**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**

**Ans -**

****

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

**Ans -**

Boundary Class:

1. Payment to option boundary
2. Card Payment Boundary
3. Wallet payment
4. Cash payment
5. Net banking payment

Controller Class:

1. Payment initiated
2. Card payment
3. Wallet payment
4. Cash payment
5. Net banking

Entity class:

1. Customer
2. Payment
3. Card
4. wallet

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

**Ans -**

Use Layer:

1. Payment method selection
2. Card payment
3. Wallet payment
4. Cash payment
5. Net banking payment

Business Logic:

1. Payment initiated
2. Card payment
3. Wallet payment
4. Cash payment
5. Net banking

Data Layer:

1. Customer
2. Payment
3. Card
4. wallet

**Q-4. Explain Domain Model for Customer making payment through Net Banking.**

**Ans -**

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

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

**Ans -**

**Q6. Explain Conceptual Model for this Case.**

**Ans -**

Answer: A conceptual model is similar to domain model. It process payment done by customer using net banking and relation involved in payment transaction.

I help to visualize flow of process.

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

Answer: To identify classes from use case diagrams we can apply MVC rules.

1. Model:

* The model class knows about all the data that need to be displayed.
* It is model who is aware about all the operations that can be applied to transform that class.
* It only represents the data of an application.
* The model represents enterprise data and business rules that govern access to and updates this data.
* All model classes are represented as entity classes.

2. View:

* The view represents the presentation of the application.
* The view class refers to the model.
* It uses query methods of the model to obtain the contents and renders it.
* The view is not dependent on the application logic.
* It remains same if there is any modification in the business logic.
* View class is the data required by query.
* View class is represented as boundary class or form class.

3. Controller:

* When user sends a request for something then it always go through controller.
* The controller is responsible for intercepting the requests from view and passes it to the model for appropriate action.
* After action has been taken on the data, the controller is responsible for directing the appropriate view to the user.
* In GUIs, the views and controllers often work very close together.
* Controller class is working based on user’s command.
* It understands command given by user through boundary class.

MVC Architecture Rules:

1. Combination of one actor and use case results in one boundary class.
2. Combination of two actors and a use case results in two boundary classes.
3. Combination of three actors and a use case result in three boundary classes.
4. Use case will result in a controller class.
5. Each actor will result in one entity class.

MVC Classes in 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)**

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

Answer: Conflict management is one of the competencies of BA. Conflict management is process of addressing disagreements and finding mutually acceptable solutions. It aimed at reducing negative aspects of conflict while increasing positive ones.

Conflict management steps:

1. Identify conflict

2. Discuss details

3. Agree with root problem

4. Check for every possible solution to avoid future conflicts

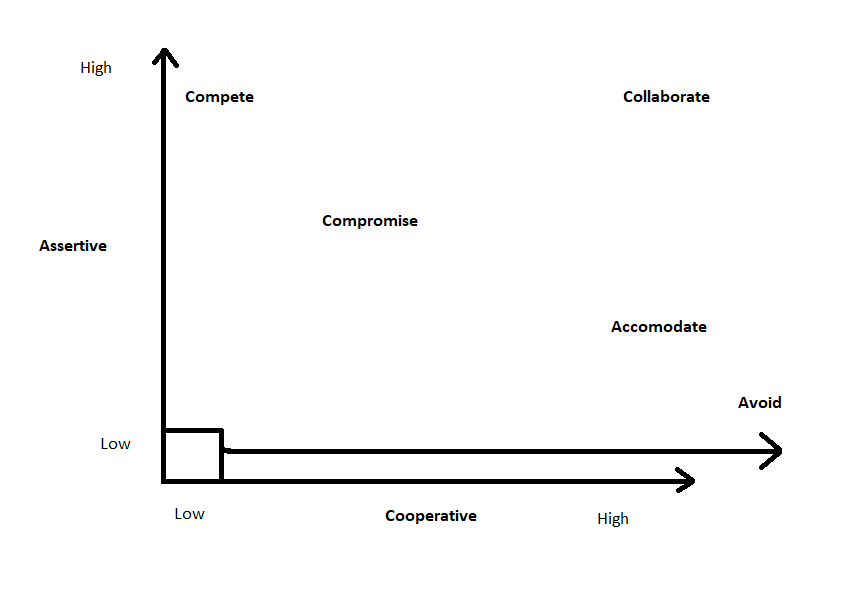
**Strategies of conflict management:**

1. Collaboration: In collaboration all parties work together to find solution.

2. Compromise: In compromise both parties give up something to reach agreement.

3. Accommodate: In accommodation, one party tries to fulfill other party needs by sacrificing their opinion.

4. Compete: In compete, we have to consider opinion of a single party as a final.



**Q10. List down the reasons for project failure**

Answer: Project can get failed by many reasons. Reasons are as follows:

1. Improper requirement gathering
2. Continuous change in requirements
3. Lack of user involvement
4. Lack of executive support
5. Unrealistic support
6. Improper planning

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

Answer:

1. Lack of Training
2. Obtaining sign-off on requirements
3. Co ordination between developers and testers
4. Conducting meetings
5. Making sure status reporting is effective
6. Dividing clients for UAT completion
7. People management (coordinating with different people and teams)
8. Overall making sure project health is good and having no issues.

**Q12. Write about Document Naming Standards**

Answer: All documents are name includes project id, document name, version no, document no with file extension.

Format to name documents is:

[Project ID][Document Type]V[x]D[y].ext

Example: PQ563FRDV1D5.docx

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

Answer: Do’s and Don’ts of BA are as follows:

1. Never say No to the client
2. There is no word as ‘By Default’
3. Never imagine in terms of GUI
4. Question existence of existence
5. Consult SME for clarification in requirements
6. Every problem of client is unique. No two problems are same. Every problem has a unique solution.
7. Go to client with plain mind with no assumptions. Listen carefully and completely until he is done and then ask your queries.
8. Never interrupt client while he is speaking.
9. Never try to give straight way solutions based on previous experience and your assumptions.
10. Try to concentrate on important and true requirements.
11. Don’t get washed away by add on functionalities or don’t imagine solutions in screen basis.

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

Answer:

|  |  |  |
| --- | --- | --- |
| Sr No | Packages | Subsystems |
| 1 | A package can be termed as container which tends to organize group of elements present in system into more manageable unit. | A subsystem is components which represents individual behavior unit in system. |
| 2 | Packages are collection of components which are not reusable in nature. | Subsystems are collection of components which are reusable in nature. |
| 3 | Application development companies work on packages. | Product based companies work on subsystems. |
| 4 | Package is represented as rectangle with tab in upper left corner contains name of package and icon. | Subsystem is displayed as rectangle that contains name of subsystem and icon along with <<subsystem>> keyword. |
| 5 | Packages are smaller and more focused in scope. | Subsystems are larger and encompass multiple package and modules. |
| 6 | They manage dependencies at class and component level. | They manage dependencies at higher level, define boundaries and interfaces between different parts of system. |

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

Answer: Camel case is way of writing where initial word is small alphabet and from second word onwards first alphabet is capital and rest are small. Camel casing are used to write method names. Method names are represented as camel casing.

One object sends message to perform an operation to other object and receiving object performs an operation. These messages are sent by methods and method names are represented as camel casing.

Ex. readAndWrite();

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

Answer: A development servers refers to dedicated environment or server that is used during software development process. It provides a platform for developers and testers to build, test and debug applications before they are deployed to a production environment. The development server typically replicates target production environment to ensure compatibility and accurate testing.

Read only access

Collaborative access

Limited configuration access

**Q17. What is Data Mapping?**

Answer: Data mapping is process of establishing a relationship between data elements in two or more different data sources. It involves defining how data from one source and transform data in another source. It is mostly used in data integration, data migration and data transformation.

Main purpose of data mapping is to ensure data can be transferred, converted. It involves identifying source data elements determine their meaning and structure.

**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**