**Prep 3 Exam**

**Q. 1 A customer can make a payment either by Card or by Wallet or by Cash or Net banking.**

**Draw use case Diagram**



**Q.2 A customer can make a payment either by Card or by Wallet or by Cash or Net banking.**

**Derive Entity class, Controller class and Boundary class**

Payment CC

Customer EC

Admin EC

Admin payment BC

Customer payment BC

**Q.3 A customer can make a payment either by Card or by Wallet or by Cash or Net banking.**

**Place the classes on a Three- tier Architecture**

Three tier architecture is a software development model that an application into three logical tiers.

A 3-tier architecture is a modular client-server architecture that consists of

1. Application layer
2. Business logic layer
3. Database layer
* **Application layer**

It is the topmost layer of an application. This is the layer we see when we use software. By using this layer, we can access the Webpage. The main functionality of this layer is to translate tasks and results to something that the user can understand. It communicates with business logic layer. Graphical user interfaces like screens and pages, validations on pages, organization specific business logic will be on the application layer.

* **Business logic layer**

Business logic layer coordinates the database layer, processes command and sends required information to the application layer. It controls an application’s functionality by performing detailed processing. This layer acts as mediator between application layer and database layer. All reusable components, frequently changing components, governing body rules and regulation, appliances should go to this layer. For e.g., Printer, payment gateways, mail servers, RBI rules for banks, IRDA rules for insurance, etc.

* **Database layer**

The data is stored in database layer. Business logic layer communicates with database layer to retrieve the data. Database components connecting to database will be at the data layer.

Customer payment BC Admin payment BC Payment CC

**Application layer**

**Business Logic layer**

 Customer EC Admin EC

**Data layer**

**Q.4 A customer can make a payment either by Card or by Wallet or by Cash or Net banking.**

**Explain domain model for customer making payment through net banking**

 Domain modeling is also known as conceptual modeling. It depicts the concepts that are easily identifiable in the problem description. It is a package containing class and activity diagrams.

A domain model is a formal representation of a knowledge domain with concepts, roles, data types, individuals, and rules, typically grounded in description logic.



**Q.5 A customer can make a payment either by Card or by Wallet or by Cash or Net banking. Draw Sequence diagram for customer making payment through Net banking**

**Q.6 A customer can make a payment either by Card or by Wallet or by Cash or Net banking.**

**Explain Conceptual model for this case.**

 A conceptual model is representation of a system. It consists of concepts used to help people know, understand or simulate a subject the model represents. It is an application that the designers want the users to understand. By using the software and perhaps reading its documentation, users build a model in their minds of how it works.

**Q.7 What is MVC Architecture? Explain MVC rules to derive classes from use case diagram and guidelines to place classes in 3tier architecture.**

**MVC architecture** – Model-View-Controller. This is used to identify classes from use case diagram.

**Model Class** -The model class knows about all the data that needs to be displayed. It only represents the data of an application. All model classes are represented as entity classes. This represents database.

**View** - View represents the presentation of the application. Refers to the model and uses query methods to obtain the contents. It is the data required by the query. It is represented as boundary class or form class.

**Controlle**r - whenever a user sends a request for something then it always goes through the controller. He is responsible for intercepting the request from view and passes it to model for appropriate action. After the action taken on data, the controller is responsible for directing the appropriate view to the user. In GUIs, the views and the controllers often work closely together.

**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 and so on...
* Use case will result in a controller class.
* Each actor will result in one entity class.

**MVC GUIDELINES**

* Place all entity class in database layer and Place controller class in application layer.
* Place primary actor associated boundary class in application layer.
* If governing body influence or reusability there with any remaining boundary class, place them in business logic layer else place them in application layer.

**Q.8 Explain BA contribution in a project-Waterfall model all stages**

Waterfall model is an old and traditional model. It is progressive implementation of the projects which is divided into different phases of SDLC. It is very simple to understand and use. Here each phase must be completed in its entirety before the next phase can begin. At the end of each phase a review takes place to determine if the project is on the right path and whether we should continue or discard the same.

Contribution of a BA in different stages of waterfall model is as below

* **Requirement gathering** - From the initial stage the BA is responsible to prepare the BRD by interacting with client via brainstorming, interviews, workshops, questionnaires, observation. The stakeholders identify the documents. Prototyping can be used to make client give more specific requirements. Then BA has to sort the requirements, prioritize them –MoSCoW method and finally validate the requirements-FURPS.
* **Requirement analysis** - here the BA draws the UML diagrams. Prepares functional requirements from business requirements. SRS will have the functional and technical requirements. BA takes sign off on SRS from client and takes RTM from SRS before design phase starts. Finally, he traces how requirements are dealt in each phase of the development cycle from design till UAT.
* **Design** - in this phase the architect will start designing the systems based on the BA’s inputs and requirement documents. The BA helps him in clearing the doubts. BA will initiate the preparation of the end user’s manual and updates RTM.
* **Development**-coding - BA organizes JAD sessions. He clarifies queries of technical team during coding. Developers refer to diagrams of the BA and code their unit. BA has to update end user manuals and RTM. BA has to conduct regular status meetings with technical team and client and tune client to participate in UAT.
* **Testing** - Here the BA helps the testing team to understand the requirements to build proper functional test cases. BA has to review if the test cases cover the whole functionality or not. BA updates end user manuals and RTM and takes sign off from clients on the Client project Acceptance form.
* **Deployment and implementation** - Once the code is developed and tested, it is ready to deploy in production environment. The BA has to verify that the product is delivered as per the requirements and business needs. BA has to forward RTM to client which should be attached to product closure document. BA has to organize training sessions for end users. Finally, he prepares lessons learnt from the project.

**Q.9 What is Conflict management? Explain using-Thomas Kilmann technique**

Conflict management can be defined as the process of dealing with (perceived) incompatibilities or disagreements arising from, for example, diverging opinions, objectives, and needs. The purpose of conflict management is to teach conflict resolution skills, such as managing conflict, self-awareness about the different formats of conflict and produce effective communication while in conflict with a team member.

In the 1970s, researchers Kenneth Thomas and Ralph Kilmann developed a model for conflict resolution. It was called the Thomas-Kilmann model after them. Under this model, the term ‘conflict’ is described as the condition in which people’s concerns can’t be compared with the others. If two or more people or groups care about things that are contradictory to each other, then the outcome is conflict.

**Thomas Kilmann’s 5 modes of handling conflict**

* **Competing** - It is assertive and non-co-operative. It refers to addressing only one’s concerns at the cost of concerns of the other. Competing is defensive. It strictly means standing up for your individual beliefs and simply trying to win.
* **Accommodating** - It is accepting and co-operative. The element of self-sacrifice is highlighted in this mode.
* **Avoiding** - It is both unassertive and un cooperative. The individual wants to neither address their own problems nor the problems of others. This ultimately means that they do not want to engage in the conflict at all. Avoiding might be seen at times as a diplomatic move involving bypassing or ignoring the issue
* **Collaborating** - Collaborating, the most beneficial outcome in the Thomas-Kilmann conflict model. It is both assertive and cooperative. This mode is the complete opposite of avoiding. Collaborating includes a voluntary effort to work alongside the opposition to find a perfect solution that wholly addresses the collective problem
* **Compromising** - The last outcome in the Thomas-Kilmann conflict model falls on the average point on both the assertiveness and cooperativeness scales. The goal here is to find a mutually acceptable and robust solution that, in some ways, satisfies both the individuals. It comes midway between competing and accommodating.

**5 steps of conflict management -**

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

**Q.10 List down the Reasons for project failure**

Reasons or project failure is as below -

* Improper requirement gathering
* Continuous change in requirements.
* Lack of user involvement.
* Lack of executive support.
* Unrealistic expectations.
* Improper planning.

**Q.11 List the Challenges faced by BA in a project**

The challenges faced by a BA are as below -

* Obtaining sign off on requirements.
* Change management-with respect to cost and timelines.
* Coordination between developers and testers.
* Driving client to UAT completion.
* People management-coordinating with different people and different teams.

**Q.12 Write about Document naming standards**

All documents will be named using some standards like -

[project ID][document type]V[x]D[y].ext

Example:-XY123BRDV1D2.docx

**Q.13 What are the Do’s and Don’ts of a business analyst**

The Do’s and Don’ts of a business analyst are as below

* Never say NO to client.
* There is no word as “by default”.
* Never imagine anything in terms of GUI.
* Never question the existence of the existence.
* Consult an SME for clarifications in requirements.
* Every problem of client is unique. No two problems are the same. The approach or technology may be different.
* Go to client with a plain mind with no assumptions. Listen carefully and completely until client is done and then you can ask queries.
* Do not interrupt the client when he is giving you the problem.
* Try to extract maximum leads to solution from the client himself.
* Never try to give solutions to client straight away with your previous experience and assumptions.
* Concentrate on the importantly required requirements.
* Don’t be washed away or don’t imagine solutions on screen basis.

**Q.14 Write the Difference between packages and sub systems**

**Package** - It is a collection of components which are not reusable in nature. Application development companies work on packages.

**Sub-system** - It is a collection of components which are usable in nature. Product development companies work on sub-systems.

**Q.15 what is camel casing and explain where is it used?**

Here the entire first word is in lowercase and subsequent words first letter should be in upper case. There is no gap between words.

Example - turnRightAndThenTurnLeft

Camel casing is often used as a naming convention in computer programming to name different files and functions without violating the naming laws.

**Q.16 Explain Development server and what accesses a business analyst has?**

 A development server is a type of server that is designed to facilitate the development and testing of programs, websites, software or application for software programmers. It provides a run-time environment, as well as all hardware/software utilities that are essential to program debugging and development. Here the BA will have access to the public documents area and code and test areas.

**Q.17 what is data mapping?**

Data mapping is the process of matching fields from multiple datasets into a schema, or centralized database. Data mapping is required to migrate data, ingest, and process data and manage data. 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.

## Steps of data mapping: -

* **Define —** Define the data to be moved, including the tables, the fields within each table, and the format of the field after it's moved. For data integrations, the frequency of data transfer is also defined.
* **Map the Data —** Match source fields to destination fields.
* **Transformation** — if a field requires transformation, the transformation formula or rule is coded.
* **Test** — using a test system and sample data from the source, run the transfer to see how it works and make adjustments as necessary.
* **Deploy** — once it's determined that the data transformation is working as planned, schedule a migration or integration go-live event.
* **Maintain and Update** — for ongoing data integration, the data map is a living entity that will require updates and changes as new data sources are added, as data sources change, or as requirements at the destination change.

**Q.18 What is API?**

Application programming Interface

API is a software intermediary that allows two applications to talk to each other. It delivers a user response to a system and sends system’s response back to the user.