|  |
| --- |
| Software Development Unit 3 Outcome 1Programming Folio School Assessed Coursework (SAC) Part 2 |

## **This SAC will consist of 4 Folio Tasks over an extended period of time.**

## U3 SAC 1:  Programming Folio (Contributes 10% of subject assessment)

## **Part 1:  T1W6 FRI 5 March (Double)**

## **Part 2:  T1W8 FRI 19 March (Double)**

## **Part 3:  T2W2 FRI 30 April (Double)**

## **Part 4:  T2W4 FRI 14 May (Double)**

**Outcome statement**

*On completion of this unit, the student should be able to interpret designs and apply a range of functions and techniques using a programming language to develop working modules.*

**Task Conditions**

1. **Allowed resources:** Teacher-provided solution designs, open book
2. **Time allocated to this task:** 2 periods (Double) Friday 19 March. **100 minutes**
3. **Marks allocated:** 21  
   **Location of files:** The zipped Visual Basic Project folder is to be submitted for marking Under:
4. **Submission of work: Learning Tasks:** “**Unit 3 SAC 1 Programming Folio 2**”.

**Task Outline**

Using the module requirements and provided designs, students are required to produce working software modules.

**Mr Mac’s Magnificent Motel**

|  |
| --- |
| **713 Vintage Neon Motel Sign Photos - Free & Royalty-Free Stock Photos from  DreamstimeMr Mac’s Magnificent Motel** is a newly established family business in the leafy suburbs of Macville. Mr Mac has requested that you add a user interface form that allows for the registering (storing in an external file) and displaying of customer detail. |

**Functional Requirements**

|  |  |
| --- | --- |
| **FR** | **Description** |
| **FR01** | A bottom on the main form opens up a new form “AddNewCustomer” |
| **FR02** | New customer details can be entered via the “AddNewCustomer” form: Given Name, Surname, Street Address, Suburb, Postcode, Mobile phone number, customer rating [Silver, Gold, Platinum] |
| **FR03** | All user input customer details are validated and error messages are provided if the data is not valid. |
| **FR04** | A button allows the user to append the new customer details the “CustomerDetails” textfile. |
| **FR05** | The user is able to click a button to display all customer details in the “CustomerDetails” textfile. |
| **FR06** | The user is able to click a button to display all customer details in the “CustomerDetails” textfile that are from a user-determined state (for example, Victoria) |
| **FR07** | Can display the total number of customers that have a “Platinum” rating. |
| **FR08** | Can Display the Names of customers that have a “Platinum” rating. |

**Non-Functional Requirements**

|  |  |
| --- | --- |
| **FR** | **Description** |
| **NFR01** | The application form size must fit the dimensions of a laptop |
| **NFR02** | The user interface must be intuitive for the user |
| **NFR03** | Font styles and colours must present a professional appearance |

**Constraints**

|  |  |
| --- | --- |
| **FR** | **Description** |
| **C01** | The solution must be produced on the School laptop |
| **C02** | You must use the Visual Basic .NET programming language |
| **C03** | You must complete the task within 100 minutes. |

**IPO Chart**

|  |  |  |
| --- | --- | --- |
| **Input** | **Processing** | **Output** |
| * Customer Given Name * Customer Surname * Customer Street Address * Customer Suburb * Customer Postcode * Customer Mobile number | * Append customer details to “CustomerDetails” textfile * Calculate the Number of customers who have a “Platinum” rating | * Display a list of all customers and their details in the “CustomerDetails” textfile * Display a list of customers from user-selected Australian States. * Display the names and total number of customers that have a “Platinum” rating. |

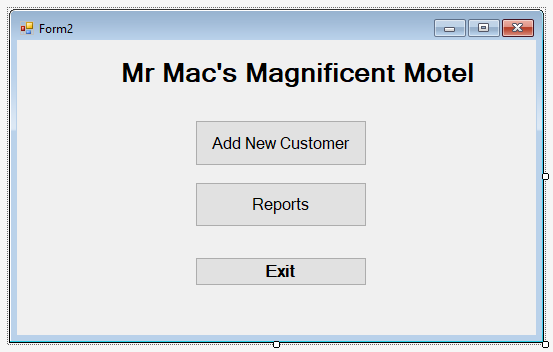
**Site Map**

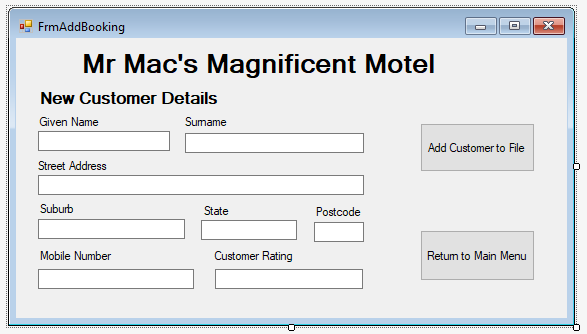
Main Form

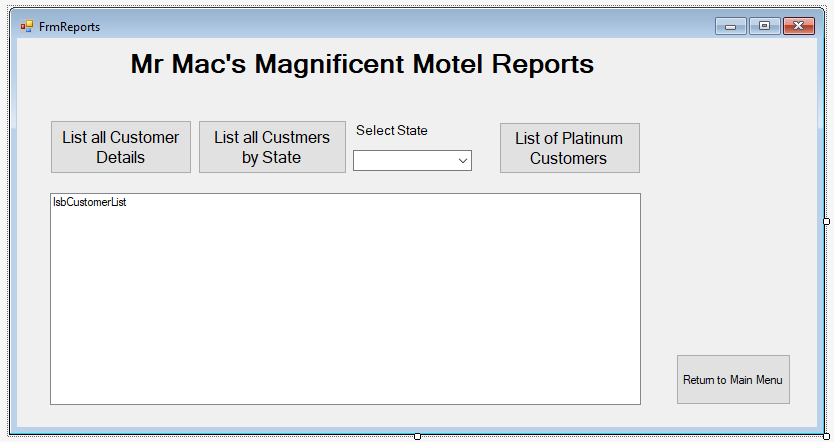
Add New Customer Form

Report Form

**User Interface Mock-ups**







**Sample Output**

All Customers on File Report

==============================================

David Smith, 34 Smith Street, Smythersville, VIC, 3333, 04567894, Silver

Trevor Treloar, 444 Treenth Road, Hawthorne, NSW, 2122, 0414567854, Gold

Todd Gakk, 5/55 Five Street, Hiveville, QLD, 7654, 0436854343, Platinum

Nugyen Vong, 88 Eight Avenue, VIC, 3587, 0412987555, Platinum

==============================================

VICTORIAN Customers on File Report

==============================================

David Smith, 34 Smith Street, Smythersville, VIC, 3333, 04567894

Nugyen Vong, 88 Eight Avenue, VIC, 3587, 0412987555

==============================================

PLATINUM Customers Report

==============================================

Todd Gakk

Nugyen Vong

==============================================

There are 2 Platinum customers.

**Data Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Format / Example** | **Purpose** |
| custGivenName | String | Todd | To store the first name of a customer |
| custSurname | String | Gakk | To store the surname of a customer |
| custStreetAddress | String | 44 Smith Street | To store the street and number details of a customer |
| custSuburb | String | Hawthorn | To store the suburb of a customer |
| custPostCode | integer | 3122 | To store the postcode of a customer |
| custMobileNum | String | 0412666666 | To store the mobile phone number of a customer |
| custRating | String | Silver, Gold, Platinum | To store the rating of a customer |

**Object Descriptions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Object** | **Format** | **Description** |
| txbCustGivenName | textbox | Font: Microsoft Sans Serif, Size: 8; Regular | For user to enter customer’s first name |
| txbCustSurname | Textbox | Font: Microsoft Sans Serif, Size: 8; Regular | For user to enter customer’s surname |
| txbCustStreet | Textbox | Font: Microsoft Sans Serif, Size: 8; Regular | For user to enter customer’s street address |
| txbCustSuburb | Textbox | Font: Microsoft Sans Serif, Size: 8; Regular | For user to enter customer’s suburb |
| txbCustPostcode | Textbox | Font: Microsoft Sans Serif, Size: 8; Regular | For user to enter customer’s postcode |
| txbCustState | Textbox | Font: Microsoft Sans Serif, Size: 8; Regular | For user to enter customer’s state |
| txbCustMobileNum | Textbox | Font: Microsoft Sans Serif, Size: 8; Regular | For user to enter customer’s mobile number |
| txbCustRating | Textbox | Font: Microsoft Sans Serif, Size: 8; Regular | For user to enter customer’s rating |
| lblCustGivenName | Label | Text: Customer Details  Font: Microsoft Sans Serif, Size:10; Bold | Form label |
| lblCustSurname | Label | Text: Customer Details  Font: Microsoft Sans Serif, Size:10; Bold | Form label |
| lblCustStreet | Label | Text: Customer Details  Font: Microsoft Sans Serif, Size:10; Bold | Form label |
| lblCustSuburb | Label | Text: Customer Details  Font: Microsoft Sans Serif, Size:10; Bold | Form label |
| lblCustPostcode | Label | Text: Customer Details  Font: Microsoft Sans Serif, Size:10; Bold | Form label |
| lblCustState | Label | Text: Customer Details  Font: Microsoft Sans Serif, Size:10; Bold | Form label |
| lsbReport | ListBox | Font: Microsoft Sans Serif, Size: 8; Regular | To display the requested report output |
| btnAddNewCustomer | Button | Text: Customer Details  Font: Microsoft Sans Serif, Size:10; Bold | Button on the main form which takes the user to the Add New Customer form and hides the other form |
| btnReports | Button | Text: Room Type  Font: Microsoft Sans Serif, Size:10; Bold | Button on the main form which takes the user to the Reports form and hides the other form |
| btnExit | Button | Text: Breakfast Selection  Font: Microsoft Sans Serif, Size:10; Bold | Exits the application |
| btnAddCustomerToFile | Button | Text: Room extras (Select your options)  Font: Microsoft Sans Serif, Size:10; Bold | Appends the new customer details to the customer file |
| btnReturnMainMenu | Button | Text: Room Request Cost  Font: Microsoft Sans Serif, Size:14; Bold | Button that returns the customer to the Main Menu and hides the other form |
| BtnListallCustomerDetails | Button | Text: Room Request Cost  Font: Microsoft Sans Serif, Size:14; Bold | Lists all customer details in the customer file |
| btnListCustByState | Button | Text: Room Request Cost  Font: Microsoft Sans Serif, Size:14; Bold | Lists the customer details of those who live in a particular state |
| btnListPlatinumCust | Button | Text: Room Request Cost  Font: Microsoft Sans Serif, Size:14; Bold | Lists the total number of customers (and their names) who have a Platinum rating. |
| cbxStates | ComboBox |  | Allows the user to select the state for which the customers come from |

**Pseudocode**

**ALGORITHM** AddNewCustomer

**BEGIN**

**INPUT** CustGivenName, CustSurname, custStreetAddress, custSuburb, custPostCode, custMobileNum, custRating

**Validate input**

**IF** input is valid **THEN**

**WRITE** data to CustomerFIle

**END IF**

**END**

**ALGORITHM** ListAllCustomerDetails

**BEGIN**

**OPEN** CustomerFile

**DO WHILE** not end of customerFile

**READ** CustGivenName, CustSurname, custStreetAddress, custSuburb, custPostCode, custMobileNum, custRating

**OUTPUT** CustGivenName, CustSurname, custStreetAddress, custSuburb, custPostCode, custMobileNum, custRating

**END WHILE**

**CLOSE** CustomerFile

**END**

**ALGORITHM** ListAllCustomersByState

**BEGIN**

**INPUT** State

**OPEN** CustomerFile

**DO WHILE** not end of customerFile

**READ** CustGivenName, CustSurname, custStreetAddress, custSuburb, custState, custPostCode, custMobileNum, custRating

**IF** custState = State **THEN**

**OUTPUT** CustGivenName, CustSurname, custStreetAddress, custSuburb, custState, custPostCode, custMobileNum, custRating

**END IF**

**END WHILE**

**CLOSE** CustomerFile

**END**

**ALGORITHM** ListPlatinumCustomers

**BEGIN**

NumPlatinumCust 🡨0

**OPEN** CustomerFile

**DO WHILE** not end of customerFile

**READ** CustGivenName, CustSurname, custStreetAddress, custSuburb, custState, custPostCode, custMobileNum, custRating

**IF** custRating = “Platinum” **THEN**

NumPlatinumCust= NumPlatinumCust + 1

**OUTPUT** CustGivenName, CustSurname

**END IF**

**END WHILE**

**OUTPUT** NumPlatinumCust

**CLOSE** CustomerFile

**END**

**Testing Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test no.** | **Test Item** | **Test data** | **Expected results** | **Actual results** |
| 1 | Main Menu **Exit** button prompts confirmation then closes program | Click Exit | MsgBox: “Do you really want to exit”  Then close | As Expected |
| 2 | **Switch** **form** buttons function  switches to FrmAddBooking  switches to FrmReports  switches to FrmExit  (applicable to all switch form buttons [Add New Customer, Return to Main, Reports] as uses same code structure) | Click Add New Customer  OR  Reports  OR  Exit to Main Menu | Switches to FrmAddBooking  Switches to FrmReports  Switches to FrmMainMenu | As Expected |
| 3 | **Validates** **Existence** for:  Given Name  Surname  Street Address  Suburb  State  Postcode  Mobile Number  Customer Rating | E.g. (Given Name = Nothing)  Surname = Nothing  Street Address = Nothing  Suburb = Nothing  State = Nothing  Postcode = Nothing  Mobile Number = Nothing  Customer Rating = Nothing | MsgBox: “Enter a [given name]”  Or  surname/  street/  suburb/  state/  postcode/  mobile/  rating] | As Expected |
| 4 | **Validates** **Existence** for:  Given Name  Surname  Street Address  Suburb  State  Postcode  Mobile Number  Customer Rating | E.g. (Given Name = 321)  Surname = Nothing  Street Address = Nothing  Suburb = Nothing  State = Nothing  Postcode = Nothing  Mobile Number = Nothing  Customer Rating = Nothing | MsgBox: “Enter letters for given name”  Or  surname/  street/  suburb/  state/  postcode/  mobile/  rating] | As Expected |
| 5 | **Validates** **Range** for:  Postcode  MobileNumber | E.g. (Postcode= 31231)  Surname = Nothing  Street Address = Nothing  Suburb = Nothing  State = Nothing  Postcode = Nothing  Mobile Number = Nothing  Customer Rating = Nothing | MsgBox: “Input a valid postcode number with 4 digits”  Or  Mobile/ | As Expected |
| 6 | Displays All Customer Details | Click Button | Displays CustGivenName, CustSurname, custStreetAddress, custSuburb, custPostCode, custMobileNum, custRating | As Expected |
| 7 | Displays By State | Click Button | Displays CustGivenName, CustSurname, custStreetAddress, custSuburb, custPostCode, custMobileNum, custRating of selected state only | As Expected |
| 8 | Displays Platinum Customers | Click Button | Displays CustGivenName, CustSurname of platinum only and number of platinum | As Expected |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| 11 |  |  |  |  |
| 12 |  |  |  |  |

|  |  |
| --- | --- |
| **Test no.** | **Image Proof** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 | Etc |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |

**Marking Scheme**

Each task will be assessed using the provided performance descriptors and table below.

|  |  |
| --- | --- |
| **Criteria / Skills** | **Marks Available** |
| Interpretation of designs to produce working modules | **2** |
| Data types and structures used | **2** |
| **Use** appropriate processing features of a programming language to develop working modules | **8** |
| **justify** appropriate processing features of a programming language to develop working modules | **2** |
| Develop and apply suitable validation techniques | **3** |
| Develop and apply suitable testing and debugging techniques using appropriate test data | **2** |
| document the functioning of modules and the use of processing features through internal documentation | **2** |
| **TOTAL** | **/21** |

VCE Applied Computing: Performance Descriptors

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SOFTWARE DEVELOPMENT UNIT 3 OUTCOME 1 - SCHOOL-ASSESSED COURSEWORK** | | | | | |
| **Performance Descriptors - typical performance in each range** | | | | | |
| ***Unit 3***  ***Outcome 1***  ***Interpret teacher-provided solution requirements and designs, and apply a range of functions and techniques using a programming language to develop and test working software modules.*** | **Very low** | **Low** | **Medium** | **High** | **Very high** |
| Limited interpretation of solution requirements and designs to develop working modules. | Some interpretation of solution requirements and designs to develop working modules. | Sound interpretation of solution requirements and designs to develop working modules. | Most solution requirements and designs are interpreted accurately to developing working modules. | All solution requirements and designs are interpreted accurately to developing working modules. |
| Limited selection and use of data types and data structures. | Some selection and use of appropriate data types and data structures. | Sound selection and use of data types and data structures to develop working modules. | Detailed selection of relevant data types and data structures to develop working modules. | Comprehensive selection of relevant data types and data structures to develop working modules. |
| Limited selection and use of processing features of the programming language to develop some working modules. | Some selection and use of appropriate processing features of the programming language to develop some working modules. | Sound selection and use of appropriate processing features of the programming language to develop some working modules. | Most processing features of the programming language have been selected and used to develop all working modules. | Comprehensive selection and use of relevant processing features of the programming language to develop all working modules. |
| Limited explanation of how the selected processing features are used to develop working modules. | Some justification and explanation of how the selected processing features are used to develop working modules. | Sound justification and explanation of how the selection of appropriate processing features are used to develop working modules. | Detailed justification and explanation of how the selection of appropriate processing features of the programming language are used to develop working modules. | Comprehensive justification and explanation of how the selection of appropriate processing features of the programming language are used to develop working modules. |
| Limited data validation techniques are applied to check the reasonableness of some input data. | Some data validation techniques are effectively applied to check the reasonableness of some input data. | Sound use of data validation techniques are effectively applied to check the reasonableness of input data. | Detailed use of relevant data validation techniques are applied to efficiently and effectively check the reasonableness of all input data. | Comprehensive use of relevant data validation techniques are applied efficiently and effectively to check the reasonableness of all input data. |
| Limited range of test data is expressed in a testing table, with incomplete or missing results. | Some testing of test data is expressed in a testing table with actual output stated. | Sound range of testing of test data is expressed in a testing table, with both expected and actual output stated and some evidence of debugging. | Detailed use of test data is expressed in a testing table, with both expected and actual output stated with evidence of debugging. | Comprehensive use of test data is expressed in a testing table, with both expected and actual output stated, and showing detailed evidence of debugging. |
| Limited internal documentation with few comments regarding the use of the selected processing features. | Some internal documentation with comments regarding the functioning of modules and the use selected processing features. | Sound use of internal documentation with comments regarding the functioning of modules and the use of selected processing features. | Most software modules include detailed internal documentation regarding the functioning of modules and use of selected processing features. | All software modules include comprehensive internal documentation regarding the functioning of modules and use of selected processing features. |

KEY to marking scale based on the Outcome contributing 100 marks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Very Low 1–20 | Low 21–40 | Medium 41–60 | High 61–80 | Very High 81–100 |