

## **Learning Outcomes**

On conclusion students should be able to:

CLO2: Build stand-alone programs using a visual development environment (C3, PLO2).

CLO3: Present an application which apply graphical user interface programming and database programming capabilities to solve a problem (A2, PLO5).

## **Programme Learning Outcomes (PLO)**

PLO2: Demonstrate logical and analytical thinking skills to develop innovative software solutions for various applications.

PLO5: Be a responsible IT practitioner within a social environment, with a capability to work in a team.

## Assignment Title

### CAFETERIA ORDERING SYSTEM

#### Section A: Program Specification

This project requires you to develop a Cafeteria Ordering System to handle all the primary information required to maintain customer orders. Once the system is launched, staff will have to login to enter the main screen. Upon login, the system should display the user's name on the system's user interface. The system must have the following functionalities:

a. **Create and modify customer account**

This module handles account creation for a new customer. In order to register with the cafeteria, the customer is required to fill a registration form to provide his personal details. The staff then enters the customer details into the system, and thereafter provides the customer with the customer number. The customer then can perform transactions using this customer number.

b. **Place order**

The system shall display a menu of food items that are available in the cafeteria. Customers will indicate the quantity of food item that they wish to order. The system shall display only the food items that has been ordered, the quantity and price of individual food items and the subtotal, tax and total payable amount. The system then will prompt the staff to confirm the meal order. If the staff does not confirm the meal order, he or she may either edit or cancel the order.

c. **Payment system**

Customers may select to pay using coins (RM0.10, RM0.20, and RM0.50) and notes (RM1, RM5, RM10, RM20, and RM50). At the end of the payment process, the change amount will be displayed if a customer pay in excess of the total amount.

**You are required to design and develop the database to store the data for the system using any relational database management tool e.g. SQLServer, SQLDatabase (standalone database in Visual Studio), MS-Access, MySQL, and so on. Ensure that the tables are normalized to prevent redundancy. At minimum the tables should include user, customer, order, and payment.**

**Section B: Deliverables**

The program submitted should compile and be executed without errors. Besides, validation should be done for each entry from the users in order to avoid logical errors.

**SUBMISSION REQUIREMENTS**

You are required to submit the following for your documentation:

**a. APU Assignment Cover Page****b. Table of Contents**

The table of contents should have the section headers and the corresponding page number.

**c. Introduction**

This should include a brief description of the system including the objectives for the new system and the functions to be included in the system.

**d. Storyboard**

This section includes a sketch design of Windows Forms used in the system, indicating the layout and the components to be used for the interface (forms and report structures). This may be drawn using mockup tools or wireframe software which can be found online. An example of wireframe software is *Pencil*, which is an open sourced software designing user interface downloadable from:

*<http://pencil.evolus.vn/Downloads.html>*

Brief description for each control used on a Windows Form is required. (*Refer to Appendix A for sample*)

**e. User Manual**

Produce a detailed user manual of at least 10-15 pages on using the system that you have developed. The manual must include step by step instructions to perform a specific task in the system which includes screen shots to assist users in using the system.

**f. Testing**

The completed system needs to be tested for the functions developed and all the tests will be required to be documented. At least 10 test cases should be documented. (*Refer to Appendix B for sample*)

**g. Critical Assessment**

Identify the strengths and the weakness that you or your group has found in the new system and the solutions. If there any weaknesses, provide recommendation(s) on how the system might be improved in the future.

**h. References**

Document any reference materials you have sought during the development of the software solution which includes websites and forums such as StackOverFlow. All references must be documented using the Harvard Referencing style. Refer to APU's Library website for examples of Harvard Referencing style.

**URL:** <https://library.apiit.edu.my/harvard-reference-style.html>

**PRESENTATION**

The presentation is divided into 2 parts:

1. Storyboard presentation (Milestone 1: Should be completed by **Week 8**)
  - You are required to prepare the storyboard showing the form layout and explain the system flow between the forms.
2. Project demonstration (Milestone 2: Should be completed by **June 1<sup>st</sup> 2020**)
  - Each student is to prepare a 10-15 minutes presentation to demonstrate the system. You do not need to prepare any presentation slides for the presentation. The presentation will be graded based on the ability to demonstrate the working of the system and your ability to answer questions asked related to the program developed which may include code modification or explanation of code used in the system during the presentation.

If anyone is absent from the scheduled presentation, an Extenuating Circumstances (EC) will need to be submitted for the presentation by the student. The possibility of being awarded zero mark for the system implementation (Group Component – Coding and Implementation, Individual Components – Program Solution and Presentation) if the EC is rejected.

*Absence from presentation will require you to submit an EC.*

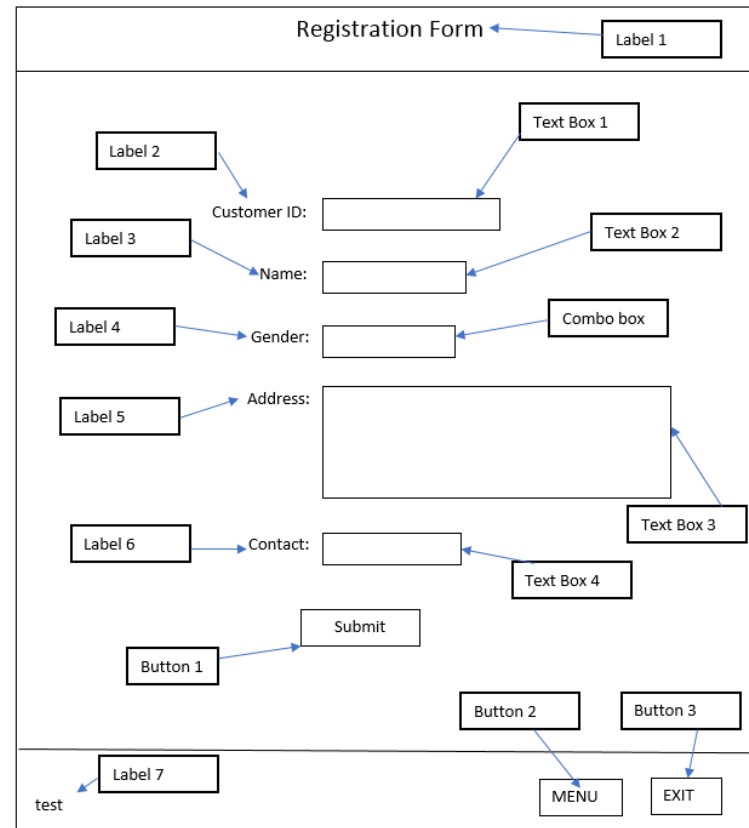
**Section C: Component Weighting**

Implementation	: 55%
Documentation	: 30%
Final presentation	: 5%
Presentation (User Interface Storyboard)	: 10%

*Plagiarism is a serious offence and will be dealt with according to APU regulations on plagiarism.*

## Appendix A – Sample Storyboard Documentation

### Customer Registration Form



Control	Control Name	Description
Label 1	lblTitle	Form title
Label 2	lblCustomerID	Label for customer ID
Label 3	lblCustomerName	Label for customer's name

Label 4	lblCustomerGender	Label for customer's gender
Label 5	lblCustomerAddress	Label for customer's address
Label 6	lblCustomerContact	Label for customer's contact
Label 7	lblUsername	Label for username
ComboBox1	cboGender	ComboBox for gender selection
Text Box 1	txtCustomerID	Textbox input field for Customer ID
Text Box 2	txtName	Textbox input field for Name
Text Box 3	txtAddress	Textbox input field for address
Text Box 4	txtContact	Textbox input field for contact number
Button 1	btnSubmit	Button to allow customers to submit their registration form
Button 2	btnMenu	Button to proceed to menu
Button 3	btnExit	Button to exit program

#### 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"> <li>Only display error message if "@" is omitted in the email.</li> <li>No error displayed for the "." omission</li> </ul>	<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"> <li>Notification is displayed correctly</li> </ul>	<i>None</i>