# CAPSTONE PROJECT 3 PART 1

**Case Study 1 (Q1-Q6  24 Marks)**

**Q1. Draw a Use Case Diagram - 4 Marks
A customer can make a payment either by Card or by Wallet or by Cash or by Net banking.
 **

 **Q2.** **Derive Boundary Classes, Controller classes , Entity Classes. - 4 Marks**

**Boundary Classes**: Boundary Class used to handle interactions between the system and external actors

**Ex:** Payment option Boundary

Card Payment Boundary

Wallet Payment Boundary

Cash Payment Boundary.

NetBanking Payment Boundary


**Controller classes**: Controller Class-act as intermediaries between boundary and entity classes.

**Ex:** Payment initiated Controller

Card Payment Controller

Wallet Payment Controller

Cash Payment Controller

NetBanking Payment Controller



**Entity Classes**: Entity Class-represent the core data and business logic of the application.

**Ex:** Customer

Payment

Card

Wallet

Server

 

**Q3. Place these classes on a three tier Architecture. - 4 Marks**
**1. Presentation Layer (UI Layer): Boundary Classes**:

Payment Option Boundary

Card Payment Boundary

Wallet Payment Boundary

Cash Payment Boundary

NetBanking Payment Boundary

**2. Business Logic Layer (Application Layer):Controller Classes:**

Payment Initiated Controller

Card Payment Controller

Wallet Payment Controller

Cash Payment Controller

NetBanking Payment Controller

**3. Data Layer (Persistence Layer):Entity Classes:**

Customer

Payment

Card

Wallet

Server

 **Q4. Explain Domain Model for Customer making payment through Net Banking - 4 Marks
A Domain Model is a conceptual representation that defines the structure, relationships, and behaviours of entities within a specific problem domain
**

**Q6. Explain Conceptual Model for this Case - 4 Marks**

* A conceptual model is a high-level representation of a system that helps in understanding, visualizing, and communicating the essential aspects of a domain.
* Its Provides a clear and simplified view of the domain, making it easier to understand.
* Key Elements of a Conceptual Model:
* Entities Customer, Product, Order & Payment
* Attributes customerld, name, email, phone Number.
* Relationships For example, a Customer places an Order.

**A conceptual model for the payment process done by a customer using net banking provides a high-level understanding of the key concepts and their relationships involved in the payment transaction.**

**It helps in visualizing the overall structure and flow of the payment process.**

 **Q7. What is MVC architecture? Explain MVC rules to derive classes from use case diagram and guidelines to place classes in 3-tier architecture - 8 Marks**

**The Model-View-Controller (MVC) framework is an architectural pattern that separates an application into three main logical components Model, View, and Controller.**

* **Model Represents the data and the business logic of the application.**
* **View Represents the presentation layer of the application.**
* **Controller-Acts as an intermediary between Model and View**

**MVC Architecture Rules**

1. Combination of One Actor and an use case results in one Boundary class

2. Combination of Two Actors and an use case results in two Boundary classes

3. Combination of Three Actors and an use case results in Three Boundary classes and so on....

Note: only one primary actor is to be considered with a use case.

4. Use case will result in a controller class

5. Each Actor will result in one entity class

 **Q8. Explain BA contributions in project (Waterfall Model – all Stages) – 8 Marks
**

 **Q9. What is conflict management? Explain using Thomas – Kilmann technique – 6 Marks
Conflict management is the process of resolving conflicts or disagreements between individuals or groups in a constructive manner**

**Thomas Kilmann technique is a widely used tool tor assessing conflict resolution styles & guiding individuals in selecting appropriate strategies to manage conflicts.
5 steps of conflict management -**

* **Identify the conflict.**
* **Discuss the details.**
* **Agree with the root problem.**
* **Musck for every possible solution for the conflict.**
* **Negotiate the solution to avoid future conflicts**

 **Q10. List down the reasons for project failure – 6 Marks**Poor Planning, Unclear Objectives and Requirements, Inadequate Risk Management, Poor Communication, Scope Creep, Lack of executive support , Lack of Stakeholder Engagement, Resource Constraints, Technical Challenges, Unrealistic expectation

 **Q11. List the Challenges faced in projects for BA – 6 Marks**

Unclear or Changing Requirements, Managing Stakeholder Expectations, Scope Creep and Scope Management, Time and Resource Constraints, Quality Assurance and Testing, Documentation and Knowledge Management, Technology Constraints and Complexity, Resistance to change

 **Q12. Write about Document Naming Standards – 4 Mark**

**A document numbering standard is a systematic approach to assigning unique identifiers to vanous documents created and used throughout the development process.**

**Ex.** Suppose we have a project with the ID "PROJ123," and we're working with a Requirements Specification Document

Project ID: PROJ123

Document Type: REQ

Version: 1.0

Date: 2024-05-26

The document identifier could be: PROJ123-REQ-10-2024-05-26

 **Q13. What are the Do’s and Don’ts of a Business analyst – 6 Marks**

| **Sr. No.** | **DO'S (As a Business Analyst)** | **DON'TS (As a Business Analyst)** |
| --- | --- | --- |
| **1** | Consult an SME for clarifications in requirements. | Never say NO to the client. |
| **2** | Go to the client with a plain mind with no assumptions. | There is NO word called as "BY DEFAULT." |
| **3** | Listen carefully and completely until the client is done, then ask queries. | Never imagine anything in terms of GUI. |
| **4** | Try to extract maximum leads to the solution from the client himself. | Don't interrupt the client when he is giving you the problem. |
| **5** | Concentrate on the important and truly required requirements. | Never try to give solutions to the client straight away with your previous experience and assumptions. |
| **6** | Don't be washed away by add-on functionalities; focus on core needs. | Don't imagine solutions on a screen basis. |
| **7** | Question the existence of existence. / Question everything. | — |
| **8** | Every problem of a client is unique. No two problems are the same, as factors like approach, technology, local laws, and place of use may differ. | — |

**Q14. Write the difference between packages and sub-systems – 4 Marks**

| **Aspect** | **Packages** | **Sub-systems** |
| --- | --- | --- |
| **Definition** | Collection of components that are **not reusable**. | Collection of components that are **reusable**. |
| **Example** | Used in **application development** companies. | Used in **product development** companies. |
| **Granularity** | Smaller and more focused in scope. | Larger and encompass multiple packages or modules. |
| **Dependency Management** | Manages dependencies at a **class/component** level. | Manages dependencies at a **higher level**, defining boundaries and interfaces between different parts of the system. |

**Q15. What is camel-casing and explain where it will be used- 6 Marks**

* Camel casing is a naming convention used in computer programming and is characterized by removing spaces between words and capitalizing the first letter of each word except for the first word.
* By using camel casing, developers can create meaningful and readable names that are easier to understand and follow coding standards. It promotes consistency within the codebase and improves collaboration among team members.
* Its used for naming variables, functions, and identifiers
* EX. CamelCase: camelCaseExample

**Q16. llustrate Development server and what are the accesses does business analyst has? -6**

* A development server refers to a dedicated environment or server that is used during the software development process
* It provides a platform for developers and testers to build, test and debug applications before they are deployed to a production environment.
* As a BA, we have only limited access only.
* The development server typically replicates the target production environment to ensure compatibility and accurate testing.
* Read-Only Access
* Collaborative Access
* Limited Configuration Access

**Q17. What is Data Mapping 2 Mark**

* Data mapping is the process of establishing a relationship or connection between data elements in two or more different data sources or data formats.
* It involves defining how data from one source corresponds to or transforms into data in another source. Data mapping is commonly used in data integration, data migration, and data transformation processes.
* The purpose of data mapping is to ensure that data can be accurately and effectively transferred, converted, or transformed between different systems, databases, or formats.
* It involves identifying the source data elements, determining their meaning and a structure, and mapping them to the corresponding target data elements.

**Q.18 What is API. Explain how you would use API integration in the case of your application Date format is dd-mm-yyyy and it is accepting some data from Other Application from US whose Date Format is mm-dd-yyyy 10 Marks**

* An API (Application Programming Interface) is a set of rules and protocols that allows different software applications to communicate and interact with each other.
* Establish API Communication: Set up API communication between your application and the other application to exchange data
* Data Formatting: When sending date data from your application to the other application, convert the date from the dd-mm-yyyy format to the mm-dd-yyyy format. This can be achieved by extracting the day, month, and year components from the date and rearranging them according to the target format.
* Data Parsing: When receiving date data from the other application, parse the mm-dd-yyyy formatted date into your application's dd-mm-yyyy format. Again, you will need to extract the day, month, and year components and rearrange them accordingly.
* Data Validation: Perform data validation and ensure that the converted date remains valid after the format conversion. Check for edge cases, such as invalid dates or date ranges that might be affected by the format conversion, and handle them appropriately.