INSTITIÚID TEICNEOLAÍOCHTA, SLIGEACH
INSTITUTE OF TECHNOLOGY, SLIGO

School of Engineering and Design

Head of School: Mr Shane Fanning

Exam Series: Semester 2	**Academic Year 2014/2015**

Module Title: **Module Code:**

Rich Application Development 302 (RAD302) COMP07150

Programme Code(s):	Programme(s) Name(s)	Year(s)	FT /PT (if applicable)
SG_KSDEV_B07	B.Sc. in Computing in Software Development	3	FT
SG_KGDEV_B07	B.Sc. in Computing in Games Development	3	FT

Internal Examiner(s): Mr. John Kelleher

External Examiners: Dr. Tom Lunney

Instructions to Candidates

Time Allowed:	**3 Hours**
Number of Questions on Paper:	**1**
Number of Questions to be attempted:	**1**
Compulsory Question:	
Any Other Special Instructions:	**Upload at http://bit.ly/rad30x Ensure you comply with upload instructions below**

Question One

Prepare an ASP.NET MVC web application. A core set of requirements are outlined below but you are free to deliver on these requirements as you see fit. However, better solutions to the requirements will attract a higher score.

Directions

You will work with the Northwind database exposed through Entity Framework and a Web API. The user interface will be populated via API calls to the back-end.

1. Begin with a basic ASP.NET MVC web application – no user authentication is required so feel free to choose an empty template or one designed for intranet use.
2. Provide styling through the use of a CSS framework. Style the presentation of data and user interaction. This should include the use of a third-party jQuery plugin to enhance the interaction.
   [15 marks]
3. Create a data connection to the Northwind database. You will find a copy on Github at https://github.com/itsligo/sampledbs or the class site (http://class.jkelleher.me/rad302/schedule/week-5-6).
   Alternatively, you may be able to conect to a hosted Northwind database I have placed on Azure. Access credentials are:

   connection string='data source=pc7sy8gm9x.database.windows.net;initial catalog=nw;persist security info=True;user id=nwuser;password="capistrano1776;"

   Note that this user has limited access rights and so cannot browse the [master] database – as such, you will need to specifiy the nw database specifically in the connection string, if you choose to use this.
   [10 marks]
4. Create a database-first Entity Framework data model to represent the Northwind database. You need not create proxy classes for all tables – only the Customer and Orders tables need be provided.
   [10 marks]
5. Create an API controller to provide access to the data. Ensure this controller serves JSON data. This controller should reply to the following requests:
   a. api/Cust/GetCustomers – returns all Customers
   b. api/Cust/GetOrders(CustomerID) – returns Orders for CustomerID

   Ensure your Web API routing configuration supports the endpoints specified.
   [15 marks]
6. Employ the Repository Pattern to implement a data abstraction layer. This should use the Ninject MVC3 (or another more suited to your version of Visual Studio) Nuget package.
   Make the necessary adjustments to your API controller to ensure it uses the repository class.
   [15 marks]
7. Provide a simple interface listing some details (not all) of the first **ten** Customers. This should be achieved using Ajax calls to the localhost Web

API that you created earlier.
[15 marks]

8. Provide support to let the user select a listed Customer and have the first **ten** (if there are >10) Orders for that Customer listed alongside the list of Customers. As above, provide this using Ajax calls to your Web API.
[20 marks]

Note on API access

Should you not be successful in establishing the Web API in the above steps, you may try to use an Azure hosted API I have created which exposes Northwind. I cannot guarantee that Azure will be available but it may help with later parts of the exam. You'll find the API (& documentation) at http://nwapi.azurewebsites.net

Notes on Coding Quality

The quality of your answer to each of the above requirements will be reflected in the marks awarded. For example, appropriate use of Razor helper functions will often be a better alternative to basic html, resulting in more maintainable, robust code. Similarly, use of Ajax and/or partial views will attract significantly more marks.

Online Access policy

You are permitted to use online resources in a limited fashion. Plagiarism is not acceptable. You may use the Internet to look up details of function calls and argument types. Access to class sample code or your own exercises/CA based on same, is prohibited.

Resources Permitted

1. www.stackoverflow.com
2. www.msdn.microsoft.com/library
3. www.getbootstrap.com
4. http://api.jquery.com/
5. www.nuget.org
6. https://developer.mozilla.org/en-US/ (Mozilla Developer Network)
7. class.jkelleher.me (but not github-hosted solutions)

Upload Instructions

- Be sure to enable Nuget Package Restore (http://docs.nuget.org/docs/workflows/using-nuget-without-committing-packages) for your Solution to avoid saving large amounts of source code to your zip file.
- Remove the bin, obj and debug folders – these are regenerated on compiling
- Only submit one attempt at the solution
- Zip your code into a single file. Ensure that this has worked by checking the zip file size. If >5Mb, you must remove the un-necessary items mentioned previously. Unzip and check your zip file if unsure.
- Upload your solution to **http://bit.ly/rad30x** **AND** email a copy of the zip file to yourself. If you cannot do so, ask the invigilator for a USB stick (if available) and copy your solution onto it in a folder named as per your student ID. You may also push a repository of your code to Github or similar if unsure.
- Mail a copy of the zip file to your own email as a precaution and do not open.