

INTRODUCTION

This assignment covers the following Course Learning Outcomes (CLO):

| | Learning Outcomes | Assessment |
|-------------|---|------------------------------------|
| CLO2 | Apply suitable concepts and programming structures in developing solution to problems (C3, PLO2) | Group Assignment – Problem Solving |
| CLO3 | Work in team to develop a software solution by applying basic object-oriented techniques in a data source based application(A3, PLO4) | Group Assignment – Solution |

ASSIGNMENT BRIEF

APU Printing Services Management System

APU Printing Services Sdn Bhd has introduced a document printing & binding services for its APU students. The details are as provided in the table below:

| No | Service Type | Fees (RM) |
|-----------|------------------------------|------------------|
| 1 | Printing A4– Black and White | 0.8 / page |
| 2 | Printing A4 – Color | 2.50 / page |
| 3 | Binding – Comb Binding | 5 / book |
| 4 | Binding – Thick Cover | 15 / book |
| 5 | Poster printing (A0 – A3) | 3 / page |
| 6 | Banner | 10 / banner |

Note: There is a surcharge of 30% for any urgent requests.

General system requirements:

1. There are four user roles in this system – Customer, Manager, Worker, and Admin.
2. Each user has their own login name and password.
3. System should recognise their user role upon successful login.
4. Each user can update their own profile.

Specific System Requirements:

1. Customer (student) :-
 - a. Can log into their system with their username and password
 - b. Can view the services provided with the fees
 - c. Can submit new request(s) for any of the services above
 - d. Can check the amount they need to pay for the services that have requested
 - e. Can check the status of their previous requests
2. Manager :-
 - a. Can log into their system with their username and password

- b. Can check the status of all requests (*New, Work In Progress and Completed*)
 - c. Can assign new request(s) to a worker
 - d. Accept payment from customer and generate receipt.
3. Worker :-
- a. Can log into their system with their username and password
 - b. Can check the new requests assigned to them
 - c. Can update the status of the request once they have completed it
4. Admin
- a. Can log into their system with their username and password
 - b. Register new users
 - c. View and generate request report (monthly)
 - d. View and generate total income (monthly)

IMPORTANT INSTRUCTIONS:

This is a group assignment. Upon submission of your assignment, you are required to present your assignment at a date and time specified by your module lecturer. Each team member is required to contribute towards some features in the system and documentation, present and explain his or her work accordingly. Each team member should also be able to answer questions during presentation with regards to the overall systems project and or specific question(s) related to the codes used in the development.

GENERAL REQUIREMENTS

The program you submit should be able to compile without errors using Visual Studio. You should comply with the following coding style guidelines for the system:

- Use C# features
- Use Graphical user interface
- Use object-oriented concepts in the solution
- Document your codes by using comments where necessary
- Use indentation
- Use meaningful names for identifiers

You should store all the data in a database management system such as Microsoft Access, Microsoft SQL Server, SQL Server Standalone database, or any other supported database so that the information can be retrieved later.

This is a standalone application and used by the specified users in the above description. **Validation of input data** should be included for various input values to prevent invalid values to be entered and to prevent errors. The implementation code must include **at a minimum** the use of object-oriented programming concepts such as classes, methods, and objects.

COMPLETED APPLICATION DELIVERABLES

The completed application and documentation must be uploaded to Moodle on or before **the due date (refer to Moodle for details)**.

The project source folder must be zipped before uploading to Moodle.

Ensure that your zip file contains all the relevant source code.

The font size used in the report must be **12pt**, and the font type used should be **Times New Roman**.

Full source code is **not** to be included in the report. The report must be typed and clearly printed.

Header and Footer

Ensure that headers and footers are included in the documentation

Header: Module code and module name on the left followed by page number (right).

Footer: The intake code (left) and institution name (center) of the footer.

DOCUMENTATION GROUP ASSIGNMENT REPORT

As part of the assessment, you must submit the project report in PDF format. Upload to Moodle with the following format

1. Cover Page:

All reports must be prepared with a *front cover*. The front cover should be presented with the following details:

- Module
- Title:
- Intake
- Students' ID Number and Name
- Date Assigned (the date the report was handed out).
- Date Completed (the date the report is due to be handed in).

2. Contents

- a. **Storyboard** that shows the draft design of the screens to be implemented in the system. This may be documented through free-hand sketches or may be documented using a wire-framing tool such as Pencil (<https://pencil.evolus.vn/>), an open-source software. (*refer to Appendix A for sample*)
- b. A **use-case diagram** showing the actors and uses-cases in the application
- c. A **class diagram** showing the classes and methods used in the application
- d. Explanation of the codes implemented in the system where the following Object-oriented programming concepts has been used:
 - Classes
 - Methods
 - Objects

In each of the above, provide code snippets to support the explanation where the concepts were applied.

- e. Test Plan and test cases documenting the functions tested in the system. At least 15 test cases should be documented. (*refer to Appendix B for sample*)

3. Conclusion

A critical assessment of the system developed which includes the strengths, the weaknesses, and recommendations for future enhancements.

4. References

- You may source algorithms and information from the Internet or books. Proper referencing of the resources should be evident in the document.
- Any references must adhere to APA referencing format which can be accessed from <http://library.apiit.edu.my/apa-referencing/>

5. Workload Matrix (refer to Appendix C for template)

The workload matrix should indicate the section or task completed by each team member. Individual Marks will be distributed accordingly by each team member's contribution towards the assignment.

ACADEMIC INTEGRITY

- You are expected to maintain the utmost level of academic integrity during the duration of the course.
- Plagiarism is a serious offence and will be dealt with in accordance to APU Academic Dishonesty regulations on plagiarism which can be found on Webspace https://lms2.apiit.edu.my/pluginfile.php/3654/mod_resource/content/3/Academic%20Dishonesty%20Regulations%202018-03-March.pdf

Appendix A – Sample Storyboard Documentation

Product Checkout Form

Product Checkout

Quantity

Product Name

Order List

Unit Price

Amount Due: RM 790.00

Add Product

Cancel Last Entry

Payment

| PCode | Prod Name | Quantity | Unit Price | Subtotal |
|-------|---------------|----------|------------|----------|
| B0123 | Stonebowl | 50 | 15.00 | 750.00 |
| K1234 | Ceramic Knife | 5 | 8.00 | 40.00 |

| Control | Control Name | Description |
|----------|---------------|--|
| Label 1 | Label 1 | To label the related controls to the right |
| Label 2 | Label 2 | |
| Label 3 | Label 3 | |
| Label 4 | lblUnitPrice | To display the unit price of a selected product |
| ComboBox | cboProduct | To allow selection of products from a list |
| TextBox | txtQuantity | To allow entering of product quantity needed |
| Button 1 | btnAddProduct | To enable adding of a selected product to the order list |
| Button 2 | btnCancel | To enable removing or editing of the last product entry in the order list |
| Button 3 | btnPayment | To calculate the total order amount and print the receipt for the products ordered |

Appendix B – Test Plan and Test Cases Template *(Two sample entries are shown)*

| Test Case | Function Name | Test Objective | Expected Result | Actual Result | Remarks |
|-----------|--|--|--|---|---|
| 1 | <i>Name of the function to be tested</i> | <i>The test objective will indicate what is to be tested</i> | <i>The expected result of the function as how it is supposed to work</i> | <i>The result from the function execution during testing</i> | <i>Optional, add remarks if needed correction to the function</i> |
| 2 | <i>Add Member – validate email address</i> | <i>To test whether validation of email is correct</i> | <i>Display error message if the email entered does not have the “@” and “.” in the text string</i> | <ul style="list-style-type: none"> ▪ Only display error message if “@” is omitted in the email. ▪ No error displayed for the “.” omission | <i>Function codes need to be checked again</i> |
| 3 | <i>Update Member status</i> | <i>Member details can be updated</i> | <i>Display notification when update is successful</i> | <ul style="list-style-type: none"> ▪ Notification is displayed correctly | <i>None</i> |

Appendix C – Workload Matrix Template

| No. | Assigned Task & Brief Description | Assigned Member Name | Completion Status / Comment |
|-----|-----------------------------------|----------------------|-----------------------------|
| 1. | | | |
| 2. | | | |
| 3. | | | |