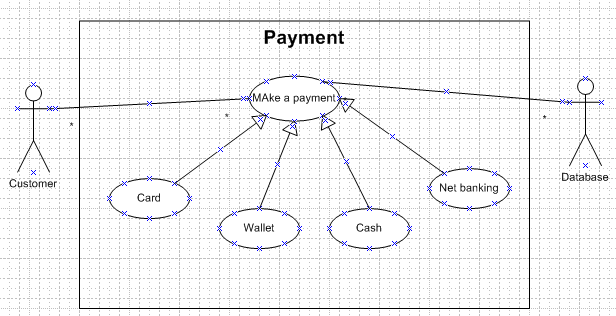
Prep Exam 3 –Part ½

Case Study 1 (Q1-Q6 ◊ 24 Marks) 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

Answer)



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

Answer)

1. **Model Class (Entity class)** : 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 class, model class or persistent class

1. **View/Boundary class:** The boundary class represents the presentation of the application. The boundary class refers to the model class for any query and then renders it to the user. 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.

Boundary class, View class or Form class

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

Controller class or transient 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 results in Three Boundary classes and so on....

4.Use case will result in a controller class

5.Each Actor will result in one entity class

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

Employee entity class

Make payment controller class

Employee login boundary class

Database layer

Business logic layer

Application layer

**Application layer:** GUI’s, screens, pages, Validations on pages, Organization specific business logic will be on the Application Layer

**Business logic layer:** All reusable components (logic pertaining to industry), Frequently changing Components, Governing Body rules and regulations, Compliances should go to middle layer Ex: Printer, Payment Gateways, mail Servers, RBI rules for banks, IRDA rules for Insurance, etc.,

**Database layer:-** Data Base Components connecting to databases will be at the Data Layer.

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

Answer) Domain model is a structured visual representation of interconnected concepts or real-world objects that incorporates vocabulary, key concepts, behavior, and relationships of all its entities.

\*) When building an application, the first step is always like a whiteboard.

\*)A domain model helps us to understand the real-world visually. In it we will see in an illustrated way:

* Define entities or conceptual classes.
* Associations between entities.

Domain model is like 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.

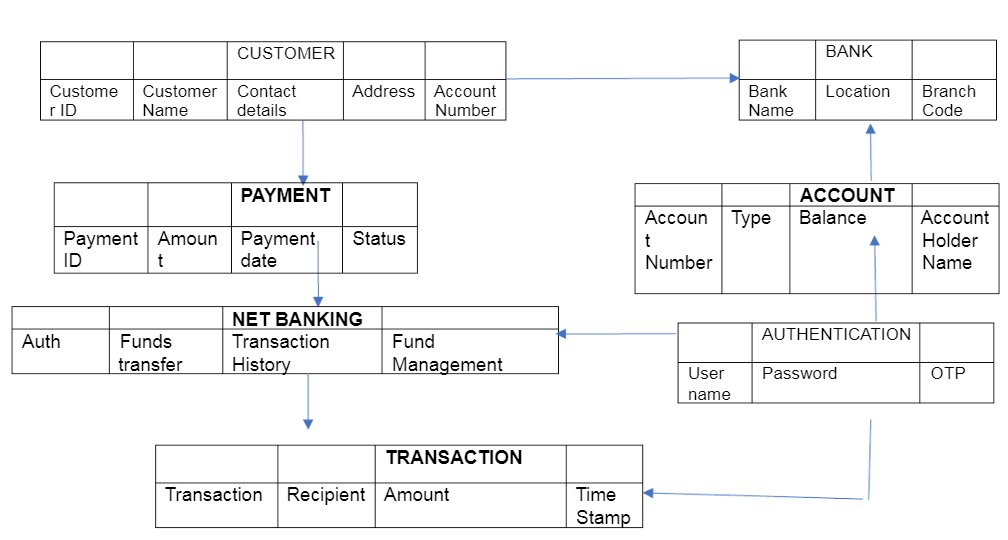
Customer table is also connected to payment table, because he should make the payment.

Now the payment is done by net banking, so payment table is connected to net banking table.

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

The authentication table is connected to both net banking table and bank table, because authentication is to be performed there.

Also, the authentication table is connected to transaction table, because authentication will be done while transaction



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

Answer) The sequence diagram is used primarily to show the interactions between classes in the sequential order in which those interactions occur. A sequence diagram can map a scenario described by a use case in step-by-step detail to define how classes collaborate to achieve your application's goals.

UML sequence diagrams are used to represent or model the flow of messages, events and actions between the classes or components of a system. Time is represented in the vertical direction showing the sequence of interactions of the header elements, which are displayed horizontally at the top of the diagram.

Sequence Diagrams are used primarily to design, document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task or scenario. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications.

One of the primary uses of sequence diagrams is in the transition from requirements expressed as use cases to the next and more formal level of refinement. Use cases are often refined into one or more sequence diagrams. In addition to their use in designing new systems, sequence diagrams can be used to document how classes in an existing ("legacy") system currently interact. This documentation is very useful when transitioning a system to another person or organization.

Customer

Bank

Netbanking

Login page

Checkout page

Customer

Cx login in to the application

Adds products and proceeds to checkout

Selects NB as payment option

Authentication done with the bank

Transaction successfule page

displayed to the customer

Q6. Explain Conceptual Model for this Case - 4 Marks

Answer) The conceptual model helps in understanding the key concepts, their relationships, and the overall structure of the net banking payment system.

A conceptual modelling is a process of developing an abstract or a graphical representation of real-world objects or entities.

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 provides such good services that it will

help to enhance the customer’s 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 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

Customer

Customer

Customer

Service

Net Banking

Customer

Data Privacy

Customer

Technology

Customer

Trust & support

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

**Model–view–controller** (**MVC**) is a [software design pattern](https://en.wikipedia.org/wiki/Software_design_pattern) commonly used for developing [user interfaces](https://en.wikipedia.org/wiki/User_interface) that divides the related program logic into three interconnected elements. These elements are:

1. **Model Class (Entity class)** : 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 class, model class or persistent class

1. **View/Boundary class:** The boundary class represents the presentation of the application. The boundary class refers to the model class for any query and then renders it to the user. 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.

Boundary class, View class or Form class

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

Controller class or transient 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 results in Three Boundary classes and so on....

4.Use case will result in a controller class

5.Each Actor will result in one entity class

**Guidelines to place classes in 3-tier architecture-**

1. **Application layer:** All GUI, screens, pages, validation on the pages, organization specific business logic will be on the application layer.
2. **Business Logic layer:** Frequently changing components or reusable components, government rules and regulations, compliance will be placed in business logic layer. Eg Payment gateway, printer, mail server, government body rules and regulations.
3. **Database layer:** Data components connected to the database will be shown in database layer.

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

This is the most common and classic of life cycle models, also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, 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 or not to continue or discard the project

* 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.
* Once that particular phase has been completed and once, we move to the next phase, we cannot go back to the previous phase to make changes

**BA contributons in waterfall model:-**

**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.

**In this phase BA will perform requirement gathering using elicitation technique and then prepare a BRD.**

**Requirements Analysis:-**The requirements are analyzed to understand the scope of the project. Analyzing means the BA will check all the requirements, if he finds that the requirements are proper 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].
* BA will also prepare non functional requirements . 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’, 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.
* **BA prepares UML diagrams and helps the designers, architects and developers to translate these requirements.**

**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 meeting

**BA will conduct JAD sessions to ensure that the requirement that the stakeholders have are understood well by the developers.**

**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.
* BA prepares end user manuals that can be used for the end users.
* BA updates RTM that can be attached to the project completion document for getting a sign off.

**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.

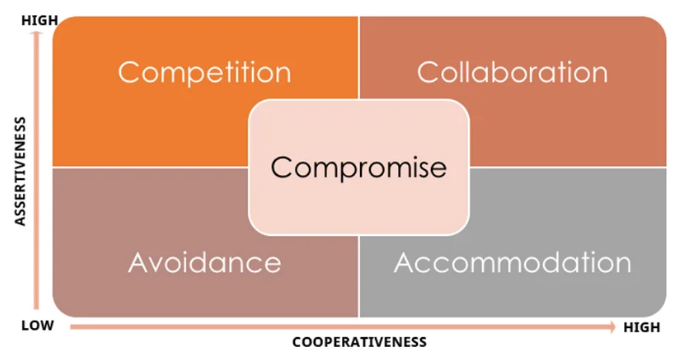
BA will Update documentation and requirements specifications to reflect changes in the system over time.

**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)

Answer) Conflicts mean any differences that happen between the team members or stakeholders.

* 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. The model has two approaches, also known as “dimensions”: assertiveness and cooperation. Most of you are probably intimately familiar with each of these dimensions on their own, as well as the associated personality traits, but not necessarily how they interact. That is where this model shines. There are five forms of conflict resolution that use these two approaches to different degrees.



Q10. List down the reasons for project failure – 6 Marks

Answer) Reasons for project failure are:

1. **Improper requirement gathering-**If the requirements of the project are not gathered correctly, then this can lead to project failure.
2. **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.
3. **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.
4. **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.
5. **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.
6. **Lack of user involvement: -** Lack of executive support.
7. **Unrealistic expectations: -**means the goals that cannot be achieved or the goals that are out of scope.
8. **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.
9. **Insufficient resources: -** Insufficient resources can also lead to project failure. The project may fail due to lack of technology knowledge or lack of finance.
10. **Lack of training:** - If the involved resources are not trained properly it may lead to delays in the project and also may be a reason for project failure.
11. **Improper processes:-** If SOPs are not in place everyone will take steps as per their convience and hence the process has to be streamlined.

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 meetings.
* 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

* Document naming standards indicate a standard for writing document names.
* The objective is to ensure that the elements included in a file name are ordered in such a way as to facilitate the quick retrieval of the file during everyday business.
* The naming standards differ with different organizations, however below is an example of a commonly used naming standards:

**[ProjectID][Document Type]V[x]D[y].extension**

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 that is with no assumptions
* Listen to the client very carefully and after he is done, then ask questions
* 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 the mud :- Bring clarity to the project so the team can deliver perfect application
* Requirement hurried-project buried.
* Never criticize the stakeholder. Always appreciate the stakeholder even for small efforts
* Never try to give Solutions to Client straight away with your previous experience and assumptions.
* Try to concentrate on the important and truly required Requirements.
* Don't be washed away by add on Functionalities or don't imagine solutions on Screen basis.

Q14. Write the difference between packages and sub-systems – 4 Marks

Packages : A Package is a grouping and organizing element in which other elements reside, which must be uniquely named. In the UML, packages are used in a manner similar to the way directories and folders in an operating system group and organizees. For example, the project management system may be decomposed into a collection of classes organized into packages.

Sub-Systems :Recall that a system is an organized collection of elements that maybe recursively decomposed into smaller subsystems and eventually into non decomposable primitive elements. For example, the project management system may be decomposed into the following: A user interface subsystem responsible for providing a user interface through which users may interact with the system. A business processing subsystem responsible for implementing business functionality. A data subsystem responsible for implementing data storage functionality. While a package simply groups elements, a subsystem groups elements that together provide services such that other elements may access only those services and none of the elements themselves. A subsystem is shown as a package marked with the subsystem keyword.

Q15. What is camel-casing and explain where it will be used- 6 Marks

* Camel case is a naming convention for writing file or object names using compounded or joined words with at least of those words beginning in a capital letter.
* Camel case is used in programming language to name different files and functions without violating the naming laws of the underlying language.
* Camel case is also known as medial capitals and Pascal case. The term camel case is derived from its appearance, which can resemble a camel's back. It is used in many programming language that doesn't allow spaces in le names. Camel case enables the creation of names that are more unique and have more meaning for the developer
* Example:- firstName, lastName
* 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 validateCustomerDetails, calculateInterestRate,
* Business rules, which should be satisfied by the system use camel-casing.
* While documenting business process or work flows, 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 increases readability
* Camel casing is also used t represent how classes communicate with each other is sequence diagrams.

Q16. Illustrate Development server and what are the accesses does business analyst has? -6 marks

* 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.
* A Development Server is a type of server used for creating and testing software applications before they are deployed to production. It is part of a set of servers that also includes production servers, test servers, and other specialized servers.
* The accesses a BA has are-

1. Read Only- BA’s may be granted with the read only access to the development server.
2. 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 behaviour of the application.

1. Limited Access-Depending upon the project needs, the BA’s will be granted limited access to the

specific modules in the application.

1. 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

* **Data mapping** **is the process of matching fields from one database to another. It's the first step to facilitate data migration, data integration, and other data management tasks.**
* Before data can be [analyzed for business insights](https://www.talend.com/uk/resources/big-data-analytics/), 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](https://www.talend.com/uk/resources/what-is-data-modeling/), 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.
* 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 data results .
* 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.

Answer) An API, is Application Programming Interface, is a software-to-software interface. APIs provide a secure and standardized way for applications to work with each other and deliver the information or functionality requested without user intervention.

An API, or application programming interface, is a set of defined rules that enable different applications to communicate with each other. It acts as an intermediary layer that processes data transfers between systems, letting companies open their application data and functionality to external third-party developers, business partners and internal departments within their companies.

APIs are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols. For example, the weather bureau’s software system contains daily weather data. The weather app on your phone “talks” to this system via APIs and shows you daily weather updates on your phone.

In the context of APIs, the word Application refers to any software with a distinct function. Interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses.

API architecture is usually explained in terms of client and server. The application sending the request is called the client, and the application sending the response is called the server. So in the weather example, the bureau’s weather database is the server, and the mobile app is the client.

API allows sharing of only necessary information and keeps the internal system details hidden, which helps the system security.

