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



**System**

Login to system

«uses»

Make payment

«uses»

«extends»

«extends»

«extends»

«extends»

Card

Net Banking

«uses»

Customer

«uses»

Cash

Wallet

Select net banking

Bank Server

Select Bank

«uses»

«uses»

«uses»

Enter Credential

«uses»

«uses»

Validate Credential

«uses»

«uses»

Confirm Payment

Database

Email confiramation

logout

Q2. Derive Boundary Classes, Controller classes, Entity Classes. - 4 Marks

Boundary class

A Boundary Class represents the interaction between the system and external actors (such as users, systems, or devices). It serves as the interface through which input is received and output is presented.

Think of it as the "front door" of the system — it defines how users or other systems interact with the software.

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 following are the feature of the Boundary class.

1. **This class is easier to be changed than the Entity and Controller 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 are shown by the following icon.**



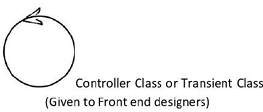
Controller classes

A Controller Class handles the processing of input, makes decisions, and coordinates the interaction between Boundary and Entity classes. It's the logic handler for a use case or a part of it.

The followings are the feature of the Controller class. This class has few attributes.

In a class diagram, there are cases that the stereotype (<<control>>) is added. This class is a class to achieve use cases in the Use case diagram.

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



Entity Classes

An Entity Class represents business objects and contains persistent data and business logic. These are typically mapped to database tables or domain models.

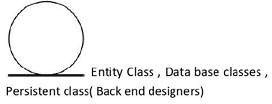
The Entity class is a class that has data. The "E" of the ER diagram means "Entity". The following are the features of the Entity class:

There are many cases that these objects of this class are perpetuated in DB. The extraction of the class is like ER diagram2.

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

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

In a class diagram, there are cases that the stereotype (<<entity>>) is added. In a class diagram, there are cases that are shown by the following icon.



|  |  |
| --- | --- |
| **Boundary class (All use cases)**  **[combination of 1 actor and a use case is one boundary class]**  **[combination of 2 actors and a use case is two boundary class]**  **[combination of 3 actors and a use case is three boundary class] and so on**  **And those actors should be primary actors.**  **Primary actors means the actors who initiate the use case and interact with the system.** | **Customer Registration Customer Login**  **Bank Server Login Customer Logout Bank Server Logout** |
| **Controller class (handles user input and process the data)**  **Use case will be considered as the controller classes** | **Registration Controller Login Controller Payment Controller Credentials Controller Net Banking Controller Email Controller**  **Logout Controller** |
| **Entity Class (All actors)**  **Each Actor will be considered as one entity** | **Customer Bank Server Cash**  **Card** |

|  |  |
| --- | --- |
|  | **Net banking** |

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

|  |  |
| --- | --- |
| **Application Layer** | **Customer Registration Customer Login**  **Bank Server Login** |
| **Business Logic Layer**  **(Primary actors associated with the Boundary class)** | **Customer Bank Server** |
| **Data Layer (All the entity classes)** | **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 tier handles data storage and retrieval

o

**/|\**

**/ \**

Actor 1

Boundary 1 Controller 1 Entity1

o

**/|\**

**/ \**

Actor 2

Entity2

Boundary 2 Controller 2 Entity3

Q4. Explain Domain Model for Customer making payment through Net Banking - 4 Marks

A domain model is a conceptual model of the domain that incorporates both behavior and data. It is a

system of abstractions that describes selected aspects of a sphere of knowledge, influence, or activity. The model then can be used to resolve problems related to that domain.

The domain model is represented of meaningful real-world concepts pertinent to the domain that need to be modeled in software. The concepts include data involved in business and rules the business uses in relation to that data.

A domain model leverages natural language of the domain. It generally uses the vocabulary of the domain, thus allowing a representation of the model to be communicated to non-technical stakeholders. It should not refer to any technical implementations such as database or software components that are being designed.

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 table, 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 netbanking table.

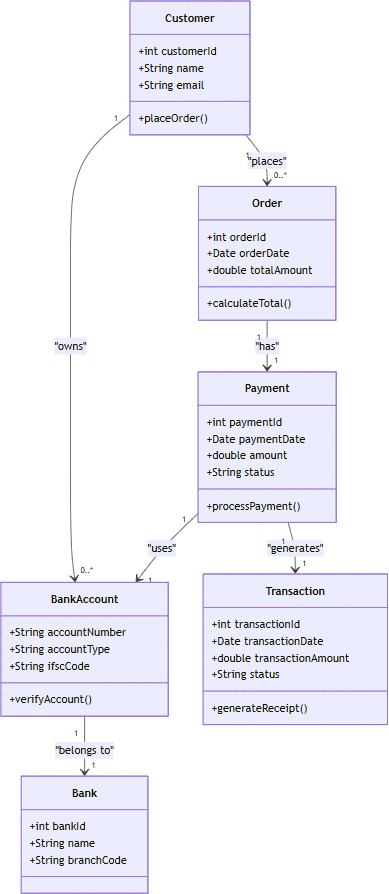
The account is in the bank, so the account table is connected to the bank table. The authentication there.

Also the authentication table is connected to transaction table, because authentication will be done while transaction. Difference between ER diagram and domain model

ER Model – do not have attributes inside the box Domain Model- do have attributes mentioned inside the box ER Model – it is a data modelling technique used in database design to represent tables.

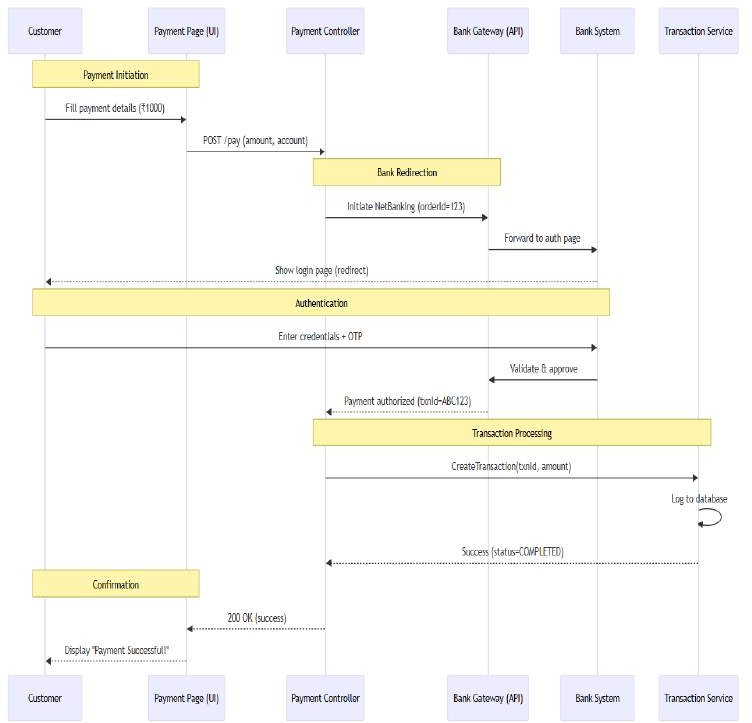
Domain Model- it is a conceptual model that represents real world entities.

ER Model – focuses on relationships required for storing and retrieving the data Domain Model- It focuses on capturing the behaviour of application



Q5. Draw a sequence diagram for payment done by Customer Net Banking - 4 Marks

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 flow of messages, events or actions between the objects of the system. This diagram helps to visualize the behaviour of the system. This diagram shows the process in detail.



Q6. Explain Conceptual Model for this Case - 4 Marks

Conceptual modeling is a representation of the business model we have. There are entities, and their

relationships among them based on which this is created. This is a Representation of a system which will be

created, and this is not a language used to communicate with technical team. Entities

In our case study, since customer is initiating online payment through net banking following entities will be there. Bank, Customer, Net banking system of a Bank, Final beneficiary (Customer’s customer).

Relationships

Customer will be initiating payment with multiple beneficiaries, so it will be One to Many Relationship. Bank System will interact with Customer, so it will be one to one relationship.

Bank will be facilitating payment for multiple customers; hence it will be One to May Relationship.

Based on above Entities and Relationships we will be creating a Entity Relationship diagram, which will be

used as a representation to take sign off from the client.

The conceptual model helps in understanding the key concepts, their relationships, and the overall structure of the net banking payment system. It serves as a foundation for designing the database schema, defining the application architecture, and implementing the necessary functionalities within the system.

The relationships between these entities can be described as follows:

1. **Customer: This node represents the customers or users of net banking services.**
2. **Service awareness: customers should be aware of the available net banking services and their features.**
3. **Privacy of data: the importance/significance of this node is to protect the privacy and confidentiality of customer data in the context of net banking.**
4. **Technology awareness: the significance of this node is that customers' should be aware and comfortable with the underlying technology used in net banking services.**
5. **Trust & Support: This node indicates that the bank provide such good services that it will help to enhance the customers trust.**
6. **Bank: This node represents a service provider responsible for offering net banking services.**
7. **Online information: This aspect highlights the importance of providing accurate and up-to-date online information about net banking services to customers.**
8. **Security & Privacy: the bank should adapt the security policies which will help the customers to keep their data related to their transaction secure and private.**
9. **Infrastructure: This component suggests that the underlying technological infrastructure, including hardware and software systems, plays a important role in enabling net banking services.**
10. **Policies: This node represents the various policies and regulations that govern the implementation and operation of net banking services, ensuring compliance and customer protection.**

Technology Awareness

Infrastructure

Policies

Trust and support

Security And Privacy

Net banking

Customer

Privacy Of data

Service Awareness

Bank

Online information

Q7. What is MVC architecture? Explain MVC rules to derive classes from use case diagram and guidelines to place classes in 3-tier architecture - 8 Marks

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 each

other. 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 MVCMVC 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

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.**

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 Selection 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.

These classes interact directly with the user. Presentation layer is responsible for displaying information and also receiving the input from the user.

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.

Data Layer

Classes which are responsible for data access and storage are in this layer. It contains the classes which interacts with the database, files and other data sources.

Model

view



controller

Presentation Tier

Order From

Order View

Order Controller

Business Logic layer

Data Tier

Order

Product

Customer

Q8. Explain BA contributions in project (Waterfall Model – all Stages) – 8 Marks

A waterfall model is very old and traditional model in IT industries. It is a progressive implementation of the projects which is divided into different phrases of SDLC.

Business Analyst will verify if the product is delivered as per the requirements, and it is meeting the business need. Maintenance: Once the implementation is done the team has to give support by installing patches, handling change requests, etc.

Stages in Waterfall Model are as follows:

* **Requirement Gathering and Analysis**
* **Designing**
* **Coding**
* **Testing**
* **Deployment**
* **Maintenance**

Requirement Gathering and Analysis: This is the initial stage of the project where an involvement of the BA is. BA is responsible for preparing BRD document (Business Requirement Document)

Artifacts: Functional Specification Document Business Requirement Document

Designing: In this phase the architect will start designing the system based on the Business analyst inputs and requirement documents. The BA helps him to clear the doubts about the requirements. Artifacts: Design Documents

UML diagrams get ready in this phase.

Coding: This phase is quite lengthy as the core development starts in this phase. Developer starts product development based on the requirement document prepared by the BA. Developer may ask questions to BA regarding the requirement, and he needs to answer the questions as and when required.

Artifacts: Code

Testing: After coding, the testing phase will start, in this phase BA helps the testing team to understand the requirements so that they will build proper functional test cases. BA has to review whether the test cases covering the whole functionality.

Artifacts: Test Cases and test results.

Deployment: Once the code is developed and tested, it is ready to deploy in the production

environment. The BA will verify the product is delivered as per the requirements and it is meeting the business needs.

Artifacts: Implementation Review document.

Maintenance: Once the implementation is done the team has to give support by installing patches,

Handling change requests etc. A BA is the person who knows every nook and corner of the project. So, every change request has to be reviewed by him and based on his inputs and reports the team will respond.

Artifacts: User Satisfaction review and change request review.

* **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 conflicting 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. HDD, 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 – 6 Marks

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.

Cooperativeness

Low

High

Accommodating

Avoid

Compromising

Compete

Collaboratin

High

Assertiveness

Y axis- assertiveness x axis- co-operativeness

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 – 6 Marks

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 – 6 Marks

Lack of training.

Obtaining sign-off on the requirement. Change management.

Co-ordination between developers and testers. Conducting meeting.

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 – 4 Marks should be Named Consistently.

1. **File names should be short (<25 characters)**
2. **Avoid special characters or spaces in a file name.**
3. **Use Capital and Underscores instead of spaces or slashes.**
4. **Use date format as per ISO 8601: YYMMDD**
5. **Include a version number.**
6. **Write down naming convention**

We must consider following naming conventions: Date of Creation

Short Description Work

Location

Project name or number Sample

Analysis

Version Number

[ProjectID][Document Type]V[x]D[y].extention

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

Q13. What are the Do’s and Don’ts of a Business analyst – 6 Marks

* **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 i.e. with no assumptions.**
* **Listen to the client very carefully and after he is done, then ask question.**
* **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 – 4 Marks

|  |  |
| --- | --- |
| **Packages** | **Sub-systems** |
| **A logical grouping of related classes, interfaces, and components within a system.** | **A self-contained, independent module or component that performs a specific function**  **within a larger system.** |
| **Organizes code for better maintainability and**  **reusability.** | **Represents a major functional unit that can**  **operate semi-independently.** |
| **Smaller in scope (e.g., utility functions, domain**  **models).** | **Larger in scope (e.g., authentication system,**  **payment processing).** |
| **Can depend on other packages within the same**  **system.** | **Often designed to minimize dependencies; may**  **communicate via APIs or interfaces.** |
| **Provides namespace management but may not**  **enforce strict boundaries.** | **Enforces strong encapsulation with well-defined**  **interfaces.** |
| **Can be reused within the same project.** | **Can be reused across different projects or**  **systems.** |
| **java.util (Java collections package)** | **A payment gateway sub-system in an e-**  **commerce platform.** |
| **Direct method calls within the same runtime.** | **May use inter-process communication (IPC),**  **APIs, or messaging.** |
| **A logical grouping of related classes, interfaces,**  **and components within a system.** | **Can be deployed independently (microservices,**  **plugins).** |
| **java.util (Java collections package)** | **A self-contained, independent module or component that performs a specific function**  **within a larger system.** |
| **Direct method calls within the same runtime.** | **Represents a major functional unit that can**  **operate semi-independently.** |
| **Deployed as part of the main application.** | **Larger in scope (e.g., authentication system,**  **payment processing).** |

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.

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- 6 Marks

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- first Name, last Name

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 validate Customer Details, calculate Interest Rate, etc

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

While documenting business process or workflows, 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? -6

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

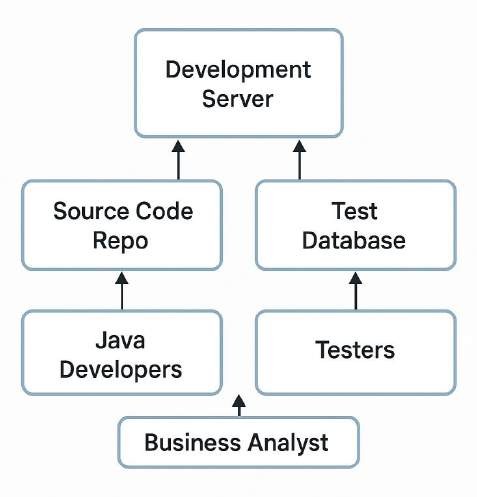
It provides platform for the developers and the testers to build, test, develop and debug the application.

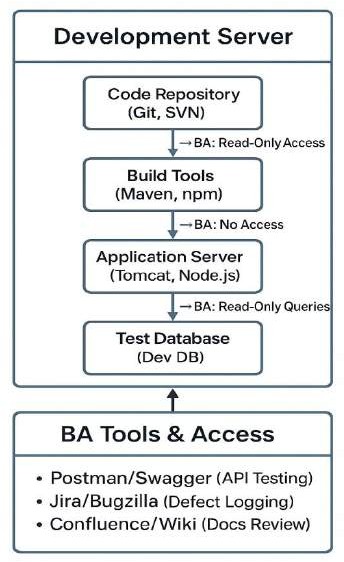
The accesses a BA has are Read Only- BA’s may be granted with the read only access to the development server.

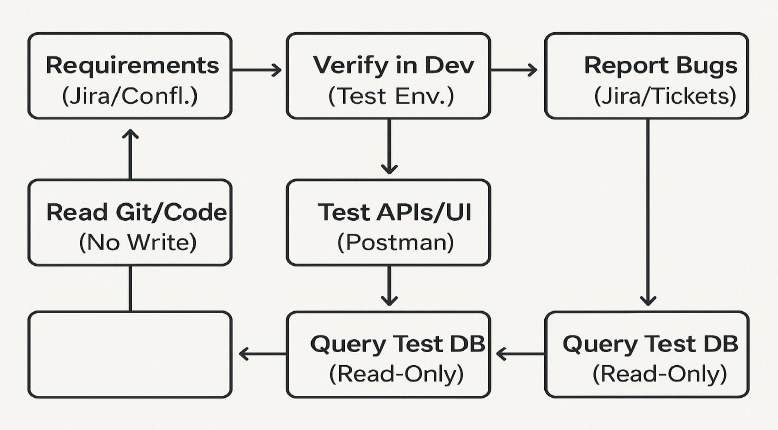
This will allow them to view the user interface of the application, navigate through the features and also they will be able to observe the behavior of the application.

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 6 Marks

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. 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

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.

Do 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.

