**Case Study**

**A Customer can make payment either by card, or by wallet or by cash or by net banking**

**Q1. Draw a Use Case Diagram**



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

To identify classes from Use case diagram, we apply MVC rules on each use case to derive Classes.

**Boundary Classes: (Used to handle interaction between the system and external actors)** Boundary Classes are type of class in Software design particularly in Object Oriented Analysis and Design, which act as intermediaries between the system and external entities. These external entities could be users, external systems or devices. The role of boundary class is to handle the interaction between the system and outside world.

Boundary class (All Use cases) combination of 1 actor and use case is one boundary class.

combination of 2 actors and use case is two boundary class.

combination of 3 actor and use case is three boundary class, and so on.

And those actors should be primary actors. Primary actors means the actors who initiated the user case and interact with the system.

Eg- Customer Registration, Customer Log in, Bank server Log in, Customer Logout and Bank server Logout.

**Controller Classes: (Act as intermediaries between boundary and entity classes,)** Controller classes are type of class in object-oriented analysis and design that handle the flow of control and manage the interaction between boundary classes (responsible for system interface) and entity classes (responsible for data and business logic). Their primary role is to process incoming request, coordinate activities and direct data between other component of the system.

Controller class handles user input and process the data, Use case will be considered as the controller classes.

Eg- Registration controller, Login controller, Payment Controller, Credential Controller, Net banking Controller, Email Controller and Logout Controller.

**Entity Class: (Represent the core data and business logic of the application).** Entity Classes are type of class in software design, particularly in Object oriented analysis and design, that represent the core business objects of a system. These classes encapsulate data and logic (business rules) that operates on the data.

Entity class (All actors) each actor will be considered as one entity.

**Eg-**Customer, Payment

**Q3. Place these classes on a three tier Architecture**

Three tier is also known as multi-tier architecture, is a software design pattern that. This application divided into 3 logical layers. Application layer, Business logic layer and Data layer.

**Application Layer-** This is a topmost layer of the architecture. This layer is also known as Presentation Layer. It focuses and handles user interface (UI), components such as scree, pages. This layer is responsible for displaying information to the user and capturing users’ action. This layer communicates with business layer and submit data for processing.

Ex- E Commerce, Website.

**Business logic Layer-** This is middle layer of architecture, its acts as intermediary between presentation layer and data storage layer. This layer consists of the core logic of the application.

Ex- (Third party application objects) Printer, email server, payment gateway.

**Database Layer-** It is bottom layer of architecture and it is responsible for storing and retrieving data. Data layer would keep information about product, customer order, inventory and transaction in database.

Ex- Oracle data base, MY SQL

**As per given case study - Place these classes on a three tier Architecture**

**Application Layer –** PaymentMethodSelectionBoundary, CardPaymentBoundary

**Business Logic Layer (Primary actors associated with the Boundary class)-** PaymentController,WalletController

**Data Layer (All the entity classes)-** Customer(EntityClass), Payment(Entity Class)

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

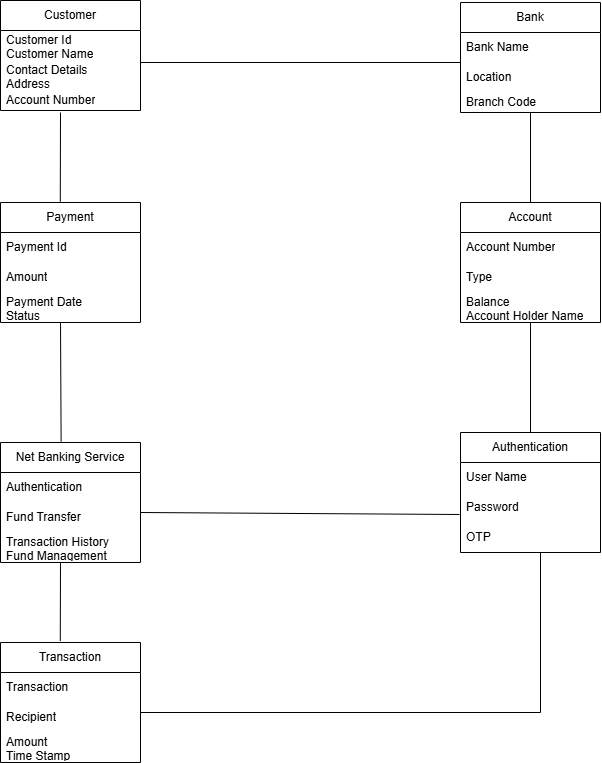
A domain Model is a Conceptual representation that defines the structure, relationships and behaviors of entities within a specific problem domain, it helps to understand and design the core functionality of a system by modelling real world objects and their interaction. The domain model typically visualized using UML (Unified Modelling Language) class diagrams or similar tools.

Domain model is similar to the entity relationship model. The tables are connected to each other.

In the below diagram, the customer table is connected to bank, which is why the customer is able to make payment. Customer table is also connected to payment table, because he should make the payment.

Now the payment is done by net banking, so payment table is connected to the bank table. The account is in the bank., so the account table is connected to the bank table, because authentication is to performed there. Also, the authentication table is connected to transaction table, because authentication will be done while transaction.

**Diagram**

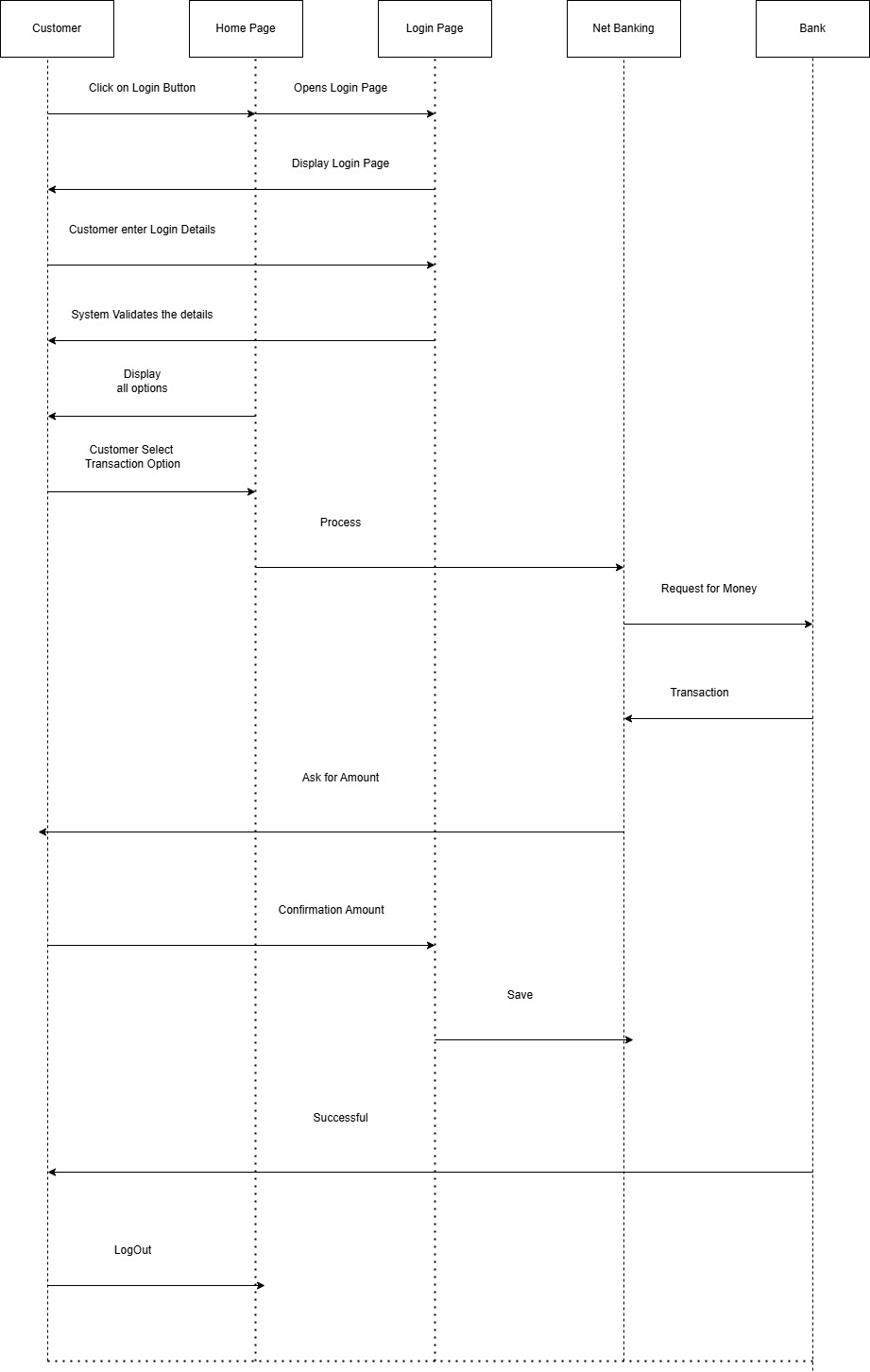


**Q.5 Draw Sequence diagram for payment done by customer Net Banking.**

UML diagram is known as language of diagram. The base of UML diagram is object-oriented approach. UML has 9 diagrams 5 static (Use case, class component, package, deployment) and 4 dynamics (Sequence, activity, state chart, collaboration). Sequence diagram come under Dynamic diagram.

In dynamic diagram we model the time where as in static diagram we do not model the time.

Below sequence diagram shows how the objects in the system interact and communicate with each other with time to achieve specific task. Developer will draw this. It is used to show the flow of message, events or actions between the objects of the system



**Q6. Explain Conceptual model for this case.**

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

A conceptual model is a high-level representation of a system that helps 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.

It Provides a clear and simplified view of the domain, making it easier to understand.

As per case study, Key element of conceptual model.

entities- Customer product order payment.

Attributes -Customer ID, Name, email, phone Number.

Relationships- Customer place 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.**

To identify classes from use case diagram, we apply MVC rules on each use case to derive classes.

MVC is a design pattern where, the application is devided into 3 logical parts. Model, view and controller.

**Model-**

The Model represent the data and business logic of the application.

Model is responsible for multiple tasks like managing the application’s data, performing data, validation, implementing business rules and handling data access operations.

Model does not depend on how the data is presented or how the user interacts with the application.

The Model class is known about all the data that is needed to be displayed.

This layer corresponds to the data related logic that the user works with.

It represent the data is being transferred between View and Controller.

It can add or retrieve the data from the database.

**View-**

The view is responsible for representing the data to the user for handling the user interface.

The View can be a web page, a desktop application window or any other form of user interface.

It retrieves input from the user and passes it to the controller for processing.

It represents the presentations of the application.

It takes the data from the Model and renders it in a way is suitable for the users display or interaction.

**Controllers-**

The controller acts as an intermediary between the Model and the View.

It retrieves input from the user (Via View), process the input by invoking the appropriate methods in the Model, and then updates View with the data or state.

The Controller handles user interactions, interprets user input, and translate it into instructions for the Model or the View.

Controllers works on the users request.

Takes input from the user/ client

It interacts with the Model and view.

Controllers class represents as use case.

Controllers acts as a Mediators between models and data base.

**MVC Architecture Rules**

Combination of One actor and use case results in One boundary class.

Combination of Two Actors and a use case results in Two boundary classes.

Combination of Three Actors and a use case results in Three boundary classes & SO On.. Only one primary actor is to be considered with a use case.

Use case will result in a controller class

Each Actor result in one entity class.

**Guidelines to place classes in 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.

Consider the examples of online shopping application with the following UseCsease.

Model Classes- Customer, Payment,Netw Banking, Cardd, Cash

View Classes- Login View, PaymentOptionView, NetBankingView, BankSelectionView,CredentialsView,PaymentAmountView,PaymentConfirmationvIEW,LougOutView.

Controller Classes- Login Controller, PaymentOption Controller, NetBanking Controller BankSelection Controller, Credentials Controller, PaymentAmount Controller, PaymentConfirmation Controller, LougOut Controller.

**Q8. Explain BA contribution in Project (Waterfall Model- all stages)**

The Sequential Waterfall Model is aa step by step approach to software development where each phase must be completed before moving on to the nixed one. It is called Waterfalled because progress flows downwards, like a waterfall. This model is common and classic of life cycle models, also referred to as a linear sequential life cycle model. This model is very simple and easy to understand, explanation of BA contribution in Project (Waterfall Model- all stages) as per below

|  |  |  |
| --- | --- | --- |
| **Stage** | **Activities** | **Artifacts &Resources** |
| **Pre Project** | Enterprises Analysis- SWOT Analysis, GAP Analysis, Market Research, Feasibility Study, Root Cause Analysis, Decision Analysis, Strategy Analysis, Enterprises Architectural Frameworks, Project Scope, Business Case Writing, Risk Analysis | Business Case, SOW (Statement of Work), PO (Purchase Order), Sr.BA, Business Architects Pre sales consultants. |
| **Planning** | 1. Understand assumption and constraints along with business rules and business goals. 2. Understand the project plan from PM. 3. BA Conducts stakeholder Analysis. 4. Plan BA approach strategy (Re. Gathering techniques, communication, documents to follow, tool to use, change request handling methodology) for this project | PM , Sr. BA |
| **Requirements Gathering** | 1. Stakeholder Identify and document. 2. Client give BRD or BA prepares BRD by interacting with client- Brainstorming, Documents 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 req, grouping into similar functionality or into modules). 5. Prioritize requirements- MoSCoW. 6. Validate requirements- FURPS | BRD (Business Requirements Documents) |
| **Requirements Analysis** | 1. Draw UML Diagrams (Use Case & Activity diagrams). 2. Prepares Functional requirements from business requirements. 3. All architecture come up with the technical requirements (SSD). 4. SRS will have functional and Technical requirements. 5. Take Signoff on SRS from the client. SRS is the first legal binding Doc between the business and technical team. 6. BA Prepare RTM from SRS, design phase starts (BA is the owner of RTM). 7. BA trace how requirements are dealt in each phase of development life cycle from Design till UAT. | Functional Requirements Specification. SSD (Supplementary Support Documents. SRS (Software requirements Specification). RTM (Requirements Traceability Matrix) |
| **Design & Development** | 1. From Use case diagram, Test manager or BA will prepare Test case. 2. Communicate with the client on the design and solution documents. 3. BA will initiate the preparation of the end user manuals. 4. RTM Updates. 5. Comes up with ER diagrams or DB Schema. 5. GUI designer will look into boundary classes and design all possible screens for the IT Solution. | Solution Document, Design Document-HDD-ADD |
| **Testing** | 1. BA Prepares Test Case from Use Case or Assist Test Manager to do so. 2.In this testing phase, the software is tested as a whole to ensure that it meets the requirements is free from defects, testers are involved I this phase, teste case documents are generated here. 3. BA performs high level testing | Test Concerning documents. Application with less errors. |
| **UAT** | 1.BA Prepares client for UAT. 2.Updates End User Manual and RTM. 3.Take sign off from client-on-client project acceptance form | UAT &Project ready for Deployment |
| **Deployment and Implementations** | 1.BA forwards RTM to client or the PM which should be attached to the project closure document. 2.Coordinate to complete the share end user manuals. 3.Plans and organises training sessions for end users. 4.Prepare lessons learned from this project. |  |

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

Conflict can occur due to various reason, such as difference in goal, values, personalities, resources or communication breakdowns. Conflict is part of any workplace and it is important to resolve it to promote learning and growth.

Conflict management is nothing but the process of identifying and addressing conflicts in healthy and constructive manner.

Conflict management is the process od resolving conflicts or disagreements between individuals or group in a constructive manner.

**5 option of conflict managements are**

Competing, Avoiding, Accommodating, Collaborating, Compromising

**Thomas Kilmann conflict resolution technique.**

Thomos Kilmann is technique is a widely used tool for assessing conflict resolution style & guiding individuals in selecting appropriate strategies to manage conflict.

X Axis Cooperation and Y Axis Assertiveness.

As a BA before we convince somebody we need to understand according to Thomas Kilman conflict resolution graph, where the person lies. Whether they are in avoiding state, competing state or accommodation state, we need to understand. Based on that we have to actually drag them to the collaborate mode. If not collaborate, at least bring them to the compromise mode.

In the graph we have cooperation with the business analyst on the X Axis from low to high and the assertiveness the process knowledge of the domain knowledge on the Y axis from low to high it is.

When the cooperation is low and assertiveness is low then people try to avoid us.

When the cooperation is high and assertiveness is low then they can accommodate us.

When the cooperation is low and assertiveness is high then they compete with us, that means as BA you tell once process and they will tell different process.

When the cooperation is high and assertiveness is high then, then we go to collaborative mode.

When the cooperation is medium and assertiveness is medium then we will going with compromise mode.

**Managing conflicts-**

Conflict required high level of energy for resolution and

Managing conflict is all about maintaining the relation

Before engaging in conflicts, we need to think about expected behhavior.

**5 steps to 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 the future conflicts.

**Q10. List down challenges faced in Projects for BA**

There might be several internal or external reason for project failure. Major reasons explanation as per below.

**Improper requirement gathering**

If the requirements of the project are not gathered correctly, then this can lead to project failure.

**Continuous changes in requirements**

If the requirements keep on changing frequently, this can also project failure. Because the scope of the project will also keep on changing which will lead project failure.

**Lack of Involvement**

A project can fail if stakeholders are not participating in the process. The project stakeholder input and feedback play very important role to mee the goals. Their lack of involvement can cause project failure.

**Lack of executive support-**

Lack of support within team can cause project failure, such as their unavailability or not spending enough time during projects.

**Unrealistic Expectations**

Sometime client state unrealistic expectations means goals can not be achieved or goals are out of the scope, can cause project failure.

**Improper Planning**

The project can fail the planning is not done properly, the milestone, goals should be discussed, if there is no proper planning, then team may face difficulties in addressing the issues or track the progress.

**Resource Constraints**- Lack of resource or unavailability of required resource, result could be not able to achieve project goals and could be the reason for project failure

Others reasons such as poor planning, unclear objectives and requirements, inadequate risk managements, poor communication, technical challenges, Scope creep could be the reason for project failure.

**Q11. List the challenges faced in projects for BA**

**Lack of training-** When team member or even the BA don’t have enough training or experience, it becomes hard to understand what the project needs how to use certain tools. This can show everything down and lead to mistake.

**Obtaining sign off on requirements**- It can be challenging to get all stakeholders to agree and approve the documented requirements, Delay in sign off can cause the project to rework later.

**Change management-** with respect to cost and timeline – During the project, client might want to change things. These can make the project more expensive or take longer, upsetting the timeline or budget is a bigger challenges.

**Coordination between developers and testers**- Developers create solution and tester make sure it works properly, but if they don’t communicate well or misunderstand each other, it can lead a problem, like bugs not being fixed or delays in testing.

**Conducting meeting-** Meetings are important to share updates and discuss issues, project progress, updates. Sometimes they can go off topic or take too long, or absenteeism in meetings. For BA its hard to keep meetings focused and useful.

**Making sure status reporting is effective-** Ensuring the project updates are clear, accurate and timely can be difficult, effective status reports are crucial to keeping all stakeholders informed.

**Driving clients for UAT completion-** Getting clients to complete User Acceptance Testin UAT on time can be challenge, delay in UAT can push back the projects delivery timeline.

**People management (coordinating with different people and different teams)-** Is a critical task it requires patience, adaptability and strong communication skill. The BA has to coordinate with many people, client, developer, tester, managers, each with their own task and way of working. Keeping everyone at the same page and resolving conflict can be difficult.

**Overall making sure project health is good shape and delivered as per the time lines without any issues-** Monitoring all aspects of the projects to ensure it is on track, within budget and meeting quality expectations is a constant responsibility. The BA must handle risk, issues to keep the project on course.

**Q12. Write about document naming standards.**

Document naming standards are used to save the file with particular names or format. This is important in sharing and keeping track of data files.

All documents will be named using some standards

[ProjectID][DocumentType]V[X]D[Y].ext

Example-

Product ID - PQ786

Document Type- BRD

Version- 1.0

Date-2025-01-21

The document identifier could be PQ786-BRD-1.0-2025-01-21

**Q13 What are the Do’s and Don’t of Business Analyst**

As business analyst we need to collaborate with different stakeholder, BA has to keep below do and don’t point.

**Do’s**

Consult an SME for clarification in requirements

Go to the client with a plan with no assumptions. Listen carefully snd completely until the client is done and then you can ask queries.

Try to extract maximum leads to the solution from the client himself.

Concentrate on the important requirements.

Question the existence of existence / question everything

**Don’t**

Never say No to the client

There is no word as By default

Never imagine anything in terms of GUI

Don’t interrupt the client when he is giving you the problem.

Never try to solution to the client straight away with your previous experience and assumption.

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

Collection of similar Objects is a Class

Collection of Classes is Components

Difference between packages and sub-systems are Collection of components which are not reusable in nature are known as package

Collection of components which are reusable in nature known as Subsystems.

**Example-** Product base companies work on Subsystem (Microsoft, Oracle, IBM, SAP) and application development companies (Infosys, TCS, Wipro)work on Packages.

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

Camel casing ia a way of writing words or phrases where the first word starts wit a small letter and every other word starts with a capital letter, without space between words.

Examples:

myFirstName

totalCostOfItem

The name “Camel Case” comes from the way the capital letters in the middle look like the humps of a camel.

Business Analyst camel casing is sued in requirements documentation, BA often use camel casing to name of entities like Use case, features, user stories like validate customer details.

While documenting business process or workflow, camel casing can be used

The database tables name also uses camel casing

Requirement naming camel casing is sued in requirement document also, to name the functional and nonfunctional requirements.

By using camel casing in the documents, it helps to maintain consistency in the entire document and also increase readability.

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

Server is a space, means we store our data.

From the IT Perspective we will have 3 servers’.

1 Development Server- Is generally own by IT Company.

2 UAT Server- Own by Client

3 Production Server- Own by client

Development server is basically devided into 2 parts.

1 Document- Public and protective

2 Technology- Coding, Testing & DB (Data base)

A development server refers to a dedicated environment that is used during the software development process.

It provides platform for the developers and the testers to build, test, develop and debug the application.

The access a BA has are-

Read Only- BA’s may be granted with the read only access to the development server.

This will allow them to view the user interface of the application, navigate through the features and also they will be able to observe the behavior of the application.

Limited access-

Depending upon the project needs, the BA’s will be granted limited access to the specific modules in the application.

Limited configuration Access means BA have the authority to make changes in certain areas of application where they have the access.

**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 we are moving data between different systems or database to ensure that the data stays consistent and accurate.

Data base contain multiple tables in it. There may come scenario, where we need to map the data from one table to another.

Data mapping is nothing but a process to establish connection between multiple data sources.

The purpose of data mapping is to ensure that the data is accurately transferred or converted into different format.

Purpose of data Mappings are

**Data Integration –** Means combine data from different sources (e.g two data base).

E.g Match similar information “First Name” in one system and “Name” in another , so they fit together.

**Data Migration-** Means move data from one system to another, e.g upgrading software.. It’s ensure the old system data e.g “ Date of Birth” correctly fits into the new system format “DOB” . This avoids errors or lost lost information during transfer.

**Data Transformation-** Means change the format, structure or content of data e.g converting “ 21-0-1-2025” into “Jan 21, 2025”. Its define how the original data should be modified to fit the new format or rules.

**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**

API Stands for Application Programming Interface. API is third party. We need to understand the third party addressal to the system for that we can use API. API it is way for two different software programs to communicate and share data with each other. API allows to access feature or data from another application without knowing its internal details. Think of an API like a bridge between two applications.

**Example of API Integration**

If I want to fetch date from US website and the format would be like “mm dd yy” but as per Indian format would be “dd mm yy”.

So, the translation to US font to Indian font that approach done through API. Generally, we integrate through the third-party API.

To solve these issues, we can use API Integration to handle the date conversion, here how.

1. Receive Data from the Other Application-

My application will use the API to get data from the US application. For E.g the API might send date like 01-22-2025 (22nd January 2025 in the US format).

2. Process the Data

Application will read this date, identify is as mm-dd-yyyy, and then convert it into my application format (dd-mm-yyyy), after conversion the date will look like 22-01-2025.

3. Use the Converted Data.

Once the date is in my format, I can use in my application ( for displaying to users or storing in my database)

**Steps for API Integration.**

Request the Data- Use an API endpoint URL provided by the US application to fetch the data.

Parse the Date- API convert in to your format

Send or Display the data- Once the data is converted, can use in your application as needed..

In short API ensure smooth communication between both application, even with different date format.