A customer can make a payment either by Card or by Wallet or by Cash or by Net banking.

 Q1. Draw a Use Case Diagram

Answer -

A **Use Case Diagram** represents the interactions between the **Customer** and the **Payment System** for different payment methods.

 

Q2. Derive Boundary Classes, Controller classes, Entity Classes.

Answer -

1. Boundary (UI) Classes:

Used to handle the interaction between the system and the external user.Boundary classes define the interface between users and the system.

Examples:

* Payment UI - Interface for selecting the payment method.
* Transaction Confirmation UI - Displays payment success/failure messages.

2. Controller (Business Logic) Classes:

Act as intermediator between boundary and entity classes. Controller classes manage business logic and process requests.

Examples:

* PaymentController - Handles payment processing logic.
* TransactionManager - Manages transaction states and interactions with the database.

3. Entity (Data Layer) Classes:

Represents the core data and business logic layer of the application. Entity classes represent data objects that interact with the database.

Examples:

* Customer - Stores customer details.
* PaymentDetails - Stores payment method details.

Q3. Place these classes on a three tier Architecture.

Answer:

Presentation Layer: Boundary Classes (e.g., Payment Screen).

Business Logic Layer: Controller Classes (e.g., PaymentController).

Database Layer: Entity Classes (e.g., Customer, Payment)

Q4. Explain Domain Model for Customer making payment through Net Banking

Answer :

A Domain Model is a conceptual representation that defines the structure, relationships and behaviors of entities within a specific problem domain.



 Q5. Draw a sequence diagram for payment done by Customer Net Banking

Answer:

A Sequence diagram is type of interaction diagram used in software engineering and system design to illustrate how processes operate with one another and in what order.

This diagram shows how the objects in the system interact and communicate with each other with time to achieve specific task



Q6. Explain Conceptual Model for this Case

Answer:

A conceptual model is a representation of a real-world system or concept that helps people understand it. Conceptual models are often visual and can be used in many fields.

A conceptual model is a high-level representation of a system that helps in understanding, visualizing and communicating the essential aspects of a domain. It provides a clear and simplified view of a domain, making it easier to understand.

Key Elements of Conceptual Model:

1. Entities - Customer, Product, Order and Payment
2. Attributes – customerID, name, email, phonenumber
3. Relationships – a customer places an order.

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

Answer -

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

View – Represent the presentation layer of the application.

Model – Represent the data and the business logic of the application.

Controller – Acts as an intermediary between Model and View.

Rules to derive the classes from use case diagram-

1. Combination of one actor and one use case results in one boundary class. Combination of two actor and one use case results in two boundary class. Combination of three actor and one use case results in three boundary class.

2. Use case will result in controller class.

3. Each actor will result in one entity class.

Guidelines to place classes in 3-tier architecture-

1. Presentation Layer-This layer is nothing but a user interface. View is inside this layer. These classes interact directly with the user. Presentation layer is responsible for displaying information and also receiving the input from the user.
2. Application Layer-This layer is nothing but business logic. Model and controller are inside this layer. Controller handles the user input, process the request and co-ordinates interaction between the model and view.
3. Data Layer-Classes which are responsible for data access and storage are in this layer. It contains the classes which interacts with the database, tiles and other data sources.

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

Answer -

Stages & BA Role:

1. **Requirement Gathering** → Documents Business Requirement Document (BRD), gathers stakeholder requirements.
2. **Design** → Works on UI/UX, Use Case Diagrams, and System Architecture.
3. **Implementation** → Ensures business goals are met by monitoring development progress.
4. **Testing** → Validates system functionality and ensures compliance with requirements.
5. **Deployment & Maintenance** → Supports go-live processes and manages documentation updates.

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

Answer -

The **Thomas-Kilmann Model** defines **five conflict resolution strategies:**

* **Competing** (Win-Lose) → One party enforces their viewpoint.
* **Collaborating** (Win-Win) → Both parties work together for a mutual solution.
* **Compromising** (Partial Win-Win) → Middle-ground approach.
* **Avoiding** (Lose-Lose) → Ignoring the conflict.
* **Accommodating** (Lose-Win) → One party gives in.

In Scrum, **collaboration** is encouraged to resolve conflicts effectively.

Q10. List down the reasons for project failure

Answer -

Below are the listed reasons for project failure.

* Unclear or changing requirements.
* Poor communication.
* Inadequate planning.
* Lack of resources.
* Unrealistic deadlines.
* Technical issues.
* Scope creep.
* Lack of stakeholder involvement.
* Poor risk management.

Q11. List the Challenges faced in projects for BA

Answer -

Business Analysts (BAs) often face challenges like:

* conflicting stakeholder expectations
* changing requirements, scope creep
* lack of domain knowledge, poor communication
* managing complex data, unrealistic deadlines
* inadequate resource allocation
* difficulty eliciting accurate requirements

 Q12. Write about Document Naming Standards

Answer -

Document Naming Standards for a Business Analyst (BA) are a set of guidelines that ensure consistency and clarity when naming project documents, facilitating easy identification, organization, and retrieval of information throughout the project lifecycle.

Consistent naming conventions are crucial. Examples:

* ProjectName\_DocumentType\_Version\_Date.docx - (e.g., PaymentSystem\_RequirementsSpec\_v1.2\_2024-10-27.docx)
* Use descriptive names.
* Include version numbers.
* Use consistent date formats.
* Use abbreviations sparingly.

Q13. What are the Do’s and Don’ts of a Business analyst

Answer -

As a business analyst, your role is to help organizations make informed decisions by analyzing data and providing insights. However, to be successful in this role, there are some important do's and don'ts that you should be aware of. In this blog post, we'll explore some of the keys do's and don'ts of being a business analyst.

**Do's:**

1. **Understand the business**: To be an effective business analyst, you need to have a deep understanding of the business you're working in. This means understanding its goals, objectives, and operations. By understanding the business, you'll be able to identify the right data to analyze and provide insights that are relevant to the organization.
2. **Listen carefully**: As a business analyst, you'll be working with different stakeholders in the organization, including business leaders, developers, and project managers. It's important to listen carefully to their needs, concerns, and feedback. This will help you understand their requirements and provide solutions that meet their needs.
3. **Be data-driven**: As a business analyst, you'll be working with a lot of data. It's important to be data-driven in your approach, which means using data to support your recommendations and decisions. This will help you provide insights that are based on facts and evidence rather than opinions.
4. **Communicate clearly**: To be an effective business analyst, you need to be able to communicate clearly and concisely. This means presenting data in a way that is easy to understand and using clear and jargon-free language. This will help you to ensure that your recommendations are understood and acted upon.
5. **Be adaptable**: The business environment is constantly changing, and as a business analyst, you need to be adaptable. This means being able to adjust your approach and methods as needed to meet changing business needs.

**Don'ts:**

1. **Make assumptions**: As a business analyst, it's important to avoid making assumptions. Assumptions can lead to incorrect conclusions and recommendations that don't meet the needs of the organization. Instead, focus on gathering and analyzing data to support your recommendations.
2. **Ignore feedback**: Feedback is important in any business environment, and as a business analyst, you need to be open to feedback from stakeholders. Ignoring feedback can lead to missed opportunities and solutions that don't meet the needs of the organization.
3. **Use jargon**: Using technical jargon can be confusing for stakeholders who may not be familiar with the terminology. Avoid using jargon and instead focus on communicating in a language that is easy to understand.
4. **Overcomplicate things**: As a business analyst, it's important to keep things simple. Overcomplicating things can make it difficult for stakeholders to understand your recommendations and may lead to too complex or difficult-to-implement solutions.
5. **Work in isolation**: Business analysis is a collaborative effort, and as a business analyst, you need to work with different stakeholders in the organization. Avoid working in isolation and instead collaborate with others to ensure that your recommendations meet the needs of the organization.

Q14. Write the difference between packages and sub-systems

Answer -

In software development, a "package" is a collection of related code elements used for organization and grouping, while a "subsystem" is a larger, more functional unit within a system, typically composed of multiple packages, representing a distinct part of the overall system with specific responsibilities and behavior; essentially, a subsystem is a higher-level concept than a package, providing a more defined functional boundary within the system.

In software engineering, packages are collections of files and headers, while subsystems are collections of packages. Both help organize and structure code.

Packages -

* Group related model elements, such as source files and headers
* Can be used to model system architecture by representing layers or subsystems
* Can be represented in UML package diagrams using notations like rectangular shapes with tabs

Subsystems -

* Are collections of one or more packages
* Can be used to partition a system
* Can be represented in a subsystem model, which is similar in structure to the overall system model

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

Answer -

Camel-casing is a writing style where words are joined together without spaces, with the first letter of each word capitalized except for the first letter of the entire compound word, essentially creating a "hump" effect similar to a camel's back; it's most commonly used in computer programming to name variables, functions, and classes, making code more readable by visually separating words within a single identifier.

Key points about Camel-casing:

* How it looks: "myVariableName", "getSomeData", "iPhone"
* Where it's used: Primarily in programming languages like Java, JavaScript, C#, where it's a standard naming convention for variables, functions, and classes
* Reason for the name: The capital letters in the middle of a Camel-cased word resemble the humps on a camel's back

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

Answer -

A "development server" is a dedicated server environment where software developers can build, test, and iterate on applications without affecting the live production system; a business analyst typically has access to view the application in its development stage, test functionalities based on the latest features, provide feedback on the user interface, and review data within the development environment to ensure alignment with business requirements, but usually does not have direct administrative access to modify code or system configurations on the development server.

Access for Business Analysts:

1. Testing functionalities: Business analysts can access the development server to test new features and functionalities, providing feedback on user experience, usability, and whether the features meet the business needs.
2. Data review: They can access sample data within the development environment to analyze how the application handles different scenarios and validate data integrity.
3. User interface evaluation: Business analysts can review the user interface design and provide feedback on its intuitiveness and alignment with user requirements.
4. Requirement verification: They can verify that the development progress is adhering to the documented business requirements by testing different functionalities.

Q17. What is Data Mapping

Answer -

Data mapping is the process of connecting data from multiple sources into a unified data model. It's a key step in data management that helps to integrate, transform, and analyze data.

How data mapping works -

1. Data mapping matches fields from one database to another.
2. It helps to standardize data and reduce errors.
3. It creates a visual representation of how data from one source corresponds to data from another.
4. It helps to define relationships between data elements.

Why data mapping is important -

1. It helps to make data more accessible and understandable.
2. It helps to reduce redundancies and ensure more accurate analysis.
3. It helps to streamline communication between different databases.
4. It helps to prevent errors and redundancies.
5. It helps to secure sensitive data.

When data mapping is used -

1. Data mapping is used in data migration, data integration, and other data management tasks.
2. It's also used in software development and integration projects.
3. It's used when setting up new business intelligence (BI) tools.
4. It's used in Electronic Data Interchange (EDI) file conversion.

Q18. 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

Answer -

API stands for "Application Programming Interface," which essentially acts as a set of rules and protocols that allows different applications to communicate and exchange data with each other.

How to use API integration for date format conversion:

* Receive data from the US application:

When your application receives a date from the US application through its API, the date will be in the "mm-dd-yyyy" format.

* Data parsing and conversion:

On the receiving end: Your application's code should first parse the received date string into its individual components (month, day, year) using the "mm-dd-yyyy" format.

* Date manipulation:

Then, rearrange these components to create a new date string in your desired "dd-mm-yyyy" format.

* Store or display the converted date:

Once the date is converted, your application can store it in its database or display it to the user in the "dd-mm-yyyy" format.