**Description: OC_Masterbrand_CMYK.eps**

in partnership with

****

ICT40515 Certificate IV in Programming

Develop a UI to interact with a database

**SP6/Module 9 Assessment**

ICTPRG403 Develop data-driven applications

ICTICT420 Develop client user interface

ICTPRG410 Build a user interface

Assessment: 33130/02

*© Open Colleges Pty Ltd, 2016*

*All rights reserved. No part of the material protected by this copyright may be reproduced or utilised in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the copyright owner.*

*All terms mentioned in this text that are known to be trademarks or service marks have been appropriately capitalised. Use of a term in this text should not be regarded as affecting the validity of any trademark or service mark.*

**Assessment task 33130/02**

## Introduction

This assessment will test your skills and knowledge gained through completing the learning and activities in **Module 9: Develop a UI to interact with a database.**

**This is a three-part assessment – the first assessment part (33130/01) will need to be completed and submitted to your trainer for marking and obtain feedback *before* you move onto this second assessment (33130/02).**

In the first part of the assessment, you will analyse a requirements brief to design and develop user interfaces for a number of screens.

In the second part, you will write code for the screens developed in the first part of the assessment to connect and interact with a database.

In the third part, you will answer a number of questions which will test your knowledge on developing and building a user interface, in addition to developing a data driven application.

**Note – you must successfully complete ALL assessments (33130/01, 33130/02 and 33130/03) to achieve competency in:**

***ICTPRG403 Develop data-driven applications***

***ICTICT420 Develop client user interface***

***ICTPRG410 Build a user interface***

**CASE STUDY**

|  |
| --- |
| **Acme Insurance Company**  https://s-media-cache-ak0.pinimg.com/564x/11/c2/92/11c29263df7ad88a1047a0bb9e5dab02.jpg  **Source:** [**Pinterest**](https://au.pinterest.com/pin/411657222161967801/)  **Outline**  The Acme Insurance Company sells a range of insurance products, from life insurance to car insurance to home and contents insurance. You have previously done some work for the company by developing a tax calculator to calculate the tax payable by their employees.  Mr Wile E., the CEO of the Acme insurance Company, has decided that he would like a database developed to store details of his customers, their insurance products and which products have been purchased by each of their customers. Each sale can only have one product associated to it. He would like an application developed so that information is easily maintained in the database.  The database, which includes relevant stored procedures and data, has already been developed. You can download the database script [here](https://app.box.com/s/opfswywdnrcm9zw4epbzhuzw3dsk4c6t/1/12036199155/100808205215/1). You could either use the stored procedures that are already developed or create new procedures as per your requirements. In SQL Server Management Studio, run this script to create the Acme Database, tables, stored procedures and populate the tables with data. Below is the data schema:    Your final task is to write the code that facilitates interaction between the UI screens you designed in the previous task and the Acme database. When deleting a record ensure that that particular record doesn’t have any dependent information in other tables to ensure data integrity. |

**Tasks 1–2**

**Data driven application**

Read the case study above thoroughly. Highlight, make notes, or extract the important information about the requirements put forth by the Acme Insurance Company. In the following tasks, you will need to use this information in creating the requested application that interacts with the Acme database.

|  |
| --- |
| **Task 1: Develop classes and database connection method**   1. If you did not create the *Acme* project in the first assessment, from Visual Studio, create a Windows Forms application. Name the project *Acme* and the solution *M9Assessment*. Create a class that contains the database connection string that will be used throughout your project to connect to the database. You are to use the SQLconnection method. This class is to be placed in a folder named Data Access Layer. Add appropriate comments to your code. Be sure to use your local SQL Server database path in your connection string. 2. Create a class and appropriate constructors for Categories, Customers, ProductTypes, Products and Sales. These classes are to be placed in a folder named Business Logic Layer. |

|  |
| --- |
| **Task 2: Write code to interact with a database**   1. For the Maintenance screen of each Categories, Customers, Product Types, Products and Sales, write the code to connect to the Acme database and display the rows from the relevant table. Be sure to cater for exceptions and add appropriate comments to your code. Rows retrieved from the database are to be placed into the relevant class before being outputted to the screen. 2. Write the relevant code for each of the Add, Update, Delete and Search events. Use stored procedures to check if a row can be deleted and to delete rows which can be deleted. Ensure you use Transactions before executing stored procedures. For Search, you are to search a minimum of two criteria for Customers, Products and Sales as well as display all rows. 3. For each of these Maintenance screens, write the code to validate input data, connect to the database and insert and update relevant rows. Be sure to cater for exceptions and add appropriate comments to your code. Use stored procedures to insert and update table rows. Ensure you use Transactions before executing stored procedures. 4. For the Sales maintenance screen, display all relevant information as well as the premium payable. For yearly payments display the Products.YearlyPremium, for monthly payments, display Products.YearlyPremium / 12 and for fortnightly payment, display Products.YearlyPremium / 26. 5. For each of these Maintenance screens, write the code to connect to the database and filter rows based on user-provided search parameters. Be sure to cater for exceptions and add appropriate comments to your code.   **Submit:**   * **The final application (Make sure to test the application thoroughly for all requirements before submitting)** * **Script the Database that you have been using and submit it as a part of the evidence.** * **Screen shots of the working of the application.** * **All screen shots to be placed in a Word document titled 33130-02\_yourName\_studentNumber\_ScreenShots.docx. (For example, 33130-02\_JohnSmith\_17756433\_ScreenShots.docx), under appropriate headings.** |

**Next steps for you:**

You have now reached the end of this assessment.

Ensure that you have completed all of the above tasks. Use the checklist to double check that you have submitted everything required for this assessment.

Zip your Visual Studio project and upload to OpenSpace. Name your file 33130-02\_yourName\_studentNumber.zip. (For example, 33130-02\_JohnSmith\_17756433.zip).   
Your trainer will provide you with feedback for this assessment – ensure that you read this feedback carefully.

**Checklist of items to submit as evidence for this assessment:**

|  |  |  |
| --- | --- | --- |
| **PART A** | | |
| TASK 1 | Database connection class | 🞏 |
| TASK 1 | Classes for each database table | 🞏 |
| TASK 2 | Maintenance screens and associated code to interact with the database | 🞏 |
| TASK 2 | Add/Update and Search screens and associated code to interact with the database | 🞏 |
| TASK 2 | Document of screen shots | 🞏 |
| **Name each assessment file appropriately and submit for marking.** | | |