Autumn 2014

SOFT331 - Object-Oriented Web Application Development

Coursework 2014

1. Overall Aim

The overall aim of the assignment is that you should analyse and design a web application as described in the scenario section below and implement it using ASP.NET MVC.

You should note that a good data model design and application functionality are more important in this assignment than a decorative interface.

2. Key Dates

Assignment issued:	22nd October 2014
1. Demo sessions	W/C 8th December 2014
2. Deliverables as per section 2	30th January 2015 at 4pm

3. Key Deliverables

1. A working web application to be completed on your allocated XServe server space.
2. ~~A demonstration of your working web application.~~
3. One zip file submitted via DLE system on the SOFT331 DLE site containing the following:
    a. Source code files as part of the zip file.
    b. Analysis and design documentation as detailed below.

4. Deliverable details

Implementation

The implementation is required to be hosted on your XServe server space.   You are not to use your own hosting. The code residing on the XServe server space must be published code, that is that it has already been compiled it is not a direct copy of your source code.

~~*Demonstration*~~

~~A demonstration of the use of your application is also required. During the final two weeks of this autumn term there will be a 10 minute slot that you can book in advance. Dates will be Monday 8th December 2014 3-5pm, Thursday 11th December 2014 1-3pm, Monday 15th December 2014 3-5pm, Thursday 18th December 2014 1-3pm. Feedback will be given to you in the demo.~~

Source Code

The source code files must be uploaded as part of your zip folder. The source code must be saved as a Visual Studio project solution with the folder structure embedded so that when it is unzipped all the files and folders will be the correct places

Analysis and design

The analysis and design documentation must be submitted as a single zipped file along with the source code for your implementation and submitted via the DLE system on the SOFT 331 DLE site.

Please be aware it is extremely unwise to wait until the last minute to upload your work. There may be difficulties or problems if you do this and carry the risk of being capped to 40% for your work should it be late in any way.

As a minimum the following analysis and design documentation is **required**.

- Use case model (diagrams & descriptions).
- Activity diagrams
- Class diagram(s).
- Reflective report
- Walkthrough of application

The reflective report should be of around 1,000 words describing the analysis and design process you undertook in the production of your application. For example, this may be listing any documents or diagrams that you created during your work but did not submit, detailing any assumptions you have made with regard to the scenario or any other information that you acquired (eg: by asking for clarification). The report should include a description of the tasks your application achieves, whether all functionality was implemented, details of the database tables that have been created and discuss any areas that may be incomplete - explaining why.

The walkthrough of your application should contain screenshots and commentary. This will form the basis of your demo. You must clearly indicate any changes made after the demo as a result of feedback received.

5. Assignment / Module details

This is an individual assignment that contributes towards 70% of the module mark.

This coursework assesses the module learning outcomes:

1. Specify and document the design of an object-oriented web application.
2. Implement a web application using an object-oriented language.

6. Software

You must use ASP.NET MVC and C#. You must use the XServe instance of the Microsoft SQL Server for final deployment and not use a local database.

7. Marking

There is no specific allocation of marks to individual parts of this work. However, the following rubric will apply.

Deliverable	Fail	>40%	>50%	>60%	>70%
~~Demon-stration~~	~~Student has clearly made no effort to put together a coherent demonstration of their work. Student is not interested in the activity. Student cannot answer questions appropriately. Student appears not to understand their own code.~~	~~Student clearly understands own code. Application demonstrated as running.~~			
Source Code	Source code does not compile/run. Source code has substantial errors.	Source code is readable but only by an expert. Little consistency in naming conventions. Code is disorganised.	Majority of code properly formatted and covers coding conventions.	Code is fairly easy to read, formatted in consistent manner, named in logical, mostly consistent manner. Most code could be reused	Code well organised, easy to follow, adheres to expected conventions. Code easily reused and well documented. Code is efficient.

Deliverable	Fail	>40%	>50%	>60%	>70%
Documen-tation	Elements missing from the analysis and design. Reflection and walkthrough not complete.	Only basic approaches covered. Content does not generate confidence in the answer. Structure not that easy to follow but there are basic feastures.	Diagrams are mostly complete. Language lacks clarity or includes the use of jargon. Acronyms are not described. Content is inconsistent but some evaluation described.	Writing of a good standard. Demonstrates a good understanding. Answers written are clear and understandable.	Answers well written and comprehensive. Diagrams excellent. Clear mapping of documentation to source code and demo.

8. Scenario

1. *Online Ticketing System*

a. *Purpose*

The Talyllyn railway (http://www.talyllyn.co.uk/) wishes to sell tickets for their trains online.

Currently there are no facilities to buy tickets in advance. All tickets have to be purchased in the booking office at the station.

A web based application is required to allow travellers to purchase tickets in advance only. Tickets for the same day are not allowed. There should be an appropriate management interface to the application which allows manipulation of the data that is separate from the standard travellers interface.

You have been tasked with conducting the analysis, design and implementation of a prototype of the application.

b. *Users and functionality*

There are three key users of the application: Admin, Booking Clerk and the Traveller.

Admin

The admin is responsible for maintaining the data about the trains, the timetable and the fares. They are also responsible for extracting gift aid information. Key information would be:

- Steam engines (http://www.talyllyn.co.uk/details/locos)
- Type of train - e.g. Victorian, Santa Special, Halloween, Service
- Timetable (http://www.talyllyn.co.uk/timetable/2014)
- Fares (http://www.talyllyn.co.uk/fares)

Booking Clerk

The booking clerk is responsible for issuing tickets to those who telephone in advance. They will also need to view, edit or delete tickets where appropriate. Suitable control should be given to the deletion of any tickets as there would be payment information with it. (NOTE: Payment information is outside of the scope of this assignment.) The booking clerk will also need to extract key information about tickets sold. Key information is as follows:

- Number of tickets sold for a particular train.
- Type of tickets sold for a particular train (eg 37 Adult rovers, 35 are gift aided, 2 dogs etc).
- Specific disability support requests (eg Wheelchair space required)

Traveller

The traveller should be able to select the type of ticket they require in advance. They should not be able to purchase a ticket for the same day. They should be able to see which trains are available to them. The total number of seats on a train are 150 (which could be changed) and advance tickets may only have a certain number allocated to them which will be determined by the admin or booking clerk.

The traveller should also be able to request a wheelchair space. Only one wheelchair can travel on a train at any one time. Carers get an automatic 20% discount and disabled children travel free.

When a traveller purchases a ticket, they should be asked if they would like to gift-aid their ticket. If they say yes, they should be able to enter their post-code, their house number/name, and their name. **It is outside the scope of this system to link into any external API's for checking (eg with Royal Mail database)**. However, appropriate validation should be in place.

Outside Scope

Whilst the system should allow the traveller to click a button to complete the purchase, for the purposes of this piece of coursework, it is **outside** of the scope of the system to collect any payment details or to link to any third party payment systems.

Privacy policies, terms and conditions and returns policies as required by the e-commerce directives are also outside of the scope of the system.

Analytics are not required.

Integration with social media is not required.

No validation with external API's is required for the prototype - for example any postcode or address validation.

2. *Getting clarification*

You may find you need to clarify certain aspects of the scenario as you work on the assignment.

The first port of call will be the FAQ which will be maintained on the SOFT331 DLE site under assessment.