Q1. Draw a Use Case Diagram

A1.



Q2. Derive Boundary Classes, Controller classes, Entity Classes

A2. Boundary Classes: Combination of One Actor and an use case results in one Boundary class. In the above example customer is an actor,

* Login is an use case, so customer login (Custlogin) is one boundary class.
* Similarly, order is another use case, so customer order (CustOrder) is another boundary class.
* Customer make payment (CustPayment) is another boundary class.
* Customer order successful (Custordersuccess) is another boundary class.

Controller classes: Use cases consider as controller classes. Below are controller classes.

* Login controller class
* Order controller class
* Make payment controller class
* Order successful controller class

Entity Class: Each actor considered as entity class. Below are entity class.

* Customer entity class
* Database entity class
* Bank server entity class

Q3. Place these classes on a three tier Architecture

A3. CC- Login, Order, Order successful, Make payment

 BC- CustLogin, CustOrder, CustOrderSuccess Application Layer

 BC- CustPayment Business logic Layer

 EC- Customer, Database, Bank Server Entity class Data Layer

(CC-Controller class, BC-Boundary class, EC-Entity class)

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

A4. Domain model is a conceptual representation that defines the structure, relationships, and behaviours of entities within a specific problem domain. Domain model is similar to entity relationship model. The tables are connected to each other.

In the below diagram, customer table is connected to bank, through which customer is able to make payment. Customer table is connected to payment table, payment is done by net banking. So these are connected. Account is in bank, so account table connected to bank table. Authentication is connected to both net banking and bank table, since authentication is to be performed there. Authentication table connected to transaction table, since authentication will be done while transaction.



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

A5.



Q6. Explain Conceptual Model for this Case

A6. Conceptual model is a high-level, abstract representation of a system, concept, or idea, focusing on the relationships between key elements rather than specific implementation details. It's used to communicate and understand complex ideas in a simplified way.

The relationship between the entities are shown below:

Customers login to the application, order the food, and select the net banking payment mode. Net banking once selected, it will direct with the bank, initially it will authenticate the details with the bank, validate them and then check the balance in the account and funds get transferred to the customers request amount. Transaction got completed and confirmation message sent to the customer.



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

A7. MVC (Model-View-Controller) is a design pattern used in software development. The application is divided into 3 logical parts – model, view, controller. Each of these parts will have specific responsibility. To identify classes from use case diagram, we apply MVC rule on each use case to derive classes.

Model: The Model component in the MVC (Model-View-Controller) design pattern demonstrates the data and business logic of an application. It is responsible for managing the application’s data, processing business rules, and responding to requests for information from other components, such as the View and the Controller. All model classes are represented as entity classes.

View: Displays the data from the Model to the user and sends user inputs to the Controller. It is passive and does not directly interact with the Model. Instead, it receives data from the Model and sends user inputs to the Controller for processing. View class is represented as boundary class.

Controller: Controller is responsible for intercepting the request from view and passes it to the model for the appropriate action. After the action has been taken on the data, the controller is responsible for directing the appropriate view to the user.

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. Note: only one primary actor is to be considered with a use case.

3. Use case will result in a controller class

4. Each Actor will result in one entity class.

Guidelines to place identified MVC Classes in a 3 Tier Architecture

1.Place all Entity Classes in DB Layer

2.Place Primary Actor associated Boundary Class in Application Layer

3.Place Controller Class in Application Layer

4.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)

A8. A waterfall model is a traditional and sequential model in software development projects. It is a progressive implementation of the projects which is divided into different phrases of SDLC.

The business analyst will verify the product is delivered as per the requirements and it is meeting the business need. BA is responsible to prepare requirement traceability matrix, conducting UAT acceptance and sign off from client.

Stages in Waterfall Model:

 1. Requirement Gathering and Analysis

 2. Designing

 3. Development - Coding

 4. Testing

 5. Deployment

 6. Maintenance

1. Requirement Gathering and Analysis: This is the initial stage of the project where BA is involved. BA is responsible for gathering requirements from stakeholder and analysing them. Preparing BRD document (Business Requirement Document)

2. 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.

3. Coding: This phase is quite lengthy as the core development starts in this phase. Developer start 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.

4. 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.

5. 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.

6. 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.

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

A9. Conflict management involves effectively addressing and resolving disagreements, and the Thomas-Kilmann technique (X axis - Cooperation, Y axis - Assertiveness) offers a framework to understand five distinct approaches: competing, collaborating, compromising, avoiding, and accommodating.

High

Assertive

 Low

Low Co-operation High

This model describes the two core dimensions while choosing a mode of conduct in a situation of conflict: ‘assertiveness’ and ‘cooperativeness’. Assertiveness is the extent to which you try to solve and resolve for your preferred outcomes. Think of this as the factor on the Y-Axis of a graph. On the other hand, Cooperativeness is the level to which you try to resolve the other party’s problems. This is the factor on the X-Axis of the graph.

Competing: This mode is an assertive and non-cooperative, focusing on pursuing one's own interests at the expense of others.

Collaborating: This mode is assertive and cooperative, aiming to find a solution that satisfies all parties involved.

Compromising: This is moderately assertive and cooperative, seeking a mutually acceptable solution where all parties make concessions.

Avoiding: This is non-assertive and non-cooperative, characterized by withdrawing from the conflict or postponing discussion.

Accommodating: This is non-assertive and cooperative, prioritizing the needs and concerns of others over one's own.

5 steps of conflict management:

1. Identify Conflict
2. Discuss the details
3. Agree with root problem
4. Check for every possible solution for the conflicts.
5. Negotiate the solution to avoid the future conflicts.

Q10. List down the reasons for project failure

A10. Reasons for Project Failures:

1. Improper requirement gathering – if the requirements of the project are not gathered correctly, then this can lead to project failure.
2. Continuous change in requirements – 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.
3. Lack of user involvement – A project can fail if the stakeholders are not participating in the process. The stakeholder’s input and feedback plays’ very important role to meet the goals.
4. Lack of executive support – Lack of timely support from executive like not providing sufficient resources who are well knowledge in the required technology or domain can lead to project failure.
5. Unrealistic expectation – The goals that cannot be achieved or out of scope.
6. Improper planning – If the planning is not done properly the project can lead to failure. The milestones, goals should be discussed. If there is no proper planning, then the team may face difficulties in addressing the issues or track the progress.
7. Ineffective communication: If there are communication issues between stakeholders, team members then this can lead to misunderstandings or delay in project or even can lead to project failure.
8. Poor risk management: If the team fails to identify the risks and do the risk mitigation, which can lead to unexpected challenges or delays in project.

Q11. List the Challenges faced in projects for BA

A11. Challenges faced by BA in projects:

1. Lack of training
2. Obtaining sign-off on requirement
3. Change Management - with respect to cost and timelines
4. Coordination between developers & testers
5. Conducting meetings
6. Driving client for UAT completion
7. People Management (coordinating with different people and different teams)
8. Managing the stakeholder with conflicting interest can be a difficult task for a BA.
9. BA may face difficulties in understanding the requirements if the domain is not familiar to him.

Q12. Write about Document Naming Standards

A12. A document naming standard is a systematic approach to assigning unique identifiers to various documents created and used throughout the development process. All documents will be named using some standards like

[ProjectID][Document Type]V[X]D[Y].ext

Example: [PQ847FRDV1D1.docx]

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

A13.

1. Never say NO to client
2. Never imagine anything in terms of GUI
3. There is NO word called as “By Default”
4. Consult an SME for clarifications in Requirements
5. Question everything in the world.
6. Go to client with plain mind with no assumptions.
7. Listen to client very carefully, and after he is done, then ask questions.
8. Don’t interrupt the client in between.
9. Never try to give solutions to the client right away.
10. Try to concentrate on important and required requirements.
11. Try to extract the leads to solution from client itself.

Q14. Write the difference between packages and sub-systems

A14. Packages: Collection of Components which are not reusable in nature is called packages. Components are collection of classes.

Sub-systems: Collection of Components which are reusable in nature.

Example: Product Development Companies like Microsoft etc work on Subsystems and Application Development Companies like TCS etc. work on Packages.

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

A15. Camel-casing refers to a naming convention for writing file or object names using joined words with at least of those words beginning in a capital letter.

In camel-casing the entire first word will be in lowercase and subsequent words first letter should be in Upper Case. There will be no gap in between words. Example: getEmpId(); turnLeftAndSlowDown();

Camel casing used for naming variables, functions and identifiers. It also used in requirement documentation, BA often use to name entities like use case, features, user stories etc. Business rules which should be satisfied by the system use camel-casing. While documenting business process or workflow, camel-casing can be used to individuals in steps. This will help to maintain consistency in document and increases readability. Database tables also use camel-casing.

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

A16. A development server is a type of server that is designed to facilitate the development and testing of programs, websites, software or applications for software programmers. It provides a run-time environment, as well as all hardware/software utilities that are essential to program debugging and development. A development server is the core tier in a software development environment, where software developers test code directly. It is comprised of the essential hardware, software and other components used to deploy and test the software under development, including bulk storage, development platform tools and utilities, network access and a high-end processor. Upon testing completion, the application is moved either to a staging server or production/live server. It provides platform for the developers and testers to build, test, develop and debug the application.

Business Analyst has the visualizing access or read only access in development server. This will allow them to view the user interface of the application, navigate through the features, they will be able to observe the behaviour of the application. BA has the access to all the functional servers and not to the technical servers.

Q17. What is Data Mapping

A17. Data mapping is the process of identifying and matching data fields between different sources or databases to ensure data integrity and consistency during data migration or integration. Data mapping helps ensure that when data is moved or integrated from one system to another (e.g., from a legacy system to a new database, or between different applications), the correct data fields are linked, preventing data loss or corruption.

Before data can be analysed for business insights, it must be homogenized in a way that makes it accessible to decision makers. Data now comes from many sources, and each source can define similar data points in different ways. For example, the state field in a source system may show Illinois as "Illinois," but the destination may store it as "IL."

Data mapping bridges the differences between two systems, or data models, so that when data is moved from a source, it is accurate and usable at the destination.

Data mapping has been a common business function for some time, but as the amount of data and sources increase, the process of data mapping has become more complex, requiring automated tools to make it feasible for large data sets.

Main purpose of data mapping is

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

Data migration: While migrating data from source to the destination the data elements are mapped accurately to the new system.

Data transformation: Means converting the data from one format to other. In data mapping, data transformation plays a very important role which ensures that the source data is mapped correctly to destination data.

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

A18. API stands for application programming interface. It is a set of defined rules that enable different applications to communicate with each other. API acts as a bridge or intermediary between two or more software applications, enabling them to interact and exchange data or services. APIs define how applications can request data or services from each other, and how they should format their requests and responses.

For the given scenario, set up API communication between your application and other US application to exchange data.

Data formatting: while sending data from one application to other, convert the date format from dd-mm-yyy to mm-dd-yyyy. While receiving data from other application, parse the data and extract the date, month and year and rearrange them accordingly. Perform data validation and ensure that the converted date remains in a valid format.