PSD Final Assignment Documentation



|  |  |
| --- | --- |
| Student Name : | Anson Ling Guang Cheng |
| Student ID : | D22124534 |
| Lecture Name : | Dr Catherine Higgins |
| Programme : | TU914/1 |
| Module : | ‘Software Dev & Problem Solving and Design’ |
| Submission Date : | 28th April 2024 |
| Word Count : |  |

Declaration of Ownership

This is to certify that the work here presented is entirely my own and that all external sources have been fully acknowledged.

**Student Signature:** Anson Ling Guang Cheng

Table of Contents

[Introduction 4](#_Toc161487265)

[Diary Report 5](#_Toc161487266)

[Understanding of Project – Step 1 7](#_Toc161487267)

[Problem and Solving 7](#_Toc161487268)

[List all the data that need to store in the resulting program 9](#_Toc161487269)

[Convert the data item into variable name 10](#_Toc161487270)

[Analysis of the Problem – Step 2 12](#_Toc161487271)

[Analysis the tasks 12](#_Toc161487272)

[Design of Complex Tasks – Step 3 14](#_Toc161487273)

# Introduction

This is the final assignment require by the school which is Technological University Dublin that we need to create and design a stock control system which will contains three steps.

# Diary Report

1. **DAY 1 – 11/03/2024 – Monday**

* On this day, the developer has started creating the file for OOSD & PSD and also doing the PSD documentation with the actual template.

1. **DAY 2 – 12/03/2024 – Tuesday**

* Start doing the analysing of the program and be able to finish the design with all the description and function I need in the program.
* Start coding the data type for inventory as super class, mobile and laptop as subclasses, and the driver class as well.
* Start coding the menu for the driver class which basically finish half of the menu like main menu, and customer menu, also doing the design the validation for the option. Furthermore, it was catching the error if the users accidently input the character instead of number, and I using try-catch statement here for catching the exception.

1. **DAY 3 – 13/03/2024 – Wednesday**

* Start coding the file handling system when customer menu required display to be able read from the file (csv file). This stage of design is also using try-catch statement to force java reading from the text file even though java think that’s wasn’t safe.
* Got a problem on menu option 2 which is when user click on the second option, it requires user to enter the password, but having a bug there and not even finish fixing after 6 to 7 hours. The problem basically is when user enter the password, if system check it doesn’t have an account, it will ask user to create a new password. After that, when I re-enter the new password it asks me to create a new password again which doesn’t make sense. But with this problem, I’m will do the simple version first and ask the lecture on Friday class.
* Start doing another coding for staff menu option 1 & 2 which require java enter the products details and add to the list, also update to the text file. With this I watch a bunch of videos through YouTube that teach us how to do that. I use try catch statement again for import the products information to the file. After that, found two problems here, first when adding the program crash because couldn’t found the location, but I was able to fix it because there is a little mistake on located the file. Another problem is although the data is been import but it was a one cell which doesn’t look good. So, I did some research which I know we can use Array and comma method to import that data in order.

1. **DAY 4 – 14/03/2024 – Thursday**

* Found a way to reducing the line of code, so instead of me create text file in all the method, I am going to create a pure module call “updateFileText” that can be call whenever the other method needs to use it for update text file.
* Start doing the coding for update the stock and the delivery. I am able to finish the updating one, but the delivery one still need times to think about it.
* Changing the code for display mobile and chair for sales, instead for using reader and print out unorder list, I’m just going to print out by using toString method.
* Found some bug like calculation problem and update the prices to the text file, but I am able to fix it because just a small little error there.
* Complete most of the function like customer menu and staff menu, but with some of the problem I have is not been fix yet like staff password and also the UI enhance.

1. **DAY 5 – 15/03/2024 – Friday**

* Found a bug on display mobile detail and got stuck on this problem for an hour, I was able to fix it by lecture helping.
* Found another bug on when I try to crash my program when I input the incorrect password, but even if I create a password already but it still asks me to create a password. – haven’t fix yet.
* I able to fix the previous problem I have on staff password, which I create an empty object on the top which there is nothing the box, so basically, I just delete it and put it at the bottom.
* Also, I enhance a little UI which make it a little bit better than the previous one.

1. **DAY 6 – 15/03/2024 – Saturday**

* Found a bug on linking the menu, I was able to fix it by checking the problem one by one. But end up by creating a pure module that doing validation for catching the input error.
* Continue enhance the UI.

# Understanding of Project – Step 1

This section can be categories into three part which is descript the problem that been solve, lists out all the data that need to be solve, and convert the data.

1. Problem and solving

* This part will be about all the problem that the developer currently facing and how was the problem been solving. This also include the detail of the business and the type of products that will be processed and an indication of developer personal action.

1. Lists out all the data

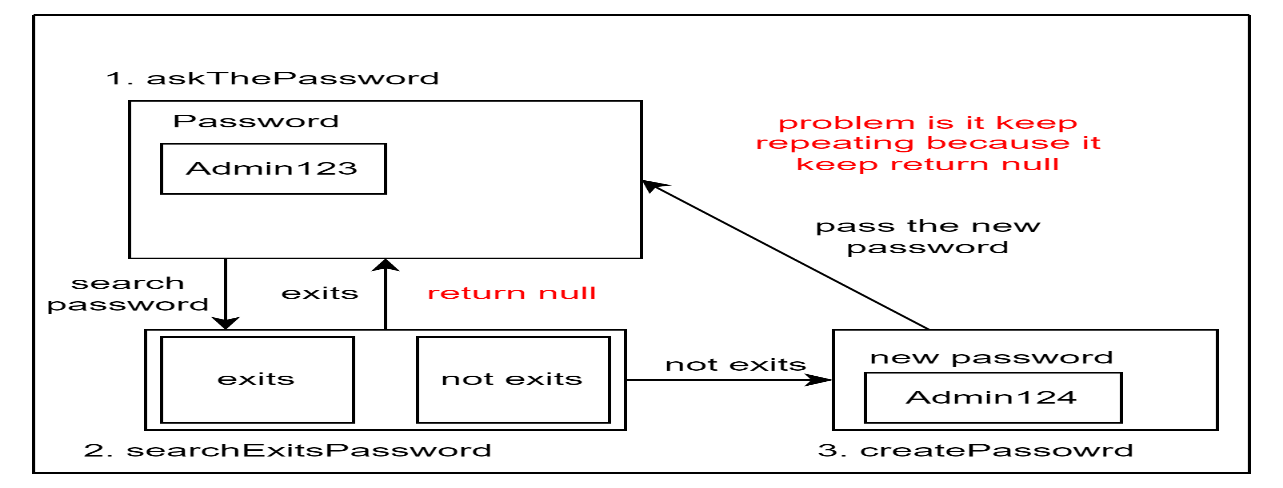
* This part will basically cover all the important value that will be use in the program.

1. Convert the data

* This part will convert the data in the program into variable name with the sample value.

## Problem and Solving

1. Problem

* So basically, when we click on the staff menu, it will ask staff to enter the password. If password exits, it will go and show staff menu, but if it’s does not exist, ask staff to create. The problem here is when I already create a password and re-enter the password, it keeps asking me to create a password which I already did. As the example below;
* As the picture show, because the search method is always return null which mean not exits than the system will always go to create password method that ask user to input the new password.

Temporary solution

* So basically, I just setup a password as private and final which on one could access and changes the password as a global variable, also make a do while loop that keep tracking the correct password, if staff enter wrong three times, display massage that tell staff you have use all three attempts.

Update Solution

* So, the problem was I create a empty object which store nothing in the box that will always return null, the solution here just simply delete and copy that line of code at the bottom which make sure the value will be passing in.

1. Problem

* The next problem is having trouble to update the text file include update the file after the action like adding stock, prices, and purchase. After that, I watch a bunch of videos that teach how to implement the writer and reader, so that I could import the value to the text file.

Solution

* The solution here is using try catch statement that will try to catch the location that can create a file that store the value. But instead of using try catch statement that import the file in most of the method which going to occupy a lot of unnecessary space, I just create a pure module that could be call whenever it needs.
* I also did a comma between the value which it will arrange the value organize by using array as well. And I learn most of the thing through the lecture note, YouTube and google.

1. Problem

* The next problem is having trouble displaying the mobile detail which is when I input the laptop detail it will crash because the java couldn’t catch Mobile for some reason. The reason why I couldn’t catch mobile because I put a catch method that force java to do this variable.

Example;

Mobile = (March) each;

* Which it forcing java convert the data with laptop, so it crashes.

Solution

* So basically, I just need to delete that line of code and create an if else statement with using “instanceof” method inside the for each loop to let the java know which products need to be display.

## List all the data that need to store in the resulting program

|  |  |  |  |
| --- | --- | --- | --- |
| Inventory (Super Class) | | | |
| Number | Input | Output | Result |
| 1) | Number of Stock | 20 |  |
| Prices | 1200 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Convert the data item into variable name

|  |  |  |  |
| --- | --- | --- | --- |
| Main | | | |
| Input: | Type of Variable | Global/Local/Instance/Reference | Location |
| finalFile, finalFile2, finalFile3. | String | Global  Global, reference | Main class, global |
| count | int |
| list, sList | Inventory, Staff |
| list, sList | Inventory, Staff | Local | Main constructor |
| Splite | String | Local | updateTextFile |
| each | Inventory | Local, reference |
| laptop | Laptop | Local, reference |
| mobile | Mobile | Local, reference |
| laptopList, list | String | Local |
| menuOption | int | askMenuOption |
| option | int | Local | checkOption |
| isValidOption | boolean |
| customerOption | int | Local | askCustomerMenuOption |
| mobile | Mobile | Local, reference | buyMobile, buyLaptop |
| brand, model,ans | String | Local |
| quantity | int | Local |
| password | String | Local | askThePassword |
| staff | Staff | Local, reference |
| staff | Staff | Local, reference | searchPassword |
| staName,  staPass,  staID | String | Local | createPassword |
| staff | Staff | Local, reference |
| staffOption | int | Local | askStaffMenuOption |
| Brand, model, memoryOption, ans | String | Local | staffMenuOption1, staffMenuOption2 |
| numOfAsk | int |
| mobile | Mobile | Local, reference |
| mobile, laptop | Mobile, Laptop | Local, reference | UpdateStockAndDelivery |
| typeOfProd,  brand,  model. | String | Local |
| each | Inventory | Local, reference | checkStock |
| stock | int | Local | setMobileStockAndPrices,  setLaptopStockAndPrices |
| prices | double |
| mobile | Mobile | Local, reference | searchMobileProducts,  searchLaptopProducts |
| laptop | Laptop |
| input | String | Local | exit |

|  |  |  |  |
| --- | --- | --- | --- |
| Inventory | | | |
| Input: | Type of Variable | Global/Local/Instance/Reference | Location |
| numOfStock | int | Global, local, instance | Two constructors, getter, setter, toString. |
| prices | double |

|  |  |  |  |
| --- | --- | --- | --- |
| Mobile & Laptop | | | |
| Input: | Type of Variable | Global/Local/Instance/Reference | Location |
| tyOfProduct | String | Global, local, instance | Two constructors,  getter,  setter,  toString,  toStockString. |
| brand |
| modelMob & Lap |
| memoryOption |

|  |  |  |  |
| --- | --- | --- | --- |
| Staff | | | |
| Input: | Type of Variable | Global/Local/Instance/Reference | Location |
| staffName | String | Global, local, instance | Two constructors,  getter,  setter,  toString. |
| staffID |
| password |

# Analysis of the Problem – Step 2

This section will be coloring all the coded steps/tasks and analyzing the thing that has to be done. However, this section will not include the actual code and any sort of text that is relevant to the code itself, yet it will only analyze how the tasks have been distributed and will be implemented.

**Colour example as below;**

|  |  |
| --- | --- |
| Blue - | Designing the data type |
| Green - | The tasks that don’t need design |
| Yellow - | The tasks need to be designed |

## Analysis the tasks

1. Creating the data type for Inventory
   1. two constructors.
   2. Getter and setter.
   3. toString.
2. Creating data type for Mobile
   1. two constructors.
   2. Getter and setter.
   3. toString.
3. Creating data type for Laptop
   1. Two constructors.
   2. Getter and setter.
   3. toString.
4. Create data type for staff

4.1Two constructs.

4.2 Getter and setter.

4.2 toString.

1. Ask the customer to enter the option 1 – 3

5.1 If the customer enters the wrong option or type of option, display the wrong option and try again.

1. Option 1 – Customer Menu
   1. Display the menu 1 – 5.
   2. Ask the customer to pick the option above.
2. Customer menu option 1 – Display all the mobile for sales
3. Customer menu option 2 – Display all the laptops for sales
4. Customer menu option 3
   1. Ask the customer what the branch of the phones.
   2. Ask the customer how many you wished to purchase
   3. If the branch of the laptop or currently no stock, display not found or no stock
      1. If there is enough stock, calculate the prices and display the purchase as successful and receipts.
      2. If there is not enough stock, display the purchase couldn’t be processed due to not enough stock and no action taken.
5. Customer menu option 4
   1. Ask the customer what the branch of the laptops.
   2. Ask the customer how many you wished to purchase
   3. If the branch of the laptop or currently no stock, display not found or no stock
      1. If there is enough stock, calculate the prices and display the purchase successful and receipts.
      2. If there is not enough stock, display the purchase couldn’t be processed due to not enough stock and no action taken.
6. Customer menu option 5 – Go back to the main menu
7. Option 2 – Staff Menu
   1. Ask the staff to input the password.
   2. If the staff enters the wrong password three times, the staff won’t be able to enter again.
   3. If the staff hasn’t registered yet, ask the staff to create the password
   4. Once login, display the menu 1 – 4.
   5. Ask the customer to pick the option above.
8. Staff Menu Option 1 – Ask the staff to add the new type of mobile phones to the stock list

12.1 Ask the staff to input the brand of the mobile phone.

12.2 Ask the staff to input the model of the mobile phone.

12.3 Ask the staff to input the memory option of the mobile phone.

1. Staff Menu Option 2 – Ask the staff to add the new type of laptops to the stock list

13.1 Ask the staff to input the brand of the laptop.

13.2 Ask the staff to input the model of the laptop.

13.3 Ask the staff to input the memory option of the mobile phone.

1. Staff Menu Option 3 – Update the stock level for new products with new delivery
   1. Ask the staff did you wish to update the stock level.
   2. If yes, Ask the staff to input the brand and model.

14.3.1 If the brand and model is found on the list, ask the staff how man stock you wish to delivery and what the price you wish to set .

14.3.2 Update the stock and prices.

14.3.3 Display a message that the stock has been delivered.

* 1. If no, the delivery will be canceled, and display message to inform staff.

1. Display the current stock status, to check which products to add.
2. Staff menu Option 4 – Go back to the main menu.
3. Option 3 – Exit the program.

# Design of Complex Tasks – Step 3

This section will need the developer think about how to implement the task for any tasks that need to be design like yellow and blue colour. Check out the colour example on step 2 (Analysis of the program).

|  |  |
| --- | --- |
| 1, 2, 3 & 4 | |
|  | * Creating a data type * Two constructors. * Getter & Setter. * toString. |
| 5 | |
|  | * Ask the customer/staff to input the option * If is not a correct option, display error message and try again * If is correct option, connect to other page base on the option * If customer input 1, it connects to the customer menu * If customer input 2, it connects to the staff menu |
| 6 | |
|  | * Display the menu. * Ask the customer/staff to input the option * If is not a correct option, display error message and try again * If is correct option, connect to other page base on the option * If customer input 1, it connects to the display mobile * If customer input 2, it connects to the display laptop * If customer input 3, it connects to the purchase mobile * If customer input 4, it connects to the purchase laptop |
| 7 & 8 | |
|  | * Display the mobile and laptop detail that is for sales |
| 9 & 10 | |
|  | * Ask the customer to enter the mobile or laptop detail like brand and model of the product. * If product is found, ask the customer how much you wished to purchase * Ask customer did you wish to process the payment * If yes, display the detail, prices and receipts. * If no, display no found the stock on the list. |
| 11 | |
|  | * Connect back to the main menu |
| 12 | |
|  | * Option 2 staff menu * Ask the staff |
| 13 & 14 | |
|  | * Staff menu option 1 & 2 * Ask the staff to input the brand. * Example: Samsung, Apple, Google * Ask the staff to input the model. * Example: Galaxy S21, iPhone 12, Google pixel. * Ask the staff to input the memory option. * Example: 128GB, 256GB, 512GB,1TB. * Ask the staff did you wish to process? * If yes update to the text file and object and display message that tell staff item is been added. * If no, no action taken, display message couldn’t process. * If not both of them, display incorrect answer. |
| 15 | |
|  | * Staff menu option 3 * Ask the staff to enter the type of product * If it mobile and laptop to the following; * Ask user to input the brand and the model. * If match, ask staff to enter the stock wish to add and the prices of the products. * If not match, display message that couldn’t found the product. * If it not both of them, display currently not support the type of product. |
| 16 | |
|  | * Staff menu option 4 * Display the product detail with stock and prices, to check which product of stock need to delivery. |
| 17 | |
|  | * Staff menu option 5 * Go back to the main menu. |
| 18 | |
|  | * Main option 3 * Exit the program. |