**Nurturing Process - Capstone Prep Exam 3 –Part ½**

**Q1. Draw a Use Case Diagram**

Use Case Diagram

* A Use case is a high-level diagram.
* The main purpose of the diagram is to identify the requirement.
* A use case diagram is an actor specific.
* A use case diagrams are designed to explain how an external user are interacting with the system.

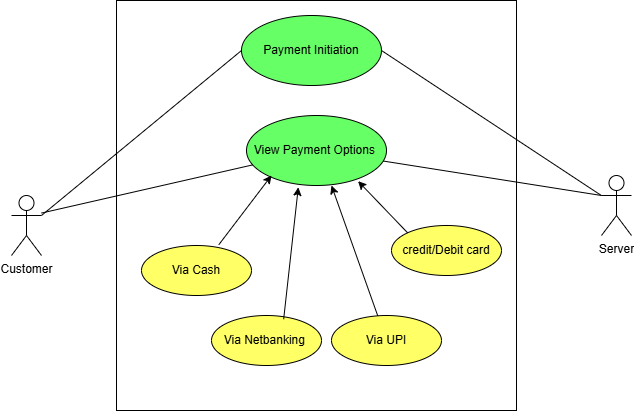
**Primary Actor** directly interacts with the system.

**Association** is a relationship between actors and use case.

**Secondary Actor** supports the system.

**Essential Use Cases –  
makes sense and completeness to the end user.**

**Supporting Use Cases –  
makes sense and supports Essential Use case.**



**-------------------------------------------------------------------------------------------------------------------------**

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

* Boundary Class - used to handle interactions between the system and external actors  
  Ex: Payment Option Boundary  
                       Card Payment Boundary
* Controller Class - act as intermediaries between boundary and entity classes.  
  Ex: Payment Initiated Controller  
                       Card Payment Controller
* Entity Class - represent the core data and business logic of the application.  
  Ex: Customer  
         Payment

**---------------------------------------------------------------------------------------------------------------------**

**Q2:Place these classes on a three tier Architecture. -**

This divides the application into 3 logical layers:

1. Application Layer:

* Topmost layer of the architecture - also known as "Presentation Layer"
* Handles user interface (UI) components such as screens, pages.
* -Payment Method Selection Boundary  
  -Card Payment Boundary

2. Business Logic Layer:

* Middle layer of the architecture
* Acts as an intermediary between the presentation layer and the data storage layer
* Layer contains the core logic of the application.
* -Payment Controller  
  -Wallet Controller

3. Database Layer:

* Bottom-most layer of the architecture
* Responsible for storing and retrieving data.
* -Customer (Entity Class)  
  -Payment (Entity Class)

**-----------------------------------------------------------------------------------------------------------------**

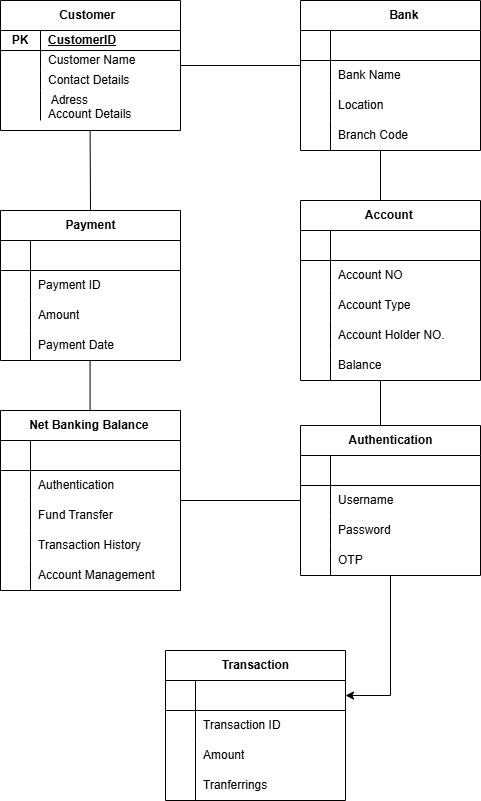
**Q4:** Explain Domain Model for Customer making payment through Net Banking

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

It is used in software development to help understand and design systems that accurately reflect business processes.

**Key Aspects of a Domain Model:**

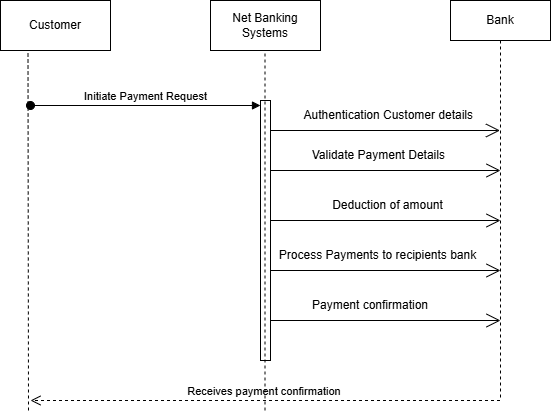
1. **Entities (Objects):** These are the main components of the system, representing real-world concepts.
   * Example: In healthcare, entities could be Patients, Doctors, Appointments, and Prescriptions.
2. **Attributes:** These are properties that define an entity.
   * Example: A Patient entity may have attributes like Name, Age, Gender, and Medical History.
3. **Relationships:** Define how entities are connected to each other.
   * Example: A Doctor can have many Appointments with multiple Patients.
4. **Business Rules & Constraints:** Rules that govern interactions within the domain.
   * Example: A patient cannot book two appointments at the same time.



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

A sequence diagram is a type of interaction diagram used in software engineering and systems design to illustrate how processes operate with one another and in what order."

This refers to a sequence diagram, commonly used in UML (Unified Modelling Language), which visually represents the sequence of messages or interactions between entities in a system. It helps understand the flow of control and data between different components over time.



**-----------------------------------------------------------------------------------------------------------------**

Q6. Explain Conceptual Model for this Case

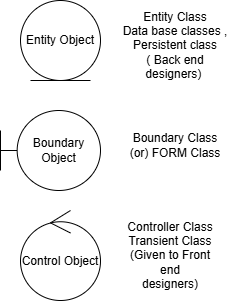
* A conceptual model is a high-level representation of a system that helps in understanding, visualizing, and communicating the essential aspects of a domain.
* It provides a clear and simplified view of the domain, making it easier to understand.
* Key Elements of a Conceptual Model:
  + Entities -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

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

This highlights the importance of a conceptual model in simplifying and communicating the structure of a system. It helps to visualize relationships between entities, their attributes, and how they interact within a domain

Guidelines to place identified MVC Classes in a 3 Tier Architecture

➢Place all Entity Classes in DB Layer

➢Place Primary Actor associated Boundary Class in Application Layer

➢Place Controller Class in Application Layer

➢If governing Body influence or Reusability is there with any of remaining Boundary Classes , place them in Business Logic Layer else place them in Application Layer

**-----------------------------------------------------------------------------------------------------------------**

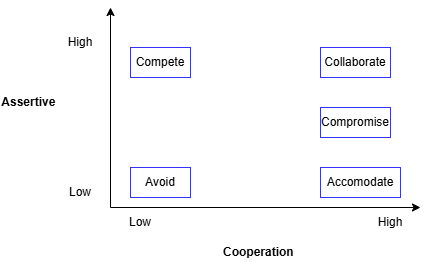
Q8. Explain BA contributions in project (Waterfall Model – all Stages)

|  |  |  |
| --- | --- | --- |
| STAGES | Activities | Artifacts and resources |
| Pre project | **Enterprise Analysis** – SWOT Analysis, GAP Analysis, Market Research, Feasibility Study, Root Cause Analysis, Decision Analysis, Strategy Analysis, Enterprise Architectural Frameworks, Project Scope and Business case writing, Risk analysis | Business Case SOW (Statement of Work)  PO (Purchase Order)  Sr. BA, Business Architects Pre-sales Consultants |
| Planning & Estimations & Assessment Project Kick Off (Big Picture Plan) | 1. Understand Assumptions and Constraints along with Business Rules and Business Goals  2. Plan Packages for Big Projects  3. Understands the project plan from PM  4. BA conducts stakeholders Analysis  5. **Plan BA approach strategy** (Req. gathering techniques, communication, Req. mgmt., Documents to follow, Tools to use, Change Request Handling methodology) for this Project | PM  Sr. BA |
| Requirements Gathering | 1. Stakeholders identify and document  2. Client gives BRD or **BA prepares BRD** by interacting with Client – Brainstorming, Document Analysis, Reverse engineering, Interviews, workshops, Focus Groups, Observation, Questionnaires.  3. Prototyping can be used by BA to make the Client to give more specific requirements  4. Sort the gathered Requirements (avoiding duplicate Reqs , grouping into similar functionality or into modules)  5. Prioritize requirements – Moscow  6. Validate Requirements - FURPS | BRD (Business Requirements Document) |
| Requirement Analysis | 1. Draws UML Diagrams (Use case and Activity Diagrams)  2. Prepares Functional Requirements from Business Requirements  3. All Architects comes up with Technical Requirements (SSD)  4. SRS will have Functional Requirements and Technical Requirements  5. Takes Signoff on SRS from Client. SRS is the first legal binding Doc between the Business and the technical Team  6. BA prepared RTM from SRS before Design phase starts. (BA is the owner of RTM).  7. BA traces how requirements are dealt in each phase of development life cycle from Design till UAT | Functional Requirements Specification SSD (Supplementary Support Document)  SRS (Software Requirements Specification)  RTM (Requirements Traceability Matrix) |
| Design | 1. From Use case Diagram , Test Manager or BA will **prepare Test Cases**  2. Communicates with Client on the design and Solution documents (updates Status to Client and make them understand how the solution would look like to prepare them to drive UAT)  3. BA will initiate the **preparation of End user manuals**  4. updates RTM  5. From Use case Diagram Solution-Architect recommends Architecture of the IT solution  6. DB Architect uses Persistence Classes (Entity Classes) and comes up with ER Diagrams or DB Schema.  7. GUI Designer will look into Transient Classes (Boundary Classes) and designs all possible Screens for the IT Solution | Solution Document Design Document – HDD – ADD |
| Coding | 1.BA organizes JAD Sessions  2. BA clarifies queries of Technical Team during Coding  3. Developers refer Diagrams and Transient (Controller Classes) of BA and code their unit  4. Update End user manuals  5. Update RTM  6. Conducts regular Status meetings with technical team and the Client and tuning Client for participation in UAT | LDD – CDD Application |
| Testing | 1.BA- Prepares Test Cases from Use Cases or assists Test Manager to do so  2. BA performs high level testing  3. BA prepares Client for UAT  4. Test Data is requested by BA from Client  5. Updates End User Manuals  6. Updates RTM  7. Take signoff from Client on Client Project Acceptance form | Test Concerning Documents Application with less errors |
| Deployment and Implementations on | 1.Forwards RTM to Client or the PM which should be attached to the Project Closure Document  2. Coordinates to complete and share End User Manuals  3. Plans and Organizes Training Sessions for End Users 4. Prepares Lessons learned from this project (to take precautions for coming projects |  |
|  |  |  |

**-----------------------------------------------------------------------------------------------------------------**

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

* **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 for 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.
  + Check for every possible solution for the conflict.
  + Negotiate the solution to avoid future conflicts.
* **X-axis (Cooperation):** Measures how much you consider others’ interests.
* **Y-axis (Assertiveness):** Measures how strongly you push for your own interests.



**Five Conflict Styles:**

**Competing (High Assertiveness, Low Cooperation)**

**Collaborating (High Assertiveness, High Cooperation)**

**Compromising (Moderate Assertiveness, Moderate Cooperation)**

**Avoiding (Low Assertiveness, Low Cooperation)**

**Accommodating (Low Assertiveness, High Cooperation)**

**-----------------------------------------------------------------------------------------------------------------**

Q10. List down the reasons for project failure

1. **Poor Planning** – Lack of a well-defined roadmap and structured approach leads to inefficiencies, missed deadlines, and project derailment.
2. **Unclear Objectives and Requirements** – Vague or constantly changing project goals result in misalignment among team members and stakeholders.
3. **Inadequate Risk Management** – Failure to anticipate and mitigate potential risks can cause disruptions, cost overruns, and project delays.
4. **Poor Communication** – Ineffective communication between teams, stakeholders, and clients leads to misunderstandings and incorrect project execution.
5. **Scope Creep** – Uncontrolled expansion of project scope beyond initial requirements results in budget overruns, missed deadlines, and resource exhaustion.
6. **Lack of Stakeholder Engagement** – Insufficient involvement of key stakeholders results in misaligned expectations, leading to conflicts and project failure.
7. **Resource Constraints** – Limited availability of skilled personnel, finances, and technology affects project execution and quality.
8. **Technical Challenges** – Unforeseen technical issues, outdated technology, or integration problems can cause project delays and failure.

**-----------------------------------------------------------------------------------------------------------------**

Q11. List the Challenges faced in projects for BA

1. **Unclear or Changing Requirements** – Frequent requirement changes lead to misinterpretation and rework.
2. **Managing Stakeholder Expectations** – Aligning conflicting interests and ensuring clear communication is challenging.
3. **Scope Creep** – Uncontrolled additions to scope impact timelines and budgets.
4. **Time and Resource Constraints** – Limited time, budget, or skilled personnel affect project execution.
5. **Quality Assurance and Testing** – Ensuring requirements are met and tested properly to avoid defects.
6. **Documentation and Knowledge Management** – Maintaining clear documentation to avoid miscommunication and knowledge gaps.
7. **Technology Constraints** – Integrating legacy systems and adopting new technologies pose technical challenges.

**----------------------------------------------------------------------------------------------------------------**

Q12. Write about Document Naming Standards

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

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

Project ID: PROJABC

Document Type: REQ

Version: 1.0

Date: 2025-02-11

The document identifier could be: PROJABC-REQ-1.0- 2025-02-11

**-----------------------------------------------------------------------------------------------------------------**

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

|  |  |  |
| --- | --- | --- |
| Sr. No. | DO'S | DON'TS |
| 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. Listen carefully and completely until the client is done, and then you can ask queries. | There is no word as "By default". |
| 3 | Try to extract maximum leads to the solution from the client himself. | Never imagine anything in terms of GUI. |
| 4 | Concentrate on the important requirements. | Don't interrupt the client when he is giving you the problem. |
| 5 | Question the existence of existence./ Question everything. | Never try to give solutions to the client straight away with your previous experience and assumptions. |

**-----------------------------------------------------------------------------------------------------------------**

Q14. Write the difference between packages and sub-systems

* Packages: Collection of components which are not reusable in nature.

Ex: Application development companies work on Packages.

* Sub systems: Collection of components which are reusable in nature.

Ex: Product development companies work on Sub systems.

**-----------------------------------------------------------------------------------------------------------------**

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

* Camel-casing is a naming convention used in computer programming.
* It is used for naming variables, functions, and identifiers.

Example:

* CamelCase: camelCaseExample

In Camel casing, the first word starts with a lower-case letter and each subsequent word begins with an uppercase letter

**-----------------------------------------------------------------------------------------------------------------**

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

* 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.
* **-----------------------------------------------------------------------------------------------------------------**

Q17. What is Data Mapping

* Data mapping is the process of connecting data from one source to another.
* It's like creating a guide or map that shows how data in one place corresponds to data in another place.
* This is especially important when you're moving data between different systems or databases to ensure that the data stays consistent and accurate.

In essence, data mapping is the blueprint for how data will be moved or transformed between different sources. It's the key to making sure that your data is usable and reliable, especially when dealing with complex systems and data sources.

**-----------------------------------------------------------------------------------------------------------------**

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

An **API (Application Programming Interface)** is a set of rules and protocols that allow different software applications to communicate with each other. APIs define how requests and responses should be structured so that systems can exchange data efficiently and securely.

**API Integration**

API integration is the process of connecting two or more applications using APIs to enable seamless data exchange. This is commonly used in:

* **Third-party services** (e.g., payment gateways, social media logins).
* **Data synchronization** (e.g., integrating a CRM with an ERP).
* **Automation** (e.g., fetching real-time stock prices or weather updates).

If your application expects dates in **dd-mm-yyyy** format but receives data from a US-based application in **mm-dd-yyyy** format, you need to implement a date transformation process before storing or processing the data.

**Solution Approach**

**Receive the API Data**: with a date field formatted as **mm-dd-yyyy**.

**Extract & Transform the Date Format**: Before saving it into your system, convert **"dob" "12-25-1990" (MM-DD-YYYY) → "25-12-1990" (DD-MM-YYYY)**.

**Store or Use the Transformed Data**.

**-----------------------------------------------------------------------------------------------------------------**

**-----------------------------------------------------------------------------------------------------------------**