IN-COURSE ASSESSMENT (ICA) SPECIFICATION

|  |  |
| --- | --- |
| Module Title:  **Web Apps and Services** | Module Leader:  **Zafar Khan** |
| Module Code:  **CIS2021-N** |
| Assignment Title:  **Element 1: Job Interview with CV (20%)**  **Element 2: ThAmCo Events Web App (80%)** | Deadline Date:  **Element 1: 2-Nov-2023**  **Element 2: 11-Jan-2024** |
| Deadline Time:  **4:00pm** |
| **Submission Method:**  Online (Blackboard) 🗹  Middlesbrough Tower 🞎 |

|  |
| --- |
| **Online Submission Notes:**   * Please follow carefully the instructions given on the Assignment Specification * When Extenuating Circumstances (e.g. extension) has been granted, a fully completed and signed Extenuating Circumstances form must be submitted to the School Reception or emailed to [scedt-assessments@tees.ac.uk](mailto:scedt-assessments@tees.ac.uk). |

FULL DETAILS OF THE ASSIGNMENT ARE ATTACHED  
INCLUDING MARKING & GRADING CRITERIA

# Web App and Services Development

## Introduction

The assessment for this module has two elements:

**Element 1 (20%) requires** you to prepare a CV and Cover Letter for a role as a software developer and then undertake a mock interview for the position.

**Element 2 (80%)** requires you to develop a business web application based on a provided skeleton application and write an evaluation report. This element will assess learning outcomes PTS2-3, RKC1-2 & PS1-2.

*Apprentices on the Digital and Technology Solutions course will normally undertake a negotiated work-based project that is agreed by the employer, apprentice and module leader/tutor. The Project will be tailored to the apprentice’s job role and the in-house software platforms available to the employer. However, where required the case study or sections of may be used.*

# Element 1: CV/Cover Letter and Job Interview

## Instructions

You will undergo a mock job interview for a software developer role either as first role or as a promotion. In preparation for the interviews, create a **CV** and **Cover Letter** for one of the available job roles (posted on Blackboard) and submit these to Blackboard as your Element 1 Submission.

Support materials to assist you with this are situated in the ***SCEDT Curriculum+*** Blackboard module in the folder ***Students Futures – Work on your employability*** >> ***Computing Employability Resources*** >> ***CVs, Covering letters and Interviews***. Interviews will take place during weeks 7 - 9. A schedule will be published during week 6 on Blackboard.

Interviews will be conducted by a small panel of tutors who will award the feedback and marks.

## Deliverables

You must submit a CV and cover letter as a *PDF or DOCX* document via *Blackboard*.

You must submit your CV and cover letter as a ZIP file via Blackboard with the following contents:

* A CV, in PDF or DOCX format.
* A Cover Letter, in PDF or DOCX format

## Element 1 Criteria

| **Excellent (85-100%)** | **Very Good (70-85%)** | **Good (55-70%)** | **Satisfactory (40-55%)** |
| --- | --- | --- | --- |
| Cover Letter and CV contain all the required information, are professionally organised and presented, and are suitably tailored towards the job role.  Professional appearance and attitude during the interview, with articulate and detailed responses to questions. | Cover Letter and CV contain most of the required information, are well organised and presented, but would benefit from revisions to better tailor for the job role.  Professional appearance during the interview but with some distracting behaviour. Responses are generally articulate and convincing. | Cover Letter and CV contain much of the required information, are organised and presented, but would benefit from revisions to improve structure and tailor for the job role.  Good appearance during the interview but with some distracting behaviour. Responses are generally articulate. | Cover Letter and CV contain the minimal required information and would benefit from revisions to their organisation and presentation.  Appearance and attitude during the interview are disengaged. Motivation seems lacking and many prompts are needed during questions |

# Element 2: Application

## Scenario

*ThreeAmigos* *Corp* is an event management company that arranges and oversees a range of functions, including conferences, parties and weddings for its clients. As well as logistics, *ThreeAmigos* *Corp* organises the catering and staffing for events.

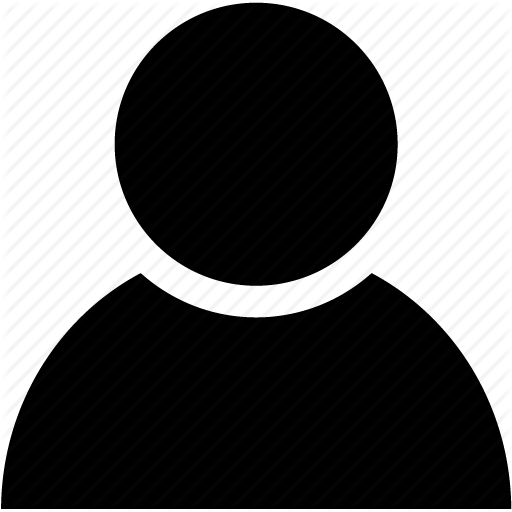
You have been given the responsibility of constructing a new intranet-based prototype system for the events management team. The internal system will be used by *ThreeAmigos* *Corp* staff to perform the key operations:

* Manage event details;
* Assign staff to events;
* Attach guests to events;
* Assign food orders to events.

A system architecture and data model have been agreed upon by all stakeholders. It has been decided that an existing web service will be used for reserving event **venues**.

It has also been decided that **catering** will be created as an independent web service because of the potential for selling on as a third-party service to other companies.

The system architecture and data model are documented below.



**Staff**[Person]

A member of ThAmCo staff working on-site

**ThAmCo**

[Software System]

Uses  
[HTTP/HTML]

Uses  
[HTTP/JSON]

**Database**  
[Container: MSSQL]

Reads / writes  
[T-SQL]

**Database**  
[Container: SQL]

**Database**[Container: SQL]

Reads / writes  
[T-SQL]

Reads / writes  
[T-SQL]

Uses  
[HTTP/JSON]

*ThAmCo.Events*  
**Web Application**  
[Container: ASP.NET Core]

Enables management of customers, staff and events

*ThAmCo.Venues*  
**Web Service**  
[Container: ASP.NET Core]

Query and book venues

*ThAmCo.Catering*  
**Web Service**  
[Container: ASP.NET Core]

Manage food orders

**System Architecture**

1

Event

Food  
Booking

Menu

Staffing

Staff

Guest  
Booking

Guest

Event

Type

Suitability

Venue

Availability

Reservation

1

\*

\*

1

1

\*

\*

1

1

\*

1

0..1

\*

1

1

1

\*

\*

1

\*

**ThAmCo.Events** [Data Model]

**ThAmCo.Venues** [Data Model]

**ThAmCo.Catering** [Data Model]

1

0..1

0..1

Food  
Item

Menu Food Item

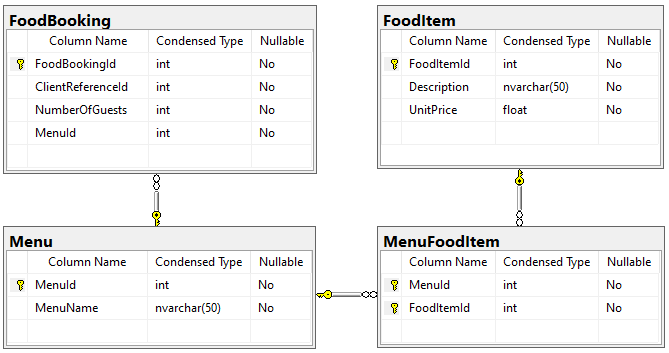
1

\*

1

\*

**Proposed Data Model**



**Entity Relationship Diagram for ThAmCo.Catering**

Note: ClientReferenceId would be supplied by the TheAmCo.Events app (normally the EventId)

## Instructions

You have been provided with a *Visual Studio* solution containing the beginnings of the ASP.NET Core MVC web app, its associated *EntityFramework* Core domain model, and an out-of-the-box ASP.NET API web service. **You are to continue and complete** the development of this web-based intranet system to incorporate the **requirements below**. The given solution also contains the code for the **venues** web service – **do not modify** the *ThAmCo.Venues* project or its code. Do not modify the ASP.NET Core runtime in any project (version 6.0).

**Notes for Development Work:**

* You are expected to **customise the user interface and the application workflow** beyond that provided by the ASP.NET scaffolding.
* You should make the best use of the ASP.NET Core and EntityFramework Core features. **Refactor your code to make appropriate use of the architectural and design patterns** studied in the module.
* You should **document alternative solutions to the key problems and challenges** of the assignment within the source code or a readme file with any references utilised (e.g. stackoverflow.com). An example would be the choice between overloading an existing method or creating a new named method within a class.

**MUST Functional Requirements**

* Web Api Services (ThAmCo.Catering) to:
* Create, edit, delete and list food items - see the ERD above for details;
* Create, edit, delete and list the details of food Menus - see the ERD above for details;
* Add and remove a food item from a menu - see the ERD above for details;
* Book, edit and cancel Food for an Event - see the ERD above for details. The service should return the FoodBookingId as confirmation of the booking;
* Via the MVC web app, the user should be able to:
* Create, list and edit guests;
* Create a new Event, specifying as a minimum its title, date and EventType;
* Edit an Event (except its date and type);
* Book a Guest onto an Event;
* List Guests (including a total count) for an Event and register their attendance;
* Display the details of an individual Guest, including information about the Events with which they are associated and their attendance;

**SHOULD Functional Requirements**

Via the MVC web app, the user should be able to:

* Cancel the booking of a guest from an upcoming Event;
* Reserve an appropriate, available Venue for an Event via the *ThAmCo.Venues* web service, freeing any previously associated Venue;
* Display a list of Events that includes summary information about the Guests and Venue within it;
* Create, list and edit Staff;
* Adjust the staffing of an Event, adding available staff or removing currently assigned staff;
* See appropriate warnings within the event list and staffing views when there is not a first aider assigned to an Event;
* Display the details of a Staff member, including information about **upcoming** Events at which they are assigned to work;
* Cancel (soft delete) an Event, freeing any associated Venue and Staff;

**WOULD Functional Requirements**

Via the MVC web app, the user should be able to:

* Display the details for an Event, which must include details of the Venue, Staff and Guests – this should be more detailed that the summary information found in the Event list;
* Permanently remove personal data by anonymising their Guest entity;
* Display a detailed list of available Venues, filtered by EventType and date range, and then create a new Event by picking a result;
* See appropriate warnings within the event list and staffing views when there is fewer than one member of staff per 10 guests assigned to an Event.

User access control should restrict the following operations:

* Can create and edit staff details (Permitted Users: Managers);
* Adjust the staffing of an event (Permitted Users: Team Leaders or Managers);
* Permanently delete (Permitted Users: Team Leaders or Managers).

**Report**

Prepare a short (500-word) report to evaluate the completeness of the solution, own working practices, and highlight security features built or planned for the app/service.

## Deliverables

You must submit your work as a ZIP file via *Blackboard* with the following structure:

* **Assessment Checklist**
* **Report**
* ***Source***: this directory should contain the entire source code for the system, including any necessary project and solution configuration files and any code documentation. Ensure that you have *cleaned* (i.e. deleted intermediate and non-essential files for) the project(s) by deleting the various *vs*, *bin, and obj* directories. This directory is used for assessing your coding.
* ***Media***: this directory should contain short (20-30 seconds) full-resolution movie captures (MP4 H.264) of the application in use

## Element 2 Criteria

| **Criterion** | **Excellent (85-100%)** | **Very Good (70-85%)** | **Good (55-70%)** | **Satisfactory (40-55%)** |
| --- | --- | --- | --- | --- |
| **Product  (35%)** | Product executes without causing run-time exception or error.  Product meets or exceeds all of the functional requirements.  User interface and application workflow have been appropriately customised. | Product executes with appropriate bespoke pages upon error.  Product meets most functional requirements.  User interface and application workflow have some customisation. | Product executes with infrequent uncaught run-time exceptions.  Product meets all of the MUST and SHOULD functional requirements.  User interface has some customisation but application workflow is mostly scaffolded. | Product executes but has frequent uncaught run-time exceptions.  Product meets all of the MUST functional requirements.  User interface and application workflow is as scaffolded. |
| **Technique  (30%)** | ASP.NET and EntityFramework features are used extensively and successfully.  Services demonstrate correct use of HTTP with suitable end-points.  View Models and Data Transfer Objects are used appropriately.  Code design shows evidence of layering around presentation, business rules and data access.  User security is fully implemented. | ASP.NET and EntityFramework features are used extensively and successfully.  Created services demonstrate successfully, but not RESTful, use of HTTP with various end-points.  View Models and Data Transfer Objects are used appropriately.  Code design shows little evidence of layering.  Security is implemented. | Most of the demonstrated features of ASP.NET and EntityFramework are used successfully.  Given HTTP services are used successfully.  Some additional View Models and Data Transfer Objects are used.  Code design shows little evidence of layering or design pattern usage.  Security not implemented. | ASP.NET and EntityFramework are generally used successfully.  Given HTTP services are used successfully.  Only given View Models and Data Transfer Objects are used.  Code design shows little evidence of layering or design pattern usage.  AJAX calls are not used. |
| **Documentation  (15%)** | Code documentation identifies the key problems and challenges of the assignment and details alternative solutions to them.  Justification is given for choices made with reference to suitable knowledge sources.  Documentation is short because it is clear and concise. | Code documentation identifies the key problems and challenges of the assignment with some detail of alternative solutions.  Justification is given for choices made but without reference to suitable knowledge sources.  Documentation feels padded in places. | Code documentation identifies the key problems and challenges of the assignment.  There is no justification given for choices made.  Documentation frequently feels padded, too short or overly long. | Aspects of the code are documented, but there needs to be further discussion of the key problems or challenges of the assignment.  Justification for the choices made should be clearly stated.  Documentation exists but is incomplete. |

# Module Learning Outcomes

**Personal and Transferable Skills**

PTS1.  Demonstrate key employability skills.

PTS2. Prepare reflective report to consider the development decisions, completeness of the solution and security features.

PTS3. Solve real-world, web-oriented problems using a variety of computer programming languages and technologies.

**Research, Knowledge and Cognitive Skills**

RKC1. Demonstrate knowledge of the multitude of viable solutions to web-oriented programming problems or challenges.

RKC2. Design a secure web app or web service, using appropriate software patterns and principles, to satisfy a requirements specification.

**Professional Skills**

PS1. Construct a secure web app or web service using industry-standard tools and techniques.

PS2. Review own working practice and professional approach to software development.

# Digital and Technology Solutions (Software Engineer)

# Knowledge Skills and Behaviours

**This module supports the delivery and assessment of the following Knowledge Skills and Behaviours (KSBs):**

**Core Skills**

**Information Systems**

C1 Is able to critically analyse a business domain in order to identify the role of information systems, highlight issues and identify opportunities for improvement through evaluating information systems in relation to their intended purpose and effectiveness.

**Systems Development**

C2 Systems Development: analyses business and technical requirements to select and specify appropriate technology solutions. Designs, implements, tests, and debugs software to meet requirements using contemporary methods including agile development. Manages the development and assurance of software artefacts applying secure development practises to ensure system resilience. Configures and deploys solutions to end users.

**Data**

C3 Identifies organisational information requirements and can model data solutions using conceptual data modelling techniques. Is able to implement a database solution using an industry standard database management system (DBMS). Can perform database administration tasks and is cognisant of the key concepts of data quality and data security. Is able to manage data effectively and undertake data analysis.

**Core Technical Knowledge**

C10 Contemporary techniques for design, developing, testing, correcting, deploying and documenting software systems from specifications, using agreed standards and tools.

C16 How to deliver a technology solutions project accurately consistent with business needs.

**Core Behavioural Skills**

C18 Fluent in written communications and able to articulate complex issues.

C19 Makes concise, engaging and well-structured verbal presentations, arguments and explanations.

C23 Able to give and receive feedback constructively and incorporate it into his/her own development and life-long learning.

C24 Applies analytical and critical thinking skills to Technology Solutions development and to systematically analyse and apply structured problem solving techniques to complex systems and situations.

C25 Able to put forward, demonstrate value and gain commitment to a moderately complex technology-oriented solution, demonstrating understanding of business need, using open questions and summarising skills and basic negotiating skills.

C26 Able to conduct effective research, using literature and other media, into IT and business related topics.

C29 Ability to perform under pressure

C30 A thorough approach to work

C31 Logical thinking and creative approach to problem solving

**SOFTWARE ENGINEERING SPECIALISM**

Skills

SE1 Create effective and secure software solutions using contemporary software development languages to deliver the full range of functional and non-functional requirements using relevant development methodologies.

SE2 Undertake analysis and design to create artefacts, such as use cases to produce robust software designs.

SE3 Produce high quality code with sound syntax in at least one language following best practices and standards.

SE4 Perform code reviews, debugging and refactoring to improve code quality and efficiency.

SE5 Test code to ensure that the functional and non-functional requirements have been met.

SE6 Deliver software solutions using industry standard build processes, and tools for configuration management, version control and software build, release and deployment into enterprise environments.

**Technical Knowledge (knows and understands)**

SE9 How to apply software analysis and design approaches.

SE10 How to interpret and implement a design, compliant with functional, non-functional and security requirements.

SE11 How to perform **functional** and unit testing.

SE12 How to use and apply the range of software tools used in software engineering.