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

	Learning Outcomes	**Assessment**
CLO2	Apply suitable concepts and programing 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 datasource based application(A3, PLO4)	Group Assignment – Solution

ASSIGNMENT BRIEF

Laptop Repair Services Management System

A shop in Kuala Lumpur is providing laptop repair services. The types of services and their related details are given in the table below:

No	Service type	Service Fee	
		Normal	Urgent
1	Remove virus, malware or spyware	RM 50	RM 80
2	Troubleshot and fix computer running slow	RM 60	RM 90
3	Laptop screen replacement	RM 380	RM 430
4	Laptop keyboard replacement	RM 160	RM 200
5	Laptop battery replacement	RM 180	RM 210
6	Operating System Format and Installation	RM 100	RM 150
7	Data backup and recovery	RM 80	RM 130
8	Internet connectivity issues	RM 70	RM 100

Once a customer visits the shop a laptop service, he/she will describe the problem found in the laptop. For simplicity, assume there is only one problem in each visit.

There are total of 4 registered users in the system. You are required to develop the above system with following functionalities:

1. Receptionist

- Login - User should key in username and password only. System should identify their user role.
- Register new customer and choose requested service.
- Accept payment from customer and generate receipt.
- Update own profile.

2. Technician

- Login - User should key in username and password only. System should identify their user role.
- View service requested by customers.
- Add description and laptop collection date upon completion of service.
- Update own profile.

3. Customer

- Login - User should key in username and password only. System should identify their user role.
- Change requested service.
- View service description, laptop collection date and total amount to be paid.
- Update own profile.

Admin

- Login - User should key in username and password only. System should identify their user role.
- Register new technician.
- Register new receptionist.
- View service report (monthly).
- View total income (monthly).

<u>IMPORTANT INSTRUCTIONS:</u>

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 GUI for the 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.

MILESTONE 1 DELIVERABLES

Each group required to submit Storyboard and Prototype Application (UI design) on week 7.

COMPLETED APPLICATION DELIVERABLES

The completed application and documentation must be uploaded to Moodle on or before **10.00pm on the due date.**

- 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 by uploading to Moodle in 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
- Student's 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.

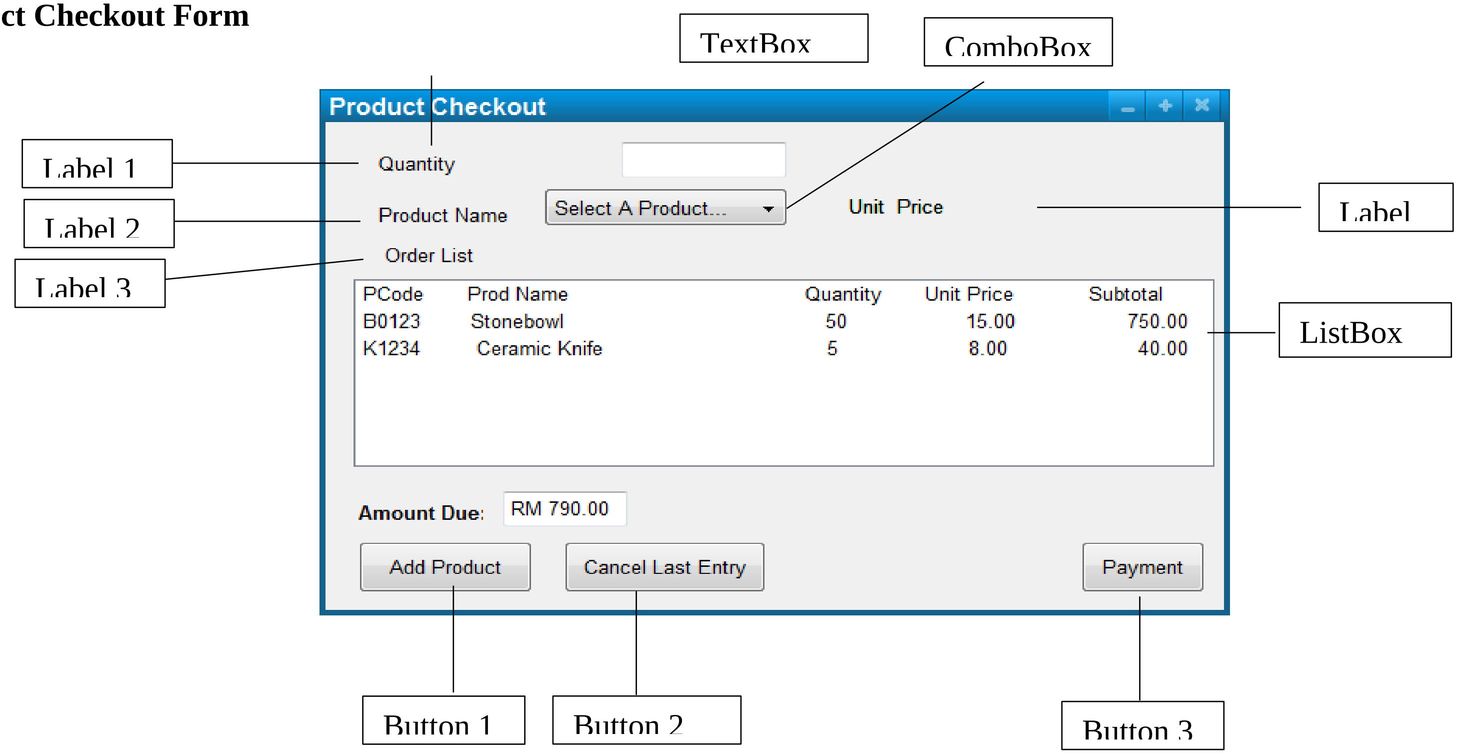
ACADEMIC INTEGRITY

- 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



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

Amount Due: RM 790.00

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	*Name of the function to be tested*	*The test objective will indicate what is to be tested*	*The expected result of the function as how it is supposed to work*	*The result from the function execution during testing*	*Optional, add remarks if needed correction to the function*
2	*Add Member – validate email address*	*To test whether validation of email is correct*	*Display error message if the email entered does not have the "@" and "." in the text string*	▪ *Only display error message if "@" is omitted in the email.* ▪ *No error displayed for the "." omission*	*Function codes need to be checked again*
3	*Update Member status*	*Member details can be updated*	*Display notification when update is successful*	▪ *Notification is displayed correctly*	*None*

Appendix C – Workload Matrix Template

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