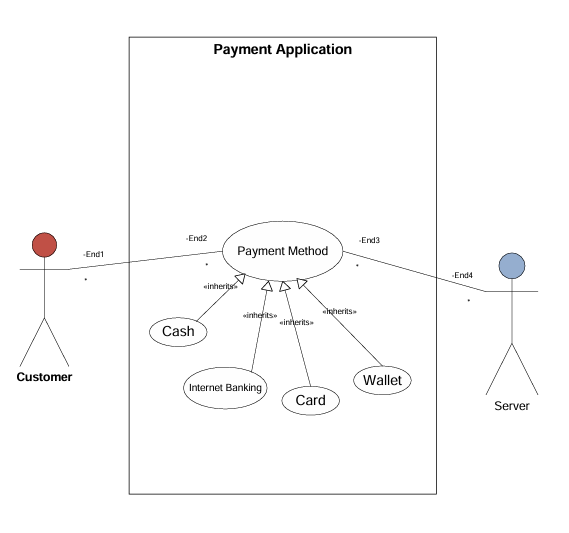
Q1. Draw a Use Case Diagram

**Ans-**



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

**Ans-**

1. **Boundary Class**- CustPayment
2. **Controller Class-** Making Payment
3. **Entity Classes**-

* Primary Actor- Customer
* Secondary Actor- Admin

Q3. Place these classes on a three tier Architecture.

**Ans-**

****

CustPayment

Making Payment

Application Layer

Business Logic Layer

Customer EC

Admin EC

Data Layer

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

**Ans-**

* Domain Modelling is also known as Conceptual Modelling.
* A Conceptual Model depicts the concepts (idea, thing, object) that are easily identifiable in the problem description.
* In above example, Customer is making payment through Net Banking.
* Here, Customer is primary actor and Admin is secondary actor in Entity class.
* Payment through Net banking is Controller class and CustPayment is Boundary class.



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

**Ans-**

****

CustPayment BC

Customer EC

CustPayment BC

Making Payment CC

VerifyPayment()

Make Payment

Status (return value)

PaymentSuccess()

Validation()

Q6. Explain Conceptual Model for this Case.

**Ans-**

* Domain Modelling is also known as Conceptual Modelling.
* A Conceptual Model depicts the concepts (idea, thing, object) that are easily identifiable in the problem description.
* In above example, Customer can make payment through card, cash, wallet or net banking.
* Here, Customer is primary actor and Admin is secondary actor in Entity class.
* Making Payment is Controller class and CustPayment is Boundary class.
* While making payment customer can choose wallet, cash, card or net banking option.
* Once payment is done customer will get verification mail for the same.

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

**Ans-**

To identify Classes from use case Diagram, we apply MVC rules on each use case to derive Classes

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 the business rules that govern access to and updates of this data. This represents Database (Tables in DB). All Model Classes are represented as Entity Classes.

Entity Classes-

1. **View:** The view represents the presentation of the application. The view class refers to the model. It uses the 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. ViewClass is the data required by the query. View Class is represented as Boundary Class or Form Class. Actor speaks to system and vice-versa through boundary Authenticating information boundary and Ent it y class Between boundary and Entity class

Boundary Class-

1. **Controller:** Whenever the user sends a request for something then it always go through the controller. The controller is responsible for intercepting the requests 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. In GUls, the views and the controllers often work very closely together.

 Controller Class or Transient Class (Given to Front end designers)

Controller class is working based on the user's command. Understands the command / request given by user through boundary/ Form Class

### MVC Architecture Rules

* 1. Combination of One Actor and a 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 s in Three Boundary classes and so on.

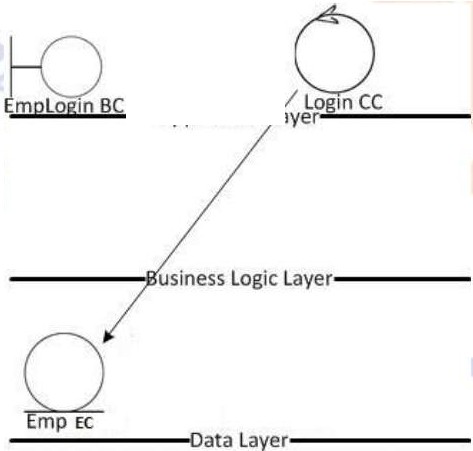
Note: only one primary actor is to be considered with a use case.

* 1. Use case will result in a controller class
  2. Each Actor will result in one entity class

1. **Guidelines to place identified MVC Classes in a 3 Tier Architecture**
2. Place all Entity Classes in DB Layer
3. Place Primary Actor associated Boundary Class in Application Layer
4. Place Controller Class in Application Layer
5. 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.
6. Let us place the Discovered Classes into this 3 tier Architecture

Transient Classes - Boundary Classes and Controller Classes

Persistence Classes - Entity Classes



Application

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

**Ans-**

|  |  |  |  |
| --- | --- | --- | --- |
| Stages | | Activities | Artifacts & Resources |
| Pre project |  | Enterprise Analysis — SWOT Analysis, GAP Analysis, Market Research, Feasibility Study, Root Cause Analysis, Decision Analysis, Strategy Analysis, Enterprise Architectural Frameworks, Project Scope and Business case writing, Risk analysis | Business Case  SOW (Statement of Work) PO (Purchase Order) |
| Sr. BA, Business Architects Presales Consultants |
| Planning & Estimations & Assessment  Project Kick Off (Big Picture  Plan) | | 1. Understand Assumptions and Constraints along with Business Rules and Business Goals 2. Plan Packages for Big Projects 3. Understands the project plan from PM 4. BA conducts stakeholders Analysis 5. Plan BA approach strategy (Req. gathering techniques, communication, Req. mgmt., Documents to follow, Tools to use, Change Request Handling methodology) for this Project |  |
| PM  Sr. BA |
| Requirements Gathering | | 1. Stakeholders identify and document 2. Client gives BRD or BA prepares BRD by interacting with Client — Brainstorming, Document Analysis, Reverse engineering, Interviews, workshops, Focus Groups, Observation, Questionnaires. 3. Prototyping can be used by BA to make the Client to give more specific requirements 4. Sort the gathered Requirements (avoiding duplicate Reqs, grouping into similar functionality or into modules) 5. Prioritize requirements — MoSCoW | BRD (Business Requirements Document) |
| BA |

|  |  |  |
| --- | --- | --- |
|  | 6. Validate Requirements - FURPS | PM |
| Requirements Analysis | 1. Draws UML Diagrams(Use case and Activity Diagrams) 2. Prepares Functional Requirements from Business Requirements 3. All Architects comes up with Technical Requirements (SSD) 4. SRS will have Functional Requirements and Technical Requirements 5. Takes Signoff on SRS from Client. SRS is the first legal binding Doc between the Business and the technical Team 6. BA prepared RTM from SRS before Design phase starts. (BA is the owner of RTM). 7. BA traces how requirements are dealt in each phase of development life cycle from Design till UAT | Functional Requirements Specification  SSD (Supplementary Support Document)  SRS (Software Requirements Specification)  RTM (Requirements Traceability Matrix) |
| BA  PM  Solution-Architect DB —Architect  NW —Architect |
| Design | 1. From Use caseDiagram, Test Manager or BA will prepare Test Cases 2. Communicates with Client on the design and Solution documents (updates Status to Client and make them understand how the solution would look like to prepare them to drive UAT) 3. BA will initiate the preparation of End user manuals 4. updates RTM 5. From Use case Diagram Solution-Architect recommends Architecture of the IT solution 6. DB Architect uses Persistence Classes (Entity Classes) and comes up with ER Diagrams or DB Schema. 7. GUI Designer will look into Transient Classes (Boundary Classes) and designs all possible Screens for the IT Solution | Solution Document  Design Document— HDD— ADD |
| BA  PM  Solution-Architect DB — Architect  NW —Architect GUI - Designer  Test Manger |
| Coding | 1.BA organizes „IAD Sessions   1. BA clarifies queries of Technical Team during   Coding   1. Developers refer Diagrams and Transient   (Controller Classes) of BA and code their unit   1. Update End user manuals 2. Update RTM 3. Conducts regular Status meetings with technical team and the Client and tuning Client for participation in UAT | LDD —CDD Application |
| Development Team BA  PM |
| Testing | 1.BA- Prepares Test Cases from Use Cases or assists Test Manager to do so   1. BA performs high level testing 2. BA prepares Client for UAT 3. Test Data is requested by BA from Client | Test Concerning Documents Application with less errors |
|  |

|  |  |  |
| --- | --- | --- |
|  | 1. Updates End User Manuals | Testing Team |
|  | 1. Updates RTM | BA |
|  | 1. Take signoff from Client on Client Project | PM |
|  | Acceptance form | Client |
| Deployment and | 1.Forwards RTM to Client or the PM which should be |  |
|  | attached to the Project Closure Document |  |
|  | 1. Coordinates to complete and share End User |  |
|  | Manuals |  |
|  | 1. Plans and Organizes Training Sessions for End Users |  |
|  | 1. Prepares Lessons learned from this project (to take precautions for coming projects) |  |

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

**Ans-**

1. Conflict management is a vital skill that involves handling confrontations tactfully and constructively. Your aim is to yield a positive result from disputes and disagreements that occur between people in the workplace. Are results of Changes Improves in Communication, personal and professional growth.
2. Thomas —kilmann Technique

5 options of Conflict Management-Competing, Avoiding, Accommodating, Collaborating, Comprising

* Managing Conflicts

1. Conflicts require high level of energy for resolution
2. Managing Conflicts is all about maintaining the relations
3. Habits a e limiting factors in managing the conflicts
4. Before engaging in Conflicts, think about expected Outcome

* 5 Steps to 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 the future Conflicts

Question 10. List down the reasons for project failure

**Ans-**

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

Question 11. List the Challenges faced in projects for BA

**Ans-**

1. Lack of training
2. Obtaining sign-off on requirements
3. Change management with respect to cost and timelines
4. Coordination between developers and testers
5. Conducting meetings
6. Making sure status reporting is effective
7. Driving clients for UAT completion
8. People management (coordinating with different people and different teams)
9. Overall making sure project health is in good shape and delivered as per the timelines without any issues.

Question 12. Write about Document Naming Standards

**Ans-**

1. Naming documents in a standardized, logical and intuitive way ensures that team members and collaborators can discover, manage and access University records when needed.

## Why use naming conventions?

* Improved retrieval of documents on shared drives and University systems
* Facilitated disposal of documents when no longer required for business
* Ensured current or active version of a templates can be easily identified
* Supported sharing of information within your team and with collaborators
* Easier and more efficient file naming for colleagues as they don't have to 're-think' the process each time.

1. Documents Naming Standards

All Documents will be named using some standards like [ProjectlD][DocumentType]V[x]D[y]. ext

Example: PQ786BRDV1D2.docx

PQ786BRD1.2.docx

### Establish good foundations

* Keep file names short but meaningful
* Include any unique identifiers, e.g. case number, project title
* Be consistent
* Indicate version number where appropriate
* Ensure the purpose of the document is quickly and easily identifiable

Question 13. what are the Do’s and Don’ts of a Business analyst.

**Ans-**

1. Never say NO to Client
2. There is NO word called as "BY DEFAULT"
3. Never imagine anything in terms of GUI
4. Question the existence of existence / question everything in the world ex: what client gives is not always correct
5. Consult an SME for Clarifications in Requirements
6. Every Problem of Client is unique. No two problems of different Client are same. May be the approach, technology, place of use, local laws may be varied to make them (Problems) to be different.
7. Go to Client with a plain mind with no assumptions. Listen carefully and completely until Client is done and then you can ask your Queries.
8. Please do not interrupt the Client, when he/ She is giving you the problem. Maximum
9. Try to extract the leads to Solution from the Client itself.
10. Never try to give Solutions to Client straight away with your previous experience and assumptions.
11. Try to concentrate on the important and truly required Requirements.
12. Don’t be washed away by add on Functionalities or don’t imagine solutions on Screen basis.

Question 14. Write the difference between packages and sub-systems

**Ans-**

|  |  |
| --- | --- |
| **Packages** | **Sub-systems** |
| 1. Collection of Components which are not reusable in nature are called as packages | * 1. Collection of Components which are reusable in nature are called as sub-systems |
| 1. Development companies works on packages | * 1. Product Development Companies work on Subsystems and Application |

Question 15. What is camel-casing and explain where it will be used

**Ans-**

1. camelCase is a naming convention in which the first letter of each word in a compound word is capitalized, except for the first word. Software developers often use camelCase when writing source code.
2. camelCase is useful in programming since element names cannot contain spaces. The camelCase naming convention makes compound names more readable. For example, myOneMethod is easier to read than myonemethod.
3. Other examples of camelCase are:

* newString;
* getNewString()
* myVariableName;

1. The name camelCase (also "camel case" or "dromedary case") comes from the hump on a camel, which is represented by the capital letter in the middle of the compound word. A camelCase word may have one or more capital letters.

Question 16. Illustrate Development server and what are the accesses does business analyst has?

**Ans-**

1. Once a Project is initiated, then some space for documentation and some for coding will be allotted by the Network Engineer.
2. If a Resource is selected in the project, then access to these spaces will be given to that resource after approval from Project Manager.

Question 17. what is Data Mapping.

**Ans-**

1. The process of matching fields from multiple datasets into a schema, or centralized database, is known as data mapping. To transfer, ingest, process, and manage data, data mapping is required. Data mapping’s ultimate purpose is to combine multiple data sets into a single one.
2. Different data sets with different ways of defining similar points can be joined in a way that makes them accurate and useable at the ultimate destination, which is known as data mapping.
3. Data mapping is a common activity in the business world. However, as the amount of data and the complexity of the systems that use it has grown, the data mapping process has become more involved, necessitating the use of automated and powerful technologies.

Question 18. 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.

**Ans-**

* An application programming interface (API) is a messenger that processes request and ensures seamless functioning of enterprise systems. API enables interaction between data, applications, and devices. It delivers data and facilitates connectivity between devices and programs.
* API can also be defined as an online programming interface of the organization. It allows applications to communicate with backend systems.
* An application programming interface creates a channel for the company to sell its products and services online. API enables access of services by adding codes to applications. It further enhances connectivity and bolsters functionality.
* API connectivity helps applications share data and communicate with each other without human interruption. You enable communication between two web tools or applications through their APIs. It allows organisations to automate systems, enhance the seamless sharing of data, and integrate current applications.
* API unlocks a new level of flexibility of information and service delivery. It also makes the embedding of content from different sites and apps easy. An API acts as the interface that permits the integration of two applications.

That’s how date format in two different formats will also handle by APIs effectively.