Case Study 1 (Q1-Q6 24 Marks)

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 - 4 Marks





Answer :

Boundary Classes :

The Boundary class is a class that is the boundary of the system and other system or user (which is actor in the use case diagram).

The followings are the feature of the Boundary class.

1. This class is more easy to be changed than the Entity and Control class.

2. The attribute of this class and screen layout are defined at the basic design.

3.In a class diagram, there are cases that the stereotype (<<boundary>>) is added.

4.In a class diagram, there are cases that is shown by the following icon.

Controller classes :

The followings are the feature of the Control class.

1. This class has a few attribute.

2.In a class diagram, there are cases that the stereotype (<<control>>) is added.

3. This class is a class to achieves use cases in the Use case diagram.

4.In a class diagram, there are cases that is shown by the following icon.

Entity Classes :

The Entity class is a class that has data.

The "E" of the ER diagram means "Entity" too, if you know the ER diagram, you easily understand.

The following are the feature of the Entity class.

1. There are many cases that this objects of this class are perpetuated 1 in the DB.

2. The extraction of the class is like ER diagram 2.

3. This class is related to the DOA (Data-oriented approach) 2.

4. The module cohesion of this class is high 3, and is not easy to be changed.

5.In a class diagram, there are cases that the stereotype (<<entity>>) is added.

6.In a class diagram, there are cases that is shown by the following icon.

Q3. Place these classes on a three tier Architecture. - 4 Marks

Answer -

Application Layer	Customer Registration Customer Login Bank server login

Business logic layer	Customer
(Primary actors associated with Boundary class)	Bank Server
Data base Layer (All the entity classes)- All actors	 Customer Bank server Cash Card Net Banking

In this three-tier architecture, the application tier handles the user interface, the

Business Logic Layer manages the Business logic and coordinates between the other tiers, the

Data layer tier handles data storage and retrieval.

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

Answer'



A Domain Model is like an Entity-Relationship (ER) Model because it shows how different entities (or tables) are related to each other. Here's how the entities are connected in the example:

- The Customer table is linked to the Bank table. This connection allows the customer to make payments.
- The Customer table is also connected to the Payment table because the customer needs to make a payment.
- The Payment table is connected to the Net Banking table since payments are made using net banking.
- The Account table is connected to the Bank table because accounts are held in banks.
- The Authentication table connects the Net Banking and Bank tables because authentication is required for secure access.
- The Authentication table also connects to the Transaction table because authentication happens during a transaction.

Differences Between ER Diagram and Domain Model

- 1. Attributes Inside Boxes:
 - ER Diagram: Does not include attributes inside the boxes.
 - Domain Model: Includes attributes inside the boxes.

- 2. Purpose:
 - ER Diagram: Focuses on designing databases and showing how data will be stored and retrieved.
 - Domain Model: Focuses on representing real-world entities and their behavior in the application.

3. Use Case:

- ER Diagram: Mainly used for database design.
- Domain Model: Used throughout the software development process to guide application design.

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

Answer -



This 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 ow of messages, events or actions between the objects of

the system. This diagram helps to visualize the behaviour of the system by

showing the process in detail

Explain Conceptual Model for this Case - 4 Marks

Conceptual Model of Net Banking Payment System

Purpose

The conceptual model provides an understanding of:

- Key concepts and their relationships.
- The overall structure of the net banking payment system.
- A foundation for designing the database schema, application architecture, and system functionalities.

Entities and Their Relationships

Customer

Represents the users of net banking services. Key aspects include:

- Service Awareness: Customers should know about available net banking services and their features.
- 2. Privacy of Data: Ensures customer data is protected and confidential.
- 3. **Technology Awareness**: Customers should understand and feel comfortable using the technology.
- 4. Trust & Support: Quality services build customer trust and provide necessary suppor

Bank

Represents the service provider responsible for net banking services. Key aspects include:

- 1. **Online Information**: Accurate and up-to-date online details should be provided about net banking services.
- 2. Security & Privacy: Strong security policies safeguard customer data and transactions.

- 3. **Infrastructure**: Reliable technological infrastructure (hardware and software) is critical for delivering services.
- 4. Policies: Clear policies and regulations ensure compliance, customer protection, and trust.

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

Ans. MVC is a design pattern where, the application is divided into 3 logical parts-

Model, View and Controller.

Each of these parts will have specific responsibility.

Model-

The Model represents the data and the 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 represents the data that is being transferred between View and Controller. It can add or retrieve the data from the database

It responds to the controller's request because the controller cannot interact with the database by itself.

The model interacts with the database and give the requested data. All the model classes are nothing but the entities. Model classes are represented as entity class.

View-

The View is responsible -for presenting 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 receives input from the user and passes it to the Controller for processing.

It represents the presentation of the application. View refers to the model.

It takes the data from the Model and renders it in a way that is suitable for the user's display or interaction.

For rendering the data, it uses query method. View does not depend upon application logic. View class are represented as boundary class.

Controller-

The Controller acts as an intermediary between the Model and the View. It receives input from the user (via the View), processes the input by invoking the appropriate methods in the Model, and then updates the View with the new data or state.

The Controller handles user interactions, interprets user input, and translates it into instructions for the Model or the View. It coordinates the flow of data between the Model and the View, ensuring that they remain separated and independent of

Whenever the user requests for anything, that request first goes to the controller. Controller works on the users request.

Takes input from the user/ client. It interacts with the model and view. Controller class represents as use case. Controller acts as a mediator between model and database

Controller cannot directly get the data from the database. So, controller interacts with the model.

Advantages of MVC-

MVC has the feature of scalability, which in turn helps the growth of application. The components are easy to maintain. A model can be used by multiple views that provide reusability of code. By using MVC, the application becomes more manageable. As all the three layers are different and independent, they are maintained separately

Rules to derive the classes from use case diagram-

- 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.

Consider the example of Online shopping application with the following use case:

Model Classes-Customer, Payment, Net Banking, Card, Cash

View Classes-Login View, Payment Option View, Net Banking View, Bank Selecton View, Credentials View, Payment Amount View, Payment Confirmation View, Logout View

Controller Classes-Login Controller, Payment Option Controller, Net Banking Controller, Bank Selection Controller, Credentials Controller, Payment Amount Controller, Payment Confirmation Controller, Logout Controller

Guidelines to place classes in 3-tier architecture-

Presentation Layer-This layer is nothing but a user interface. View is inside this layer

Tier architecture: - has 3 layers

- o Application layer
- o Business logic layer
- o Data base layer

Data flowing from the 3 layers explains the 3-tier architecture where the information is passed. Here in elicitation techniques 3 tier architecture works as a medium where as a BA we get to know the information shared by the requester will it fit in application layer, business logic layer or data base layer.



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

Ans.

Waterfall model is useful in the situation where the project requirements are well defined and the project goals are clear.

- Waterfall model follows sequential approach.
- In this model each phase is completed entirely and then only moved to the next phase.
- Waterfall model relies on documentation to ensure that the project is well defined and project team is working toward clear goals.
- Ones that particular phase has been completed and ones we move to the next phase, we cannot go back to the previous phase to make changes.
- This model is stable for the projects when the requirements are clear.

Requirements Gathering-

First, the stakeholders are identified. In this phase, all the requirements are gathered from the stakeholder.BA and Project Manager participates in this phase. After completing this phase, BRD will be generated.

Requirements Analysis-The requirements are analysed to understand the scope of the project. Analysing means the BA will check all the requirements, if he founds convincing requirements then the BA will talk to the concerned stakeholder to clear it, remove the ambiguous requirements

BA will prepare functional requirement. The document which contains the functional requirements is called (FRS).[Functional Requirement Specifications] Technical team will prepare non-functional requirement. The document which contains the non-functional requirements is called (SSD). [Supplementary Support Document]BA will combine FRS and SSD to form SRS.[Software Requirement Specifications]BA will prepare RTM by referring SRS

Design-

After the requirements are cleared, Design phase starts. This has a detailed design document that outlines the software architecture, user interface, and system components', ADD and solution document will be generated here. [High-level Design Doc.]

BA Collaborate with designers, architects, and developers to translate requirements into system design.BA Ensure that the design aligns with the documented requirements and addresses stakeholder needs.

Development-

The Development phase include implementation. It involves coding the software based on the design specifications. Programmers or developer are involved in this phase. Here BA acts as a mediator between the development team and the stakeholders.BA clarifies the requirements, check if the development is going on right track or not.BA also participates in scrum meetings.

Testing-

In the testing phase, the software is tested as a whole to ensure that it meets the requirements and is free from defects. Testers are involved in this phase

Test documents are generated here.BA works with the testing team to ensure that the solution meets the requirements.BA facilitate UAT.BA helps the users to know the functionality of the system and also helps them to use the system.

Deployment-

Once the software has been tested and approved, it is deployed to the production environment.BA ensures that there is smooth transition from development phase to the production phase.

Implementation-

This is the final stage of waterfall model. It involves running the code for the very first time in production phase. Release manager handles this phase.

BA will Update documentation and requirements specifications to reflect changes in the system over time

Maintenance-Running the code for second time in the production phase is called maintenance. This is done by support team.

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

Ans.

Conflicts can occur due to various reasons, such as differences in goals, values, personalities, resources, or communication breakdowns.

Conflict is an inevitable part of any workplace. So it is important to resolve it to promote learning and growth.

Conflict management is nothing but the process of identifying and addressing conflicts in a healthy and constructive manner. It consists of strategies and techniques aimed at resolving disputes, disagreements, or differing perspectives among individuals or groups

By identifying the conflicts efficiently, it will in turn be helpful to reduce negative impact and increase positive impact.

It is a process or skill to find creative ways to handle the disagreement. Thomas – Kilmann approach is widely used to recognize the approaches for conflict management



Y axis- assertiveness X axis- co-cooperativeness

High Assertiveness and High Cooperativeness

- Collaboration- means working together to find solution

High Assertiveness and Low Cooperativeness

- Competition- means defensive, that is standing for your individual beliefs and trying to win.

Low Assertiveness and High Cooperativeness

- Accommodation- stakeholder will prioritize their needs over others.

Low Assertiveness and Low Cooperativeness

– Avoidance- means ignoring the conflict Assertiveness- the extent to which the person attempts to satisfy his own concerns.

Cooperativeness- the extent to which the person attempts to satisfy the other persons concerns.

Q10- List down the reasons for project failure

Ans. Reasons for project failure are:

Improper requirement gathering-

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

Lack of stakeholder involvement-

A project can fail if the stakeholders are not participating in the process. The stakeholders input and feedback plays very important role to meet the goals.

Ineffective or less communication-

If there are communication issues between stakeholders, team members then this can lead to misunderstandings or delays in project or even can lead to project failure.

Continuous change in the requirement-

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

Poor risk management-

Poor risk management can also lead to project failure. The team fails to identify the risks and do the risk mitigation, which can lead to unexpected challenges or delays in project. Lack of user involvement. Lack of executive support.

Unrealistic expectations-

means the goals that cannot be achieved or the goals that are out of scope

Improper planning-

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

Insufficient resources- Insufficient resources can also lead to project failure. The project may fail due to lack of technology knowledge or lack of finances.

Q11. List the Challenges faced in projects for BA

Ans

- Lack of training
- Obtaining sign-off on the requirement
- Change management
- Co-ordination between developers and testers
- Conducting meetings.
- Making sure status report is effective
- Driving clients for UAT completion
- Making sure that the project is going on right track and delivered as per the timelines without any issues
- Gathering clear and unambiguous requirements can be challenging
- Unable to understand what stakeholder is trying to convey
- Scope creep- change in requirement or scope of the project during the project lifecycle can lead to scope creep
- Managing the stakeholder with conflicting interest can be a difficult task for BA
- BA may face difficulties in understanding the requirements if the domain is not familiar to him
- Poor communication between stakeholder and BA can affect the process of gathering the information
- Technical complexity

Q12.Write about Document Naming Standards

Ans. [ProjectID][Document Type]V[x]D[y].extension

Example- [PQ777FRDV1D1.docx] or [PQ777FRD1.1.docx]

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

Ans. Never say "NO" to the client.

- There is no word called as "BY DEFAULT"
- Never imagine anything in terms of GUI
- Question everything in the world
- · Go to the client with plain mind that is with no assumptions
- Listen to the client very carefully and after he is done, then ask questions
- Don't interrupt the client.
- Never try to give solutions to the client right away.
- Try to concentrate only on important and required things.
- Be like a lotus in mud- if a client comes with a fancy requirement, then talk to the project manager first.
- Requirement hurried-project buried.
- Never criticize the stakeholder. Always appreciate the stakeholder even for small efforts.

Q14. Write the difference between packages and sub-systems

Ans. Packages- it is a group of classes or use cases that are used to organize model elements.

Packages can be nested within other packages.

These are used as containers to organize elements.

It is very useful to represent system architecture.

PACKAGE

Subsystems- it is logical grouping of related components.

It is collection of classes, packages, libraries and other sub systems that work together to deliver a specific set of functionalities

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

Ans. Camel-casing refers to the naming convention of variable, parameters or properties.

Here, multiple words are combined together.

In camel-casing, the starting letter of first word starts with small letter and other words first letter starts with capital letters.

Ex- firstName, lastName

In BA, camel-casing is used in requirements documentation.

In requirement documentation, BA often use camel-casing to name the entities like use case, features, user stories like validateCustomerDetails, calculateInterestRate, etc

Business rules, which should be satisfied by the system use camel-casing.

While documenting business process or work flows, camel-casing can be used to individual in steps.

This will help maintain consistency in the document.

The database tables name also uses camel-casing.

Requirement naming- camel casing is used in requirement document also, to name the functional and non-functional requirements.

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

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

Ans. 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 accesses 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 behaviour 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

Ans. The database contains multiple tables in it.

There may come a scenario, where we need to map the data from one table to another.

Data mapping is necessary in cases where we want quick manner.

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.

The main purpose of data mapping is-

Data integration-

While combining the data from different sources, it ensures that the data is properly matched.

Data Migration-

While migrating the data from legacy system(source) to the new system(destination), the data elements are mapped accurately into the new system.

Required techniques are applied to covert the data into the format that is required by the new system.

Data Transformation-

Data transformation means converting the data from one format to other. In data mapping, data transformation plays very important role which ensures that the data of legacy system(source) is mapped correctly to the data in new system (destination

Q18. at is API. Explain how you would use API integration in the case of your appli ion Date format is dd-mm-yyyy and it is accepting some data from Other Appl ion from US whose Date Format is mm-dd-yyyy

Ans. API stands for Application Programming Interface

It is a software intermediary that allows the two applications to communicate with each other.

It is the set of rules, protocols and tools that define how different software application should interact with each other.

API allows sharing of only necessary information and keeps the internal system details hidden, which helps the system security.

For the above scenario, Establish API communication- set up API communication between your application and other application to exchange data.

Data formatting- While sending the data from one application to other, convert the date format from dd-mm-yyyy to mm-dd-yyyy.

While receiving the data from other application, parse the data and extract the date, month and year and re-arrange them accordingly. Perform Data Validation and ensure that the converted date remains in a valid format.