Q no. 1 use case diagram



Q.2- Boundary Classes, Controller Classes, Entity Classes

Boundary Classes-
The Boundary class is a class that is the boundary of the system and other system user ( which is actor in the use case diagram).

The following are the features of the boundary class :

* This class is easier to be changed than the entity and control class.
* The attribute of this class and screen layout are defined at the basic design.
* Ina class boundary, there are cases that the stereotype(<<boundary>>) is added.
* In a class diagram, there are cases that is shown by the following icon



Controller classes:

The following are the features of the controller class.

* This class has a few attribute.
* In a Class diagram, there are cases that the stereotype (<<control>>) is added.
* This class is called to achieve use cases in the Use case diagram.
* In a class diagram, there are cases that is shown by the following icon



Entity Classes:

The entity class is a class that has data.
The “E” of the ER diagram means “Entity” too, if you know the ER diagram, you easily understand.
The followings are the features of the Entity class.

* There are many cases that these objects of this class are perpetuated in the DB.
* The extraction of the class is like ER diagram.
* This class is related to the DOA (Data-oriented approach).
* The module cohesion of this class is high, and is not easy to be changed.
* In a class diagram, there are cases that the stereotype (<<entity>>) is added.
* In a class diagram, there are cases that are shown by the following icon.



QNo.3 3tier Architecture



Payment method selection boundary
Card payment boundary
Wallet payment boundary
Cash payment boundary
Net banking payment boundary



Payment controller
card payment controller
wallet payment controller
cash payment controller
Net banking payment controller



Customer(entity class)
Payment(entity class)
card(entity class)
wallet(entity class)
Bank account(entity class)

Q.No4- Domain model for customer making payment through net banking

A domain model is a structured visual representation of inter connected concepts. It incorporates key concepts, behavior and relationships of all its entities. A domain model also serves as a conceptual model of a specific domain capturing both behavior and data.

A domain model of customer making payment might look like below:



Q.No5- Sequence diagram for payment done by customer through net banking



Q.No6-Conceptual model of the case

The conceptual model for a customer making a payment using different methods: **card**, **wallet**, **cash**, and **net banking**.

1. **Card Payment:**
	* A customer initiates a payment using their debit or credit card.The card details of the customer such as card number etc are entered into an online payment gateway. The payment gateway connects with the bank and performs the actions according and If the transaction is approved, the payment is processed, and the customer receives a confirmation.
2. **Wallet Payment:**
	* A customer uses a wallet like Google Pa or phonepe to make a payment. The customer links their bank details to the wallet app.
	* During checkout, the customer selects the wallet as the payment method. The wallet securely transfers the payment details to the merchant or service provider.
	* The transaction is confirmed, and the customer receives a notification.
3. **Cash Payment:**
	* The customer pays in physical currency i.e. cash. This method is commonly used for in-person transactions at retail stores, restaurants, or other establishments.
	* The customer hands over the cash to the cashier, who provides a receipt or change if necessary.
4. **Net Banking (Online Banking) Payment:**
	* The customer accesses their bank account through online banking.
	* They log in using their credentials.They provide the recipient’s details and the payment amount.
	* The bank processes the transaction, deducts the amount from the customer’s account, and sends a confirmation.

Q.No7-MVC Architecture

The Model View Controller (MVC) is a well-known design pattern in the web development field. It is way to organize our code. It specifies that a program or application shall consist of data model, presentation information and control information. The MVC pattern needs all these components to be separated as different objects.

* Model: It represents the business layer of application. It is an object to carry the data that can be also contain the logic to update controller if data is changed.
* View: It represents the presentation layer of application. It is used to visualize the data that the model contains.
* Controller: It works on both the model and view. It is used to manage the flow of application, i.e. data flow in the model object and to update the view whenever data is changed.

Advantages of MVC Architecture

The advantages of MVC architecture are :

* MVC has the feature of scalability that in turn helps the growth of application.
* The components are easy to maintain because there is less dependency.
* A model can be reused by multiple views that provide reusability of code.
* The developers can work with the three layers (Model, View and Controller) simultaneously.
* Using MVC, the application becomes more understandable.
* Using MVC, each layer is maintained separately therefore we do not require to deal with massive code.
* The extending and testing of application is easier.

Three-tier architecture, which separates applications into three logical and physical computing tiers. Is the predominant software architecture for traditional client-server applications.



Q.No 8- BA contributions in Project

|  |  |  |
| --- | --- | --- |
| Stages | Activities | Artifacts & Resources |
| Pre project  | Enterprise Analysis – SWOT Analysis, GAP Analysis, Market Research, Feasibility Study, Root Cause Analy-sis, Decision Analysis, Strategy Analysis, Enterprise Ar-chitectural Frameworks, Project Scope and Business case writing, Risk analysis  | Business Case SOW (Statement of Work) PO (Purchase Order) Sr. BA, Business Architects Pre sales Consultants  |
| Planning & Esti-mations & As-sessment 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 tech-niques, communication, Req. mgmt, Documents to follow, Tools to use, Change Request Handling meth-odology )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, Re-verse 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 dupli-cate Reqs , grouping into similar functionality or into modules) 5. Prioritize requirements – MoSCoW 6. Validate Requirements - FURPS  | BRD (Business Requirements Document) .BA PM  |
| Requirements Analysis  | 1. Draws UML Diagrams ( Usecase and Activity Dia-grams) 2. Prepares Functional Requirements from Business Requirements 3. All Architects comes up with Technical Require-ments (SSD) 4. SRS will have Functional Requirements and Tech-nical Requirements 5. Takes Signoff on SRS from Client. SRS is the first le-gal binding Doc between the Business and the tech-nical 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 Speci-fication SSD(Supplementary Support Document) SRS (Software Requirements Specification) RTM (Requirements Traceability Matrix) BA PM Solution-Architect DB – Architect NW – Architect  |
| Design  | 1. From Usecase Diagram , Test Manager or BA will prepare Test Cases 2. Communicates with Client on the design and Solu-tion 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 recom-mends Architecture of the IT solution 6. DB Architect uses Persistence Classes (Entity Clas-ses) 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 JAD Sessions 2. BA clarifies queries of Technical Team during Cod-ing 3. Developers refer Diagrams and Transient (Control-ler Classes) of BA and code their unit 4. Update End user manuals 5. Update RTM 6. 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 2. BA performs high level testing 3. BA prepares Client for UAT 4. Test Data is requested by BA from Client 5. Updates End User Manuals 6. Updates RTM 7. Take signoff from Client on Client Project Ac-ceptance form  | Test Concerning Documents Application with less errors Testing Team BA PM Client  |
| Deployment and Implementation  | 1.Forwards RTM to Client or the PM which should be attached to the Project Closure Document 2. Coordinates to complete and share End User Man-uals 3. Plans and Organizes Training Sessions for End Users 4. Prepares Lessons learned from this project (to take precautions for coming projects)  |  |

Q.No 9- Conflict management using Thomas- Kilmann technique

# Conflict management using Thomas – Kilman technique

In the 1970s, researchers Kenneth Thomas and Raplh 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.

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.

Thomas-Kilmann’s Five Modes for Handling conflicts

From the correlation of these two and the scale of implementation, Thomas-Kilmann gave us the following five modes for handling the presented conflicts:

Competing

Competing, the first Thomas-Kilmann conflict mode is assertive and non-cooperative. It refers to addressing only one’s own concerns at the cost of the concerns of the other. It is a power-oriented mode- one uses whatever power dynamic seems appropriate to get a favourable outcome for oneself.
An individual’s ability to debate, their position in the hierarchy, or their financial power matters the most. Competing is defensive- it strictly means standing up for your individual beliefs and simply trying to win.

Accommodating

According to the Thomas-Kilmann model, the Accommodating mode is both accepting and cooperative. It is the opposite of completing. While accommodating, the individual in question neglects their own problems or beliefs to address the problems of the other party. The element of self sacrifice is highlighted in this mode. Accommodating typically involves selfless understanding, generosity or charity. At times, accommodating would require you to follow the other person’s orders when you would not like to do so, or submit to the other’s perspective or decisions.

Avoiding

In the Thomas-Kilmann model, avoiding is both unassertive and uncooperative. 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. It could also involve putting off the issue until the time is favorable, or simply stepping back from an uncomfortable or hazardous situation.

Collaborating

Collaborating, the most beneficial outcome in the Thomas-Kilmann conflict model 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. Collaborating involves deep-diving into an issue to locate the critical demands of the concerned individuals or parties. Collaborating between two or more people might take the form of a guest to understand the ‘why’ of the disagreement. It involves striving to look for creative answers to interpersonal issues and enriching yourself from the other person’s insights.

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. It addresses an issue more directly than avoiding but falls short of investigating it with as much depth and rigor as collaborating. In certain situations, compromising might involve seeking middle-ground solutions, providing concessions or looking for a quick solution that provides some way forward from the impasse.

The Thomas-Kilmann model is based on two dimensions: assertiveness and empathy. There are 5 conflict resolution strategies: Complete, Avoid, Accommodate, Collaborate and Compromise. Each strategy has its benefis and disadvantages. Choose the appropriate one according to the situation.

Q.No 10- Some reasons for project failure

* Poor planning
Although sometimes overlooked in importance, lack of planning can make a project fail.
Having a successful project depends on properly defining in detail the scope, the time frame, and each member’s role. This way, you’ll have a route laid out to follow.
* Inconsistently defined resources
Let’s be clear: planning shouldn’t be limited to agendas, meetings, and responsibilities. Itshould also include human, intellectual, financial or structural resources. If these are not consistently determined, deadlines can’t be met, which can jeopardize the project’s conclusion.
* Unclear objectives
Objectives should be clearly defined, so as time goes by, you’ll know if you’re doing what’s right or not. Remember that choosing measurable goals helps you better visualize your progress and helps you see how you are to achieving your results.
* Lack of detail control
Monitoring is essential for successful projects, even knowing the details of many projects simultaneously can be very challenging.
As a result, it’s important to know how your project is going; if it is on schedule and if the budget is under control. This way, if there are any divergences from the initial plan,you can still correct them.
* Lack of transparency
It’s essential that everyone involved in the projects have complete project visibility so that if it doesn’t fail- not only the project manager, but other team members too.
This includes clear communication, good document management, and transparency abouttaks status, all of which can be achieved with centralized, all-digital files.
* Lack of communication
Communication is the key to good project management. Without the right tools and processes to allow interaction among team members and the project manager from the beginning, efficient communication can seldom be achieved.
* Change of direction
Among the ways project fail, a very common one is scope creep. This concept refers to changes requested when the project has already started which had not been planned before. This is very common when projects are not appropriately documented and defined beforehand.
* Unrealistic expectations
When you want to do something fast, with a limited budget, and a reduced team, it can really make your project fail. You should be realistic when it comes to your teams’ capabilities, deadlines and the resources available only then can you obtain results you want.
* Lack of monitoring

Providing a schedule to the team is not enough for a project to be successful. You should also make sure everything goes as planned. This means having frequent progress checks or meetings as well as making adaptations, when necessary, is essential.

* Unrealistic due dates
Planning co unrealistic due dates tasks for short due dates is definitely one of the causes for project failure. It is vitally important to carefully consider how long each project phase will take, in addition to extra time for unexpected events. This is the only way to develop a quality project.
* Poorly assigned roles
When each team member receives their responsibilities clearly, they will know what, when and how to perform their activities without someone needing to constantly ask for it.

Q.No 11 -Challenges faced in projects for BA

A BA is responsible for multiple tasks at the same time. From handling the projects, maintaining client relationships, interacting with stakeholders, and managing project deadlines, Business Analysts gota lot on their plate. Some of the challenges faced by BA are

* Lack of domain knowledge
A Business Analyst needs to collaborate with the business users to understand the requirements. Domain Knowledge plays a vital role in having a clear and complete understanding of the requirements. It is challenging for Business Analysts to be assigned to a wide variety of projects as learning new domains needs time and energy.
Whenever a new project is assigned, sit with the responsible person and understand the project requirements. Take notes whenever necessary and understand them thoroughly. It is challenging to learn new domains sometimes, but you must make mistakes.
So go on a loop until we make every bit of our knowledge count on our finger tips. It will help while implementing and processing the outcome of the project.
* Lack of Up To Date Process
The success of a project does not happen overnight. First, much effort and mental exhaustion are poured in to bring results. Following this, the lion’ share is up-to-date process of maintaining and evolving the project. The biggest challenge is the lack of up-to-date techniques and documentation. In most cases, the Project Documentation is incomplete, which hampers productivity.
This maybe overcome by requesting a demo from SME or concerned staff member after conducting extensive testing.
* Changing Business Needs or requirements
Business stakeholders frequently request revisions to requirements even after they have been finalized and approved, as experienced by Business Analysts. It might happen more once, even for the exact requirement, making it one of the most frequent issues. These adjustments could have an impact on the Business Analysts effort as well as the project effort, cost and schedule.
A change in the implementation cycle might impact the delivery process even if there are approaches that, like agile, accept change. Business Analysts and other essential stakholders must therefore determine how the difference may be implemented in the best way.
* Inadequate stakeholder involvement
Lack of crucial stakeholders: If this occurs there will be multiple problems since they will not be up-to-date on discussions about the most recent requirements. Either they won’t be able to express their ideas, or they will subsequently propose revisions.
Stakeholder’s lack of Cooperation: Occasionally, you may encounter one or more stakeholders who are unwilling to cooperate. It could cause delays, sign-off problems and even approval problems.
* Unrealistic Timelines
 As a Business Analyst, you find yourself in a problematic situation where timlines might be the concern. In that case, pressure is created, which might hamper your work. In that case, understand how to tackle such a situation while maintaining the quality of the work.
As a BA you cannot change the terms of the agreement, but you can evaluate its effects and inform management of the probable costs and losses. Unrealistic Expectations from stakeholders are widespread. It’s crucial to manage these expectations balanced without permanently damaging the relationships.
* Technical skills
When it comes to Business Analysts, it’s a myth that they don’t require technical skills. On the contrary, most of them are champions in coding, know hot to maintain business processes, and have a knack for technically undertaking the requirements. Moreover, Business Analysts are involved in every step of the product development cycle; hence they must understand the technical and functional side of the business as well.
* Professionalism
Business analysts areone of the most underappreciated, underpaid, and ignored members of the IT world. They frequently serve as the binding agent between a project’s technical and business aspects. They are the one who contribute to the development of the project plan and whosupports the project from beginning to end. They will collaborate with developers to ensure the project is constructed following the most current standards and satisfies the business expectations.
* Managing Communication
When you communicate effectively you aid developers in understanding the needs, limits and requirements of the business. You contribute to the development of solutions that benefit the client as well as the company. You guarantee the work is completed on schedule and to the required standards. But communicating the point is difficult. It involves a variety of abilities and trade secrets.
* Conflict with users
Sometimes, you might find yourself in situation where you cannot understand the user’s complaint. It happens during the product release stage and might come as rude feedback. Even conflict between stakeholders and business analysts may arise when a team suggests new strategy pertinent to the existing business process.
* Mindset
Business analysts must be prepared to deal with various difficulties throughout their work, from limitations of the technologies they employ to push back from stakeholders nd other team members. But how one approaches their task can significantly alter if they are ready for the most typical obstacles.

Q.No12- Document naming standards

Establishing file naming conventions is essential for maintaining an organized and efficient system.
All documents will be named using some standards like
[ProjectID][Document type]V[X]D[Y].ext
some practices which can be considered while naming a document:

* Keeping a precise, short and meaningful name:
Strive for concise file names while ensuring they convey relevant information. Clarity is crucial.
For example considering “ProjectX\_MeetingNotes\_20240530.docx,” instead of “Meetingnotes\_Jan2024.docx.”
* Using consistent and relevant elements:
Incorporate consistent elements in your file names to make them machine friendly.
Common elements include project names, dates, version numbers or any keywords.
* Avoid special characters
Special characters can cause issues in different operating systems or software applications.
* Logical file organization
Arrange the files into directories(folder)following a consistent pattern.
Avoid repetition of semantic elements among file and directory names.

Q.No13- DO’s and Don’ts of Business Analyst

* Every problem of client has uniqueness, so talk to the client with a plain mind with no assumptions from your previous experience.
* Never come to any conclusion before listening or understanding all the aspect of requirement from client, if you have a slight amount of doubt about any demand or change it’s always preferable to clear it with the client, subject matter expert, or with your team member.
* You can take inputs from experienced people about any requirement and for that conducting meeting with them is not the only way, you can have coffee with them, have a walk, meet them sometime in between or end of the day; you can use you own creative ideas to interact with them.
* Listen very carefully and completely to the client as well as to the end user and then ask question, don’t interrupt them in between, sometimes the solution are itself hidden in the problem.
* Always remember to use best of your time it’s not always compulsory or beneficial to attend the entire meeting, try to prioritize them and always have a prior discussion with your project manager and sponsor before conducting a meeting.
* Maximum try extract solution from client itself.
* Don’t be washed away by add on functionalities; just make sure to prioritize them.
* Always try to build a repo with your senior, colleague and your team, take care not to break confidentially and earn their trust.
* Make sure that you have gathered all the requirements from the stakeholder for your project, missing out any information can results to unwanted redo the work as well as delay projects and increase cost.
* It’s better to ensure the activities of you analysis plans are synchronizing with the project manager schedule time to avoid any delay in project deliverables.
* Sometimes non functional requirements of clients are not feasible because of budget or time constraint, so it’s always better to liaison with your OM to find out what is out of scope so that all will be in the same page and avoid misunderstanding.
* In a project, PM aim is to minimize new requirement to add in project scope, so as a BA we need to understand this and help to minimize the scope creep.
* As a BA understand the root cause of the problem, to facilitate the solution don’t jump into the conclusion.
* Prepare you mind to work through challenging situation to negotiate and facilitate the project to complete on time.

Q.No14-Packages and Sub-systems

## 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 ina operating system group and organize files. For example, the project management sytem may be decomposed intoacollection of classes organizedinto packages as follows:

## Sub-systems:

Recall that a system is an organized collection of elements that may be recursively decomposed into smaller subsystems and eventually into non decomposable primitive elements. For example, the projectmanagement 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 sub system responsible for implementing dta 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 hsown as a package marked with the subsystem keyword.

Q.No15-Camel-casing

Camel-casing 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.

Camelcase is used in programming language to name different files and functions without violating the naming laws of the underlying language.

Camelcase is also known as medial capitals and Pascal case.

The term camelcase 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 file names. Camelcase enables the creation of names tha are more unique and have more meaning or the developer.

For example, files name Bigball, BigBall and bigBall can be read much more easily than bigball

CamelCase is a way to separate the words in a phrase by making the first letter of each word capitalized and not using spaces. It is commonly used in web URLs,programming and computer naming conventions.

Q.No16-Development server

A development server is a type of server that is designed to facilitate the development and testing ofprogams, 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 platforms tools and utilities, network access and a high-end processor. Upon testing completion, the application is moved either ta staging server or production/live servers.

Q.No17-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.
Data mapping bridges the differences between two systems, or data models, sothat when data is moved from a source, it is accurate and usable at the destination.

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

Data mapping is an essential part of many data management processes. If not properly mapped, data may become corrupted as it moves to its destination. Quality in dta mapping is key in getting the most out of your data in data migrations, intgrations, transformations and in populating a data warehouse.

Data mapping is an essential part of ensuring that in the process of moving data from a source to destination, data accuracy is maintained. Good data mapping ensures good data quality in the data warehouse.

Q.No18-API

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.