**Question-1**



**Question 2- Derive Boundary Classes, Controller classes, Entity Classes.**

A **Boundary Class** represents the interaction between the system and external entities such as users or other systems. It is responsible for handling user inputs, displaying outputs, and managing communication between the UI and the internal logic.

**Characteristics:**

* Acts as an interface between the system and users.
* Handles UI components, web pages, or API endpoints.
* Does not contain business logic.

A **Controller Class** manages the application's business logic and coordinates between **Boundary Classes** and **Entity Classes**. It processes requests, enforces business rules, and interacts with the data layer to perform required operations.

**Characteristics:**

* Controls data flow between the UI and the database.
* Implements business logic and validation rules.
* Calls the appropriate **Entity Classes** for data operations

An **Entity Class** represents the core business objects and their persistence in the system. It encapsulates data and typically maps to a database table or an in-memory object model.

**Characteristics:**

* Represents real-world entities (Customer, Payment, and Order).
* Stores data in attributes and provides methods for accessing or modifying it.
* Used for database operations such as CRUD (Create, Read, Update, Delete).

**Question 3-Place these classes on a three tier Architecture.**

The 3- tier architecture divides the application into 3 layers

* Application layer-payment U interface ( view- Boundary class)
* Business logic layer- Payment process- ( controller class)
* Database layer- Payment services that connects with the bank database- (model class)

**Question4-** Explain Domain Model for Customer making payment through Net Banking

A **Domain Model** represents real-world objects and their relationships in a system. It is a **high-level conceptual model** that helps understand business entities and their interactions without focusing on implementation details.

****

**Question 5- Draw a sequence diagram for payment done by Customer Net Banking**

****

**Question 6-Explain Conceptual Model for this Case**

A **Conceptual Model** represents **high-level entities** and their **relationships**, focusing on **what the system does** rather than technical implementation.

## ****Key Entities in Conceptual Model****

For a **Customer making a payment via Net Banking**, the main entities are:
**Customer** → Initiates payment.
**Payment** → Stores payment details (amount, status, method).
N**et Banking** → Processes the transaction via the bank.
**Bank** → Validates & processes the payment.

**Question 7-What is MVC architecture? Explain MVC rules to derive classes from use case diagram and guidelines to place classes in 3-tier architecture**

**MVC (Model-View-Controller)** is a **design pattern** used in software development to **separate concerns** in an application. It divides the application into three interconnected components:

**Model** → Handles **data and business logic**
**View** → Manages **UI and presentation**
**Controller** → Manages **user input and communication between Model & View**

**Model-** The model class knows about all the data that needs to be displayed. It is model who is aware about all the operation that can be applied to transform that class. It only represents the data of an application. The model represents enterprise data and the business rule that govern access to and update this data. This represent database. All model class classes are represent as entity class.

**View-** The view represents the presentation of the application. The view class refers to the model to obtain and renders it. The view is not dependent on the application logic. View class is the data required by the query. View class representing as boundary class.

**Controller-** Whenever the user sends a request for something then it always go through the controller. The controller is responsible for intercepting the requests from view and passes it to the model for the appropriate action. Controller class is working on user’s command, understand command request given user through boundary class.

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

|  |  |
| --- | --- |
| Stages  | Activities |
| Planning | Enterprise analysis, Gap analysis, project budget, scope of project, project planning  |
| Pre project | Project planning with the involvement of project manager |
| Requirement gathering | Involvement of BA’s in gathering data and info from stakeholders by conducting session like brainstorming, interviews, workshops and to create BRD (Business requirement document) |
| Requirement analysis | Creation of FRS AND SRS,  |
| Design | Providing Use case diagrams, wireframes, mockups to assist the UI/UX design |
| Development | Acts as bridge between the stakeholders and the developers, handle change requirement |
| Testing | Assist the QA team in creating test case scenarios |
| UAT | Performs UAT with stakeholders, assist them and mark the changes and feedbacks from stakeholders for further improvement or change, ensuring the defects are resolved and requirements are met. |
| Implementation | Support the go live process by ensuring all requirements are addressed, assist in preparing training materials, sops and manuals  |
| Maintenance | Gather feedback from user post production issues; assist in identifying enhancement and future improvements. |

**Q9**. **What is conflict management? Explain using Thomas – Kilmann technique**

Conflict Management is the process of **identifying, addressing, and resolving disagreements** in a constructive way. In business analysis, conflicts can arise between stakeholders, teams, or departments due to differing priorities, expectations, or constraints.

#### Thomas-Kilmann Conflict Management Model

The **Thomas-Kilmann Conflict Mode Instrument (TKI)** identifies five strategies to handle conflicts based on two factors:

X-axis- co-operation

Y-axis- assertiveness.

5 option of conflict management

* Competing
* Avoiding
* Accommodating
* Collaborating
* Comprising

5steps to conflict management

* Identify the conflict
* Discuss the details
* Agree with the root problem



**Question 10 List down the reasons for project failure**

* **Answer-** Improper requirement gathering by BA
* Continuous change in requirements by clients
* Lack of user involvement
* Lack of executive support
* Unrealistic expectation
* Improper planning by PM

**Question 11-List the Challenges faced in projects for BA**

* Lack of training
* Obtaining sign-off on requirement
* Change management
* Coordination between developers and testers
* Conducting meeting
* Making sure status reporting is effective
* Driving clients for UAT completion
* People management
* Overall making sure project health is in good shape and delivered as per the time-lies without any issues.

**Question 12- Write about document naming standard**

A document numbering standard is a systematic approach to assigning unique identities to various document created and used throughout the development.

Suppose we have a project with the ID PROJ123 and we are working with requirement document specification

PROJECT ID- PROJ123

Document type- REQ

Version- 1.0

Date-24.03.2025

The document identifier will be PROJ123-REQ-1.0-2024-03-24

**Question 13. What are the Do’s and Don’ts of a Business Analyst?**

* Never say NO to client
* There is NO word called as “ BY DEFAULT”
* Never imagine anything in terms of GUI
* Question the existence of existence/ question everything

**Question 14. Write the difference between packages and sub-systems**

A **package** is a logical grouping of related classes, interfaces, or components within a software system. It is mainly used to organize code and improve modularity. Packages do not have independent functionality; instead, they serve as namespaces to manage related elements efficiently.

A **system**, on the other hand, is a complete, self-contained application or a set of interconnected sub-systems that perform specific business functions. A system consists of multiple components, including packages, sub-systems, databases, and external integrations, working together to achieve a goal.

In summary, a package is just an organizational unit within a system, whereas a system is a fully functional software solution that may contain multiple packages and sub-systems.

**15. What is camel-casing and explain where it will be used**

Camel casing means Initial word will be small alphabets and from second word onwards beginning alphabet is capital and the rest are small.

It is used in message sending. Object work together in a system. They do this by sending messages to one another. One object sends another a message- a request to perform an operation and the receiving object performs that operation. Messages are sent by methods. Methods name are represented by camel casing.

**Q16. Illustrate Development server and what are the accesses does business analyst has?**

A development sever is a testing environment where developers test and writes code, tester performs functional and integration testing, business analyst requirements and UI flow.

Typical server environment of a project

|  |  |
| --- | --- |
| Environment  | Purpose |
| Development server | Used for coding, unit testing, and requirement validation |
| Testing(QA) server | Used for system testing, integration testing and bug testing |
| UAT( user acceptance testing) server | Used for final validation by BA |
| Production server | Live environment used by actual users. |

A business analyst does not have coding right but they do have access to

* Application UI( Testing features)
* Database ( read access)
* APIs
* Logs and report
* Test cases and bug tracking

**QUESTION 17- What is Data Mapping**

**Data Mapping** is the process of linking data fields from one system or format to another. It ensures that **data is correctly transferred, transformed, and interpreted** between different systems, databases, or applications.

**Data Mapping ensures accurate data transfer between systems.
Used in APIs, migrations, ETL, and format conversions.
Avoids data inconsistency and errors.**

**Question 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**

Application programming interface is a set of rule that allows different software application to communicate with each other. Essentially an API acts as an intermediary that lets two application interact, enabling them to share data and functionality without needing to understand each other’s internal workings.

* **Step 1: API Call from the US Application**
* The external system sends data in **MM-DD-YYYY** format via API.
* Example API request (JSON format)
* **Step 2: API Receives Data & Processes It**
* Your application’s API receives the request.
* A function **converts the date** from MM-DD-YYYY to DD-MM-YYYY.
* **Step 3: Convert Date Format in Backend**
* Example Python function for date conversion:

**Key Takeaways:**

**APIs enable seamless data exchange between systems.
Date conversion is crucial for integrating systems with different formats.
Backend logic (Python, Java, etc.) can handle format changes before storing data.**