Case study**:  Retail – Order Management**

* Created / Update Order (Just have a form to add an order) and add one or more products under an order
* Create order functionality should accept gift message as a optional attachment (System should accept Jpg or Word)
* View Order & Products
* Cancel Order
* Notification (optional): Asynchronous email notification on successful order creation / cancellation

Expectations and Deliverables:

* PowerPoint Presentation.
* Class Diagrams
* Some kind of Kanban board.
* Project must be on GitHub.
* REST API & ORM (Hibernate with JPA for Java / Entity Framework for MS) should be used to implement business logic.
* Angular / React latest version should be used to build user interfaces.

| **Topics / Areas** | **Deliverables** |
| --- | --- |
| REST API with CRUD operation | **API:**  Proper use of HTTP methods, Parameters, Headers & response codes JSON should be used for request and response with proper input validation.  Basic Authentication  Proper exception handling  Average /Maximum number of requests to be processed: 10/50 request per sec. 98 % real-time operation should be completed within min 1 sec to max 5 sec  **Data Base:**  RDBMS design and DDL script  ORM implementation with Transaction management |
| Single page responsive - User Interface | Responsive UI integrated with APIs (Angular or React)  Common UI best practices (forms, CSS, validation, proper content including success / failure messages, spacing… etc.) |
| Asynchronous processing (Optional) | Fire and forget implementation using naked thread implementation or messaging queues   Date loss prevention in case of exceptional scenario (Reliable messaging) |
| Unit testing | Unit testing (Automation using Framework i.e., Junit, NUnit.. etc., Test case depth i.e. coverage, addressing positive, negative scenarios… etc.) |

PS:

* Have a static login page with hardcoded role/s (should be based on json configuration)