**CS6004ES –Application Development Group Coursework - 2  
(2023/24)**

CS6004ES – Application Development Group Coursework – 2 (2023/24)

This assessment is a group coursework to be completed by a team consisting of **a maximum of 4 students**. The objective is to develop and document a web application using **ASP.NET MVC with the C# programming language in Visual Studio**. The application development methodology will be based on Object-Oriented Design principles and coding practices. It is essential to adhere to the following guidelines:

* Encapsulation Techniques: Implement encapsulation to ensure that data and methods are appropriately encapsulated within classes, providing access only through well-defined interfaces.
* Abstraction: Utilize abstraction to create abstract classes, interfaces, and methods that represent general concepts and hide implementation details, promoting modularity and flexibility.
* Inheritance: Apply inheritance to establish a hierarchy of classes, enabling code reuse and specialization. Utilize base classes and derived classes to model relationships between entities in the system.
* High Cohesion and Loose Coupling: Aim for high cohesion by ensuring that each class has a clear and well-defined responsibility. Encourage loose coupling between classes by reducing dependencies and utilizing interfaces or dependency injection.

Compliance with the above criteria is crucial as marks will only be awarded if these guidelines are followed.

Your software artifact should be submitted as a Visual Studio project within a Visual Studio Solution. The project will be evaluated using Visual Studio. The coursework carries 30% of the module mark.

Submission Deadlines:

Project Submission: 28th December 2023

In-class Demo: 2nd of January 2024

Groups that fail to meet the deadline must submit their work to the Undergraduate Registry along with a completed *mitigating circumstances form.*

**Case Study: Raythos Aerospace - Imaginary Aircraft Manufacturing Company**

**Raythos Aerospace** is an established aircraft manufacturing company looking to develop an online system to streamline their manufacturing and delivery processes. The system should enable customers to browse and order aircraft models, while providing comprehensive management features for the company. The requirements for the system are as follows:

1. User Registration and Login: Implement a user registration and login system, allowing customers to create accounts and securely log into the system.
2. Aircraft Catalog: Display a catalog of available aircraft models, providing detailed information about each model, including specifications, images, and pricing.
3. Customization Options: Allow customers to customize their aircraft by selecting various options such as seating configuration, interior design, and additional features.
4. Ordering and Payment: Provide a seamless ordering process, including the selection of desired aircraft models, customization options, and secure online payment.
5. Manufacturing Tracking: Develop a system to track the progress of aircraft manufacturing, from initial order placement to final delivery. Include features for status updates, estimated delivery dates, and real-time notifications.
6. Inventory Management: Implement an inventory management system to track the availability of aircraft components and ensure timely procurement to fulfill customer orders.
7. Shipping and Delivery: Integrate shipping and logistics functionalities to handle the shipping and delivery of completed aircraft to customers' designated locations.
8. Reporting and Analytics: Incorporate reporting capabilities to generate performance reports, track sales, and analyze customer preferences, aiding in decision-making and process improvement.
9. Admin Dashboard: Create a comprehensive administration dashboard that allows authorized personnel to manage user accounts, monitor orders, update inventory, and access various system reports.
10. Security and Data Protection: Implement robust security measures to safeguard customer data, including encryption, secure authentication, and protection against common security threats.

**Deliverables:**

1. Use Case Diagram: Illustrate the interactions between system actors and functionalities using a use case diagram.
2. Class Diagrams: Design and document the class structure of the system, depicting the relationships and interactions between classes.
3. Entity-Relationship Diagram (ERD): Create an ERD to model the database schema and relationships between entities.
4. System Architecture: Define the architectural design of the web application, including the choice of frameworks, technologies, and deployment strategy.
5. User Interface Design: Design intuitive and user-friendly interfaces for customer browsing, ordering, and account management.
6. Implementation: Develop the web application using ASP.NET MVC with C# programming language, following the principles of object-oriented design and coding practices.
7. Testing: Conduct comprehensive testing to ensure the system functions as intended, including unit testing, integration testing, and user acceptance testing.
8. Documentation: Provide detailed documentation, including installation instructions, user manual, and code documentation.
9. Presentation: Prepare a presentation to demonstrate the features and functionalities of the developed web application.
10. Submission: Submit the Visual Studio project, solution, documentation, and any additional artifacts required for evaluation.

Remember to adhere to the given guidelines for encapsulation, abstraction, inheritance, high cohesion, and loose coupling in the design and implementation of the web application. Compliance with these criteria will contribute to the overall evaluation and assessment of your coursework.

**Marking Scheme for the CS6004ES Group Coursework**

This group coursework counts for 30% of the module mark. The following are guidelines for marking. Mark each item listed below on a scale 0 to 5 where the marks correspond. Then multiply the mark by the weighting indicated, total and divide by 2 to get the total mark.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mark** | | | **Characterized by** | | | | | | |
| 0 | | | No work or work totally irrelevant | | | | | | |
| 1 | | | Work started on right lines but no result | | | | | | |
| 2 | | | Some result, with major lack and/or errors | | | | | | |
| 3 | | | Acceptable result but incomplete, or some good result with minor errors | | | | | | |
| 4 | | | Good result but can be further polished | | | | | | |
| 5 | | | Excellent result | | | | | | |
|  | | **Item** | | **Weight** | ***Mark***  ***(0 - 5)*** | | ***Weight x Mark*** | |
| **Software Implementation** | | | |  |  | |  | |
| 1 | | The application user interface and menu | | 2 | 5 | | 10.00 | |
| 2 | | Task1 : Admin Account creation and login system. | | 2 | 5 | | 10.00 | |
| 3 | | Task2: Admin can manage Aircraft details and participating team details | | 2 | 5 | | 10.00 | |
| 4 | | Task3: Customers can register with their personal data. | | 4 | 5 | | 20.00 | |
| 5 | | Task4: Aircraft owners can view the Aircraft’s profile. | | 5 | 5 | | 25.00 | |
| 6 | | Task5 : Admin can Set Base price and Maximum price details. | | 3 | 5 | | 15.00 | |
| 7 | | Task6: Admin Can manage Aircraft Sales. | | 3 | 5 | | 15.00 | |
| 8 | | Task7: Sold/Unsold Aircrafts' details should be visible for everyone. | | 3 | 5 | | 15.00 | |
| 9 | | Task 8: Every Customer should be able to view the Status of their Aircraft they are interested in buying. | | 3 | 5 | | 15.00 | |
| 10 | | Task9 : Admin should be able to generate various reports. | | 3 | 5 | | 15.00 | |
| **Reflective Essay** | | | |  | |  | |  |
| A | Installation Guide and Configuration and Manual | | | 1 | | 5 | | 5.00 |
| B | Concise description of your logical solution to each of the implemented function of the application. | | | 1 | | 5 | | 5.00 |
| C | The software architecture | | | 1 | | 5 | | 5.00 |
| D | Detailed description of the classes’ properties and methods and the class diagram | | | 1 | | 5 | | 5.00 |
| E | Individual member own reflection of own experience | | | 1 | | 5 | | 5.00 |
| **Programming style** | | | |  | |  | |  |
| 1 | Clarity of code which shows the underlying algorithm | | | 1 | | 5 | | 5.00 |
| 2 | Sensible naming of programmer-defined variables, classes, properties, and methods | | | 1 | | 5 | | 5.00 |
| 3 | Useful comments in code | | | 1 | | 5 | | 5.00 |
| 4 | Data validation and exception handling | | | 1 | | 5 | | 5.00 |
| 5 | Interface design and usability of the system | | | 1 | | 5 | | 5.00 |
|  | **Total** | | |  | |  | | **200.00** |