**Learning Outcomes**

|  |  |  |
| --- | --- | --- |
| CLO | Learning Outcome | Assessment |
| 1 | Outline algorithms using computing logic to solve a problem (C1, PLO1) | Test |
| 2 | Transform a developed algorithm into a computer application (C2, PLO2) | Assignment |

**Assignment Specification**

*Addie’s Equipment* provides rental of garden equipment to customers. A customer’s information such as full name, address, phone number and / or mobile phone number needs to be entered into the application. Rental service is provided based on per day use and the number of days, up to a maximum of 14 days. So, if a customer rents only for one day only, the daily rate applies. However, if the customer renting for more than one day, the days rate will be used in the calculation. In addition, a 15% discount will be applied to the total rental if three or more equipments are rented.

|  |  |  |
| --- | --- | --- |
| **Equipment Type** | **Daily Rate** | **Days Rate** |
| Lawn Mower – Rotary | RM50.00 | RM38.00 |
| Lawn Mower – Ride On | RM80.00 | RM60.00 |
| Lawn Mower – Handheld | RM35.00 | RM26.00 |
| Mobile Garden Sprinkler (small) | RM100.00 | RM75.00 |
| Mobile Garden Sprinkler (large) | RM150.00 | RM113.00 |
| Garden Spray – 5L | RM30.00 | RM25.00 |
| Garden Spray – 10L | RM40.00 | RM32.00 |
| Outdoor High Pressure Pump | RM50.00 | RM38.00 |

Create an application for Addie’s Equipment to produce an estimated total rental of garden equipment for customers.

* The program should display the items selected for rental and calculate the estimated amount. This could be easily done using a ListBox control.
* The program should check that fields is not empty, otherwise, an appropriate error message should be displayed and the customer’s total payment due should not be calculated.
* The system should also check for non-numeric values and display an appropriate error message to the user.

***Note****: The system that you develop must meet the requirements described above. You are allowed to make your own judgments and assumptions in developing your system beyond meeting the minimum requirements. In addition, marks will be awarded for additional features to enhance the application.*

**Requirements**

**Outline of Task:**

You have been given the task to analyze, design and develop a prototype for the given Case Study individually.

###### Suggested Presentation Format

##### This assignment should be treated as though you are proposing and developing a system for an organization. The following is a suggested minimum documentation format. It is recommended that your document contain at least the following:

##### **Cover Page**

Your cover page should contain the following Items

* APIIT Logo
* Subject Title
* System Name
* Intake Title
* Intake Code
* Student Name
* Student ID

# **Table of Contents**

The table of contents should have the topic title and reference page number attached to each section.

# **Introduction**

This should include your task, the scope of the proposed system, and the objectives for the proposed system.

**Main Body**

1. **Flowcharts or Pseudocode**

This should include your flowcharts with necessary explanations/summary of the flowchart.

1. **Storyboard / Mockup Screen Designs**

This will be the draft designs for your program

1. **Conclusion**

The conclusion should identify the weaknesses in the program and how the program might be improved in the future.

###### **Programming Environment**

The assignment requires you to use **any version of Visual Basic Studio** that enables the development of graphical user interfaces (GUIs) for developing a prototype system.

# **The Documentation**

The documentation has to be word processed, printed on single side A4 size paper with 1.5 line spacing (optional) and comb bound (sample of the binding will be shown accordingly).

You have to **present your program** during a scheduled presentation session to the module tutor. The presentation will be judged accordingly and will be held normally one week after the submission of the assignment.

**What You Need to Hand In?**

1. You are required to hand in the individual assignment on the due date mentioned on the cover sheet of the assignment.
2. You should present an executable system and the presentation will be judged accordingly to the requirements mentioned in the assignment specification.
3. You have to submit your assignment with the Coursework Submission and Feedback Form (CSFF).

### **Marking Criteria**

The assignment will contribute **70%** towards the in-course assessment, as mentioned on the Student Assessment Information Sheet (SAIS). The document **must also contain the marking grid as per the sample shown below,** which should be attached after cover page.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Fail | Marginal Fail | Pass | Credit | Distinction |
| 0 - 8 | 9 - 11 | 12 – 14 | 15 – 20 | 21 - 30 |
| Pseudocodes/  Flowcharts  (30%) | * Show little understanding on flowcharting in an inappropriate and logical manner. * Improper use of symbols as it contains errors. * Relationship is not labeled correctly. | * Incomplete flowcharting in an inappropriate and logically manner * Less than 40% of basic requirements of the system runs * Improper use of symbols as it contains many errors. * Relationship may not be labeled correctly. | * Average design in terms of logic and style * Design solution covers between 40% - 60% of the basic requirements of the system * Some errors in design such as symbols used / logic of solution | * Good design with a variety of styles and unique logic * Design solution covers between 60% - 70% of the basic requirements of the system. * Minor errors in design in terms of symbols used / logic of solution | * Excellent design with a variety of styles and unique logic * Design solution covers more than 70% of the basic requirements of the system. * Hardly any errors in design in terms of symbols used / logic of solution |
|  |  |  |  |  |
| Coding  (30%) | 0 - 8 | 9 - 11 | 12 – 14 | 15 – 20 | 21 - 30 |
| * Not done * Less than 20% of basic requirements are met. The coding solution lacks proper structure and may not even related to the assignment * Very poor coding styles | * At least some coding done * Less than 40% of basic requirements of the system runs * Program / solution logically sound but due to errors in compilation, program could not run. * Coding solution done covering some basic Visual Basic. Net concepts * Common solution * Little or no mapping between design and program solution * Poor coding styles | * Between 40% - 50% of the basic requirements of the system runs * Coding solution done covering all basic Visual Basic. Net concepts * Common solution * Some mapping between design and program solution * Average coding styles | * Between 60% - 70% of the basic requirements of the system runs * Coding solution done contains basic and intermediate Visual Basic.Net concepts * Appropriate comments incorporated * Unique solution Good mapping between design and program solution * Good coding styles | * More than 70% of the basic requirements of the system runs * Coding solution done contains basic, intermediate and advanced Visual Basic.Net concepts * Excellent and relevant comments incorporated * Unique solution * Excellent mapping between design and program solution * Excellent coding styles |
|  |  |  |  |  |
| Input  Output  (10%) | 0 - 2 | 3 | 4 | 5 - 6 | 7 - 10 |
| * No input / output produced from the system | * Barely able to see any relevant input output * Output is not as expected * Inconsistency of design during execution | * Input values are validated but caused errors during execution * Output are not formatted * Meet expectation of design during execution with minimum errors | * Good input / output * Most of the input values are validated * Most of the output produced are formatted * Meet expectation of design during execution | * Excellent input /output * All input values are validated * All outputs are formatted * Exceed expectation of design during execution |
|  |  |  |  |  |
|  | Fail | Marginal Fail | Pass | Credit | Distinction |
| Documentation  (20%) | 0 – 5 | 6 -7 | 8 – 9 | 10 – 13 | 14 - 20 |
| * Not complete * No test cases documented * Sections of the assignment merely put together with many missing components * Very poor critical evaluation * Very poor Harvard referencing style | * Less than 40% of documentation complete * Less than 8 test cases documented * Poor layout / flow * Poor critical evaluation * Poor Harvard referencing style | * Between 40% - 50% of the documentation complete * Average layout / flow * Sample outputs available with hardly any explanation * Between 8 to 14 test cases documented * Appropriate critical evaluation * Appropriate Harvard referencing style | * Between 60% - 70% of the documentation complete * Good layout / flow * Sample outputs available with some relevant explanation * 15 test cases documented * Excellent critical evaluation with minimum errors * Excellent Harvard referencing style with minimum errors | * Above 70% of the documentation complete * Excellent layout / flow * Sample outputs available with good explanation * 15 or more test cases documented * Excellent critical evaluation * Excellent Harvard referencing style |
|  |  |  |  |  |
| Presentation  (20%) | 0 - 2 | 3 | 4 | 5 - 6 | 7 - 10 |
| * Did not present * Was not able to answer any questions posed correctly | * Barely able to explain the codes and / or work done * Did not know how to run the system * Not able to answer most questions posed correctly | * Able to explain some codes / work done * Was able to answer some questions posed correctly | * Good explanation of codes and / or work done * Was able to answer most questions posed correctly | * Excellent explanation of codes / work done * Able to show additional / new ideas / codes * Able to answer all questions posed correctly |
|  |  |  |  |  |

**Performance Criteria**

|  |  |
| --- | --- |
| 70% + | Work of distinguished quality. Documentation and implementation reflects a thorough understanding of designing flowchart and developing a system. An excellent assignment structure. Documentation is meaningful and complete. Good amount of references, varied and current. |
| 60% - 69% | Documentation of above average quality and implementation reflects a thorough understanding of designing flowchart and developing a system with very few errors. Documentation is meaningful and complete. A good assignment structure. References available and current. |
| 50% - 59% | Work on an average quality. Documentation and implementation reflects an average understanding of designing flowchart and developing a system. Documentation may be incomplete. A reasonable assignment structure. Adequate references. |
| 40% - 49% | Weak documentation. Documentation and implementation reflects a poor understanding of designing flowchart and developing a system. Documentation is incomplete. May have many grammar and spelling mistakes. Incomplete assignment structure. Little references and not appropriately cited. |
| 0% - 39% | Very poor documentation. Show very little understanding on drawing a flowchart and developing a system. Implementation fails to demonstrate a sufficient understanding. None or little references which are not cited appropriately. |

**Sample Storyboard Documentation**

The storyboard/mocked up design will include the 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 wireframe software which can be found online. An example of wireframe software is *Pencil,* whichisan open sourced software designing user interface downloadable from:

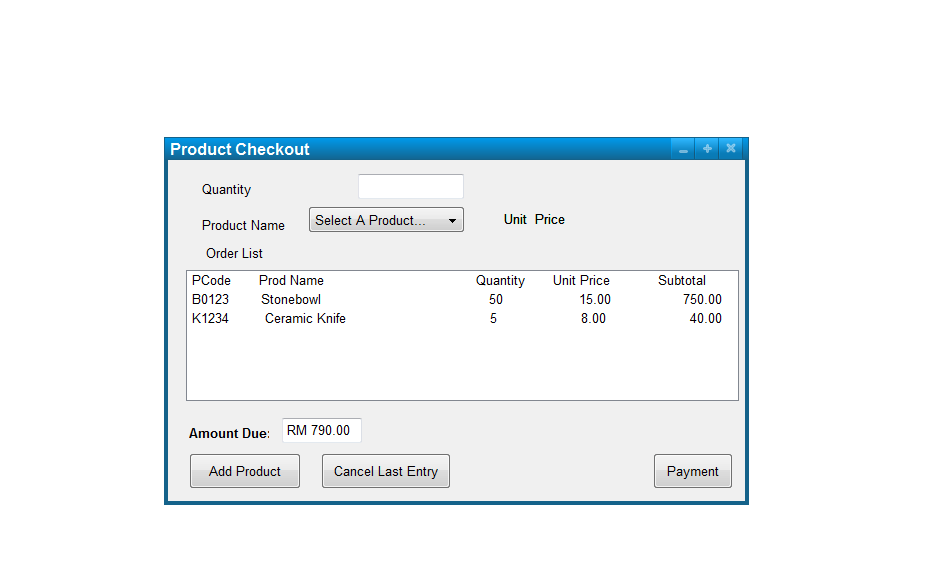
*http://pencil.evolus.vn/Downloads.html*

Below is an example of a wire-framed screen using the Pencil software:

**Product Checkout Form**

ComboBox1

TextBox1

****

Label 3

Label 2

Label 1

ListBox1

Label 4

Button 1

Button 3

Button 2

|  |  |  |
| --- | --- | --- |
| **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 |
| ComboBox1 | cboProduct | To allow selection of products from a list |
| TextBox1 | txtQuantity | To allow entering of product quantity needed |
| ListBox1 | lstItems | To display the list of items purchased |
| 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 |