**Software Pathway** - **Software Development Project**

**The Purpose of the Project.**

The purpose of this project is to provide you with the experience of working through the entire lifecycle of creating software. You will do this by creating two versions of CRUD software applications.

The first version will be written using C# and the .Net Framework and will save data into a relational database. The second version will be written in Python and store the data as serialised objects in programmer-defined files.

You will design, plan, code and test two versions of this application.

**What is a CRUD Application?**

CRUD is an acronym used in computer programming that refers to the four functions necessary to implement persistent application data storage: **create, read, update** and *delete*.

Persistent storage refers to any data storage component that retains data after the device is powered off, such as a hard disk or a solid-state drive. In contrast, random access memory and internal caching are two examples of volatile memory which contain data that will be erased when the device loses power.

**Explaining CRUD Operations**

Organisations that keep track of customer data, accounts, payment information, health data, and other records require data storage hardware and applications that provide persistent storage. Data is frequently stored in a database, an organised collection of data. There are many types of databases: hierarchical databases, graph databases, and object-oriented databases, to name a few. The most widely used type of database is a relational database, which consists of data stored in rows and columns and connected to other tables with complementary information by a system of keywords that includes primary keys and foreign keys.

The CRUD acronym identifies all the primary functions inherent to relational databases and the relational database management applications used to manage them, such as Microsoft SQL Server, MySQL, Oracle Database and others.

It is also possible to implement all the CRUD applications' functionality by storing data in programmer-defined files without using a relational database.

An example project is provided which can be used as a guide.

**What Software Developers Do**

In general, the software developers' job is to convert problem solutions into instructions for the computer. That is, the developer prepares the instructions of a computer program and runs those instructions on the computer, tests the program to see if it is working properly, and makes corrections to the program. The developers may also document a program. These activities are all done for the purpose of helping a user fill a need, such as paying employees, billing customers, or admitting students to college.  
  
The development activities just described could be done as solo activities but typically a developer interacts with a variety of people. For example, if a program is part of a system of several programs, the developer coordinates with other developers to make sure that the programs fit together well. If you were a programmer, you might also have coordination meetings with users, managers, systems analysts, and with peers who evaluate your work-just as you evaluate theirs.  
  
Let us turn to the programming process.

**The Programming Process**  
Developing a program involves steps like any problem-solving task. There are five main steps in the programming process:

* Defining the problem
* Planning the solution
* Coding the program
* Testing the program
* Documenting the program

**Example Project**

Here is an example project to provide guidance on what you need to achieve.

Figure 1 below shows you a data entry form, below this is a list of the components used to create it and a description of the code needed to make it work.



*Figure 1 Example Data Entry Form*

This screen provides the functionality to enable the user to do the following:

* Display individual records from the database or file.
* Add a new record to the database or file.
* Delete the displayed from the database or file.
* Edit the displayed and save the updates to the database or file.
* Cancel changes to or a record.
* Search records (this functionality for this is on another form)

The detailed steps to create data entry form shown above are as follows:

* Create a data form that shows a single record to appear like the form shown above and includes:
* Add a label for the form heading in this example 'Speedy Car Hire' in bold type with a different and larger font.
* Add six controls and associated labels to display the data for the record.
* Add navigation controls to move to the first, previous, next and last records.
* Add a control to hold a record count in the form *n* of *nn.*
* Add six buttons for Update, Add, Delete, Search, Cancel and Exit adding the accelerator keys shown.
* Change the background to a suitable colour (not the default colour).
* Ensure the data input controls receiving focus in an appropriate order.

Then the following functionality was added with code:

|  |  |
| --- | --- |
| 1 | Set the title bar to display the current date |
| 2 | Make sure that the formats of the displayed fields are suitable. |
| 3 | Set up the program so that when the data entry form is loaded the dataset is loaded automatically and the data for the first record is displayed in the controls. |
| 4 | Write code for the Update, Add, Delete and Cancel buttons. |
| 5 | Write code for the controls to move to the first, previous, next and last records. |
| 6 | Write code in a method/function to display the total record count and current record number each time one of the navigation buttons Is used. This will be displayed in the control associated with the navigation buttons as shown on the form. |
| 7 | Insert the code required to handle errors for database/file access which prevents run-time errors. |
| 8 | Write code for the Exit button to terminate the program. |
| 9 | Write code for the Search button to open a second that will provide search functionality |

You also need to define the table(s) for your database or the attributes for the object that will store the data an example is shown in the table below:

|  |  |  |
| --- | --- | --- |
| **Field or Attribute Name** | **Data Type** | **Field Length** |
| VehicleRegNo (Primary Key) | Text | 10 |
| Make | Text | 50 |
| EngineSize | Text | 10 |
| DateRegistered | Date | dd/mm/yyyy |
| RentalPerDay | Currency, 2 decimal places | |
| Available | Logical (Yes/No) | 1 |

Example Search Form

An example search form is shown in Figure 2 below.



*Figure 2 - Search Function*

The steps to create this form are set out below.

|  |  |
| --- | --- |
| Create a Search form similar to the form shown above and includes:  a group from on the form to contain the following:  two combo boxes to choose a field to be searched and search operator with associated labels  A text box for data entry with an associated label  A Grid control to display the results of the search.  A Button to run the search and a Button to close the search box.  Make sure that the formats of the displayed fields are appropriate. | |
| 1 | Save the form as with a proper name (use naming conventions). |
| 2 | Set the title bar to display the word "Search" |
| 3 | Write code in to do the following tasks as the form loads:   * Populate field combo with the field names shown * Populate Operator combo with the following operator symbols, each one as a single list item: =, <, >, <=, >= |
| 4 | Write code for the Run button to match the search criteria entered using the combo boxes and the value in the data entry text box.  The fields shown should be displayed in the data grid. The search should be run only if data exists in **all** three query criteria controls. A criteria string that is **not** matched by any record **must** return nothing. |
| 5 | Write code for the Close button to hide the form and return to the data entry form. |

**Testing and Documenting the Program**

*In this task, you are required to test and document the software created.*

|  |  |
| --- | --- |
| 1 | Create test data and expected results to test all program functions: |
| 2 | Prepare a test plan and test the software. Compare the actual results to the expected results keeping a log for each test that identifies any discrepancies between actual and expected results and records any amendments to correct errors. Use debugging tools to help locate and remove errors. |
| 3 | Run the program file to demonstrate the software. |
| 4 | Produce technical documentation to describe the connection details and the purpose of the software. |
| 5 | You should check that the program produced meets the following requirement:  The program must conform to the design specification.  The program uses the most appropriate controls and events.  Meaningful names are used for constants, variables, objects, forms and controls using a consistent naming convention. |