ASSESSMENT AND INTERNAL VERIFICATION FRONT SHEET (Individual Criteria)

(Note: This version is to be used for an assignment brief issued to students via Classter)

Course Title	**B.Sc. (Hons.) in Software Development**		**Lecturer Name & Surname**	**Ryan Attard**	
Unit Number & Title	ITSFT-506-2011 Enterprise Programming				
Assignment Number, Title / Type	1, Building An Enterprise Application using a clean architecture / Home				
Date Set	28/2/2025	**Deadline Date**	23/03/2025		
Student Name	Jacob Chircop	**ID Number**	0186005L	**Class / Group**	SWD6.2B

Assessment Criteria	Maximum Mark
KU1.1: Describe what is meant by enterprise software architecture and the role that it offers in practice	5
KU1.2: Interpret what is meant by refactoring	5
KU1.4: Clarify and relate the process models used	5
KU2.1: Show correct use of software design pattern	5
KU3.1: Describe the management standpoint in implementing enterprise solutions	5
KU3.4: Find and select a perspective of versioning, configuration and scalability of the solution	5
KU4.1: Describe what is meant by cloud services and delivery models	5
AA2.2: Demonstrate software design patterns in specific problems found in enterprise applications	7
AA2.3: Select and implement appropriate design patterns to a solution being implemented	7
AA3.2: Choose and develop the correct delivery model	7
AA4.2: Illustrate what methods can be used to upload content onto cloud services	7
AA4.3: Use and draw practical application to upload application content to cloud services	7
SE1.3: Construct and ascertain that enterprise standards fit within an enterprise solution	10
SE3.3: Identify and design basic security in data integrity features	10
SE4.4: Revise and evaluate the content application and its appropriate use on cloud services	10
Total Mark	100

Notes to Students:

- This assignment brief has been approved and released by the Internal Verifier through Classter.
- Assessment marks and feedback by the lecturer will be available online via Classter (Http://mcast.classter.com) following release by the Internal Verifier
- Students submitting their assignment on Moodle/Turnitin will be requested to confirm online the following statements:

Student's declaration prior to handing-in of assignment

- ❖ I certify that the work submitted for this assignment is my own and that I have read and understood the respective Plagiarism Policy

Student's declaration on assessment special arrangements

- ❖ I certify that adequate support was given to me during the assignment through the Institute and/or the Inclusive Education Unit.
- ❖ I declare that I refused the special support offered by the Institute.

Assignment Guidelines

Read the following instructions carefully before you start the assignment. If you do not understand any of them, ask your invigilator.

- This assignment is a HOME-based assignment but the following criteria will be assessed online in a time-constrained multiple choice questions:
    - KU4.1
    - KU1.4
    - SE4.4

- The time-constrained part consists of 15 questions, randomly picked, at it will be 20 minutes long.
    - Link: https://vle.mcast.edu.mt/mod/quiz/view.php?id=61028
    - Link for IEU students (with extra time): https://vle.mcast.edu.mt/mod/quiz/view.php?id=61029

- The Home based part must be completed in 3 days

- Copying and Pasting code from any source is **Strictly Prohibited** and will be penalised according to disciplinary procedures – code has to be yours. There won't be any interview so code and answers have to be clear and working as instructed. Any signs that show that the work has been copied or AI-generated and no effort was put to understand what's happening will result in no acceptance of the work submitted.

- Deadline: See front page

- This assignment has a total of 100 marks: 80 marks assessed as a home assignment and 20 marks in a time-constrained assignment. Follow the rubric at the end of each question for detailed assessment of marks.

- Submission of home part must be done by inputting your Git repository link on VLE. Check that your link is open for public access.

TCA Part (20 marks):

See assignment guidelines for links and instructions***.***

Home Assignment: Building a simple Online Poll Website

1. (KU3.4) *–Find and select a perspective of versioning, configuration and scalability of the solution [5]*

Make use of a git repository.
Delivery:

- Link with public access and code committed & pushed in it already [5]. In the submission link provided there will be space where you can post link to your git (which should be public) –

Rubric: failure to access the git will result in loss of marks i.e. check your accessibility before submitting in incognito mode to avoid misunderstandings.

2. (KU1.1) *– Describe what is meant by enterprise software architecture and the role that it offers in practice [5]*

Create a .NET 6 or .NET 8 Web Application which consists of the below listed projects. Name it using **your** name (e.g. RyanAttardEPSolution):

a. **Domain** contains all the models you will be using in your application
b. **DataAccess** contains all the repository and context classes you will make use of to interact directly with your database
c. **Presentation** is the web application with the respective Controllers and Views and ViewModels

Delivery: You will need a number of classes to make this function properly however marks will be given on the correct placement of the following classes:

- PollDbContext.cs
- Poll.cs
- PollRepository.cs

Rubric:- Every incorrect placement cost you 2 marks;

3. (KU1.2) - *Interpret what is meant by refactoring [5]*

The PollRepository.cs class must contain a method (CreatePoll) which creates a **Poll** having a title, Option1Text, Option2Text, Option3Text, Option1VotesCount, Option2VotesCount, Option3VotesCount and DateCreated. It should save the contents successfully in the database.

Rubric: Any missing details from the above will result in loss of all marks;

4. (AA2.2) Demonstrate software design patterns in specific problems found in enterprise applications

Make use of Constructor Injection and Method Injection in PollRepository and PollController (which you have to create) in order to implement a feature whereby a user can input the details outlined in KU1.2 to create a new poll and saved in the database. Feature has to work using two different injection methods. You have the flexibility to decide where to use each injection method, but both must be implemented to meet the criterion.

Rubric: Any missing or improperly used DI method will result in loss of 3.5 marks;

5. (AA3.2) - *Choose and develop the correct delivery model [7]*

In the PollRepository.cs class implement a method called GetPolls() in such a way that it returns all the polls in the database. Use a proper **efficient** return type. [2]
In the PollController, develop the necessary code to show all the polls on a view each of which can be selected to vote in. [3]
Polls must be displayed sorted by date, most recent one first. Sorting must be done in the controller keeping an eye on the efficiency of the call. [2]

Rubric: Marks are given if code is according to requirements. Failure to adhere to any of the above requirements will result in the loss of all the allocated marks.

6. (KU2.1) – *Show correct use of a software design pattern [5]*

In the PollController, develop a feature which shows the poll details in a clear way for the end-user to vote on (note: the marks here do not assess the voting but the presentation of the poll). Use the GetPolls(…) implemented in (AA3.2) to filter the poll selected (in the controller) by id and pass the details to the View.

Rubric: Failure to adhere to the requirements will result in loss of all marks.

7. (AA4.2) Illustrate what methods can be used to upload content onto cloud services [7]

In the PollRepository, add a method called Vote(…). Call this method from the HTTPPOST Vote method inside the PollController. Also find a way how to pass the poll and option which was voted upon to these respective methods and save it in the database in a way which fits what was described in KU1.2.

Rubric: Any missing or improper method (there should be two methods) will result in loss of 3.5 marks.

8. (AA2.3) – *Select and implement appropriate design patterns to a solution being implemented [7]*

The PollRepository class contains 2 methods among others which are CreatePoll(…) and GetPolls(…). Implement these in a different class called PollFileRepository.cs in such a way that they add a poll to a json file and read polls from a json file respectively.

Rubric: Each correct method carries 3.5 marks.

9. (KU3.1) - Describe the management standpoint in implementing enterprise solutions

Make the necessary changes to your web application such that your solution at any point in time can easily adapt to using the PollFileRepository.cs methods (described in AA2.3) instead of the PollRepository.cs, with minimal changes.

Rubric: There are 3 changes that must be implemented. One cannot exist without the other, so failure to apply all the changes will result in the loss of all the marks.

10. (SE1.3) - *Construct and ascertain that enterprise standards fit within an enterprise solution [10]*

Poll results can be viewed by all users, even anonymous ones. Use a 3rd party component such as Chart.js which facilitates the graphical visualisation of the results in a chart (e.g. bar chart) when a poll is selected. Implement whatever necessary to achieve this.
Alternatively (BUT you get half the marks) you can display just the figures of the voting count per option of the Poll.

Rubric: If a graphical display showing the voting count per Poll is achieved you get 10 marks. If a textual display of the votes is achieved you get 5 marks. If in any case feature is left out or voting count is displayed inaccurately no marks are given.

11. (SE3.3) - *Identify and design basic security in data integrity features [10]*

Identify, design and implement a way to ensure that a user can only vote in a poll **once**. To answer this question, you must make use of an ActionFilter and a Login feature.

Rubric: Login – 5 marks. ActionFilter working properly – 5 marks.

12. (AA4.3) - *Use and draw practical application to upload application content to cloud services*

Host the application on any hosting provider of your choice. The link to access it must be provided.

In this assignment there are these features:

- Login
- Creation of a poll
- Viewing of a poll including chart/ textual voting count
- Voting (only once)

Rubric: The above features must work on the hosted website. Each non-working feature on the hosted website will result in loss of 2.5 marks per feature.