Assignment 2: C# Application & Test Report

The Brief

Your small development team of (ideally) three people has been asked to implement and test the Human Resources Information System desktop application. Your software product will be a database-backed desktop application with a Windows Presentation Foundation (WPF) graphical front end, implemented in C#. As part of your development efforts you will prepare and apply a small collection of test cases to verify that the completed application meets some of the key requirements agreed with the client.

Submission

Your progress will be assessed, and feedback given, during your Week 11 tutorial. Additional details of this checkpoint will be provided after the mid-semester break.

By **1500 (3pm) Friday October 16th** submit deliverables 1 and 2 to the *Assignment 2 – C# Application & Test Report* assignment folder on the unit's MyLO site. Each member should also complete the peer review activity via Buddycheck on MyLO.

Team Formation

Your teammates should be selected from other students in your tutorial group whom you have not worked with in Assignment 1. Once the grouping has been approved by your tutor, each member should go to the Groups page on the unit's MyLO site, select the same group from the list, and add themselves. A fourth member, if required, can only be added by your tutor.

Deliverables

There are two deliverables in this assignment:

1. a zipped VisualStudio project that contains the WPF-based application that *must be submitted to the assignment folder*, accompanied by
2. a completed test report in PDF, which must also *be submitted to the assignment folder*.

The starting point for developing the application

A standard OO model for the system (some scenarios, all class diagrams and some sequence diagrams) will be released after the Assignment 1 due date, along with details of the database schema. You may then use your group members' joint experience of developing their own OO models to determine the rest of your design. You may also deviate from the standard design as you see fit. This will give different development teams some freedom in their implementation choices.

OO Model versus WPF

In the OO model each view is a separate class, which in WPF corresponds to defining a UserControl for each major view. Because this may complicate some of the event handling code, it is acceptable in the assignment to place separate views directly within the main window. Note, however, that the HD level of the *Use of WPF* assessment criterion requires that at least one user control be defined, and that defining your own controls will actually make each individual source file easier to manage.

WPF is not available in MacOS; **development of the GUI and the final stages of the software will need to be completed in Windows**. Please email me (Julian.Dermoudy@utas.edu.au) if this will present problems.

Development approach

Because the final application requires knowledge of the C# language, how to communicate with a database using C# (and LINQ) and how to design and construct GUIs using WPF, and because these topics are to be covered over a period of some weeks, as foreshadowed in the Week 6 lecture, we recommend that you take a prototyping approach to the development of this application. This will allow you and your team to make useful progress on parts of the system even before you have obtained the skills required to implement the whole system. (Note that this will likely entail *some* manual copying of source files between projects as you migrate from a console application to a WPF application.) We suggest the following implementation stages (you may of course adopt a different development path if you find it suits your team better):

Stage	Timeframe (semester week)	Application type & functionality	Notes
1	8–9	Console application with entity (data) classes, partially-implemented control classes and at least one database adapter class. One or more driver classes containing Main() that can test features such as filtering of the staff or unit lists.	1. *May* have a GUI, but this should be no more than necessary to exercise/test implemented functionality (should be able to throw this GUI away later). 2. Database adapter will need to return artificial data.
2	9–10	As above but using live data, which will substantially ease the burden on you to create artificial test data. Draft test cases that are complete in all details apart from test methods (which will likely still be high-level until the GUI is built).	(1) as above
3	11–13	WPF application with custom controls for different views. Control classes updated to know about GUI components and event handlers connected. Test cases completed and applied to your application.	1. Transition will be easiest by creating a new WPF Application project and importing the pre-existing class files into it. 2. **This cannot be completed on an Apple computer running MacOS** as WPF is not available on that platform.

OO Packages and C# Namespaces

As C# applications have a top-level namespace, which was not part of the OO Design, we recommend that each package in the OO Design becomes a nested namespace within your application's top-level namespace (which should be HRIS). Note that the namespace and project name do not have to be the same, even though Visual Studio will initially make them the same.

The School Database (aka HRIS Database)

The case study MySQL database is available via the following settings:

Database: kit206
User Id: kit206
Password: kit206
Data Source: alacritas.cis.utas.edu.au

You can also browse the data via *phpMyAdmin* (append that to the end of the server name). As the database is shared it will not appear in your own list of databases on alacritas if you log in under your own user name. An EER Diagram for the database plus some details on enumerated value columns accompanies this document. If you are not on campus, you will need to use VPN (https://utas1.service-now.com/selfservice/?id=kb_category&kb_category=c06bbabbdbd86bc08bf8413b3a9619bb) to access the database.

Data quality

The database is currently live and contains fictitious data. The data are consistent in that foreign key relationships are all valid, although initially there will not be much information in the database. More entries, or more realistic entries, may be added later in semester, but this is not guaranteed.

The Testing Report

Concurrently with development of the application you will create a test plan containing four test cases to verify the following subset of requirements in the HRIS RTM:

- SWC 1 (Entry 1 in the RTM)
- UC8_User_views_StaffList
- UC16_User_selects_StaffDetails
- UC24_User_selects_Unit

Each test case will be based on the **Test Report Template**, which is available on MyLO. Each test case that is based on a use case should include at least two *methods* by which the requirement will be tested. NTH features of use cases should also be tested and so will provide additional methods, even if they were not implemented; the quality of the test case will be assessed, not the one line outcome of whether it passes or not. The test cases for the SWC will likely need only one, brief method in order to be tested.

Ensure that all instructional entries (*Repeat the following box…*, *<Extract the text…*, *Repeat Method and Outcome…*) and unused method–outcome rows are removed from your submitted test plan. The document should include a title and your team name on the first page; the first test case can start immediately below that.

When the application is complete, apply the test plan to it and record the outcome for each method. Remember that if the outcome is 'fail' then an explanation of when (which step) or how must also be recorded.

Sources of information for compiling the testing plan

Follow the procedures described in *Module 5 Use case-Based Testing* to derive the test cases for the indicated SWC and SW entries. The RTM will be your primary resource for identifying the criteria for each test case. The original requirements document may also provide some useful information. The structured scenarios you prepared during Assignment 1, in conjunction with the specific application being tested, will provide the basic structure of the testing methods in each test case, since they describe the user's actions in each use case.

Assessment

The assessment criteria for the final submission are available in an accompanying document, in Excel and PDF formats.

Your progress will also be assessed during your week 11 tutorial, which will contribute up to 5% to your final grade in the unit. The assessment criteria for this check will be released after the mid-semester break.

Peer assessment

Peer assessment will be completed as part of the tutorial in week 13. You will have the opportunity to provide both qualitative and quantitative feedback on your teammates' performance via Buddycheck on MyLO.

Revision history

2020-08-27   Initial release

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder