …
* Student can provide a table for this section
* …

1. **Database (4 tables)**

* Using flat file (.txt, .csv, …) or database (MySQL, MS SQL, …)
* Data Diagram (tables if you use relational database system management, the text files you use to store the data, …) - **5-7 classes, max. 10 classes**
* …

1. **GUI (4 figures)**

* List a table of the name and the orders of user interfaces in the projects

Figure 1: Student Input Dialog

……(paste the figure 1 in here)…

(explain the function of the buttons, and the button-related event, what happen if I press that button? When a user presses the button “Input”, an ID is automatically generated for that student and his/her information (i.e., Name, Department, ….) is stored to the table Student in the database, through the method inputStudent(…) of the class StudentINfoInteraction.

* Paste the figures of user interfaces/dialogs/webpages in here according to the order listed in the above table.
  + For each figure, describe its functions, and its GUI components on the user interface, how to use it. Should give examples.
  + …
* ...

1. **Conclusion**

* Assess your project by discussing on its pros and the cons.
* I want to hear what you really think about your project quality, and leave a score you think your project can gain along with the reason why you think so.
* What more function you think you can improve and add to the project in the future.
* …

**DUTY ROSTER**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Task** | **In Charge** | **Start** | **End** | **State** | **Note** |
| 1 | Design Class Human (Manager, Employee);  Class Inventory | Hoang The Vinh -17469 | 19-Oct-22 | 23-Oct-22 | Done |  |
| 2 | Design Class Vehicle, Warehou-se | Nguyen Van Khanh - 17096 | 19-oct-22 | 23-Oct-22 | Done |  |
| 3 | Report Section II | Le Thi B | 02-Jan-19 |  | In progress |  |
| … | … | … | … | … | … | … |
| n | … | … | … | … | … | … |

**REFERENCE**

1. Tutorial Page, Oracle https://...
2. …

*[Students, please put here whatever sources you referred or used in here]*

APPENDIX A: DATABASE

* Add database diagram
* Functions
* …

APPENDIX B: GRAPHICAL USER INTERFACE (GUI)

* Add the UI diagram
* Classes (description: its functions, how to use)
* Screenshots of dialogs

APPENDIX C: CLASS DESCRIPTION

* Class 1 : Grade…. (Source: Src/Grade.java)