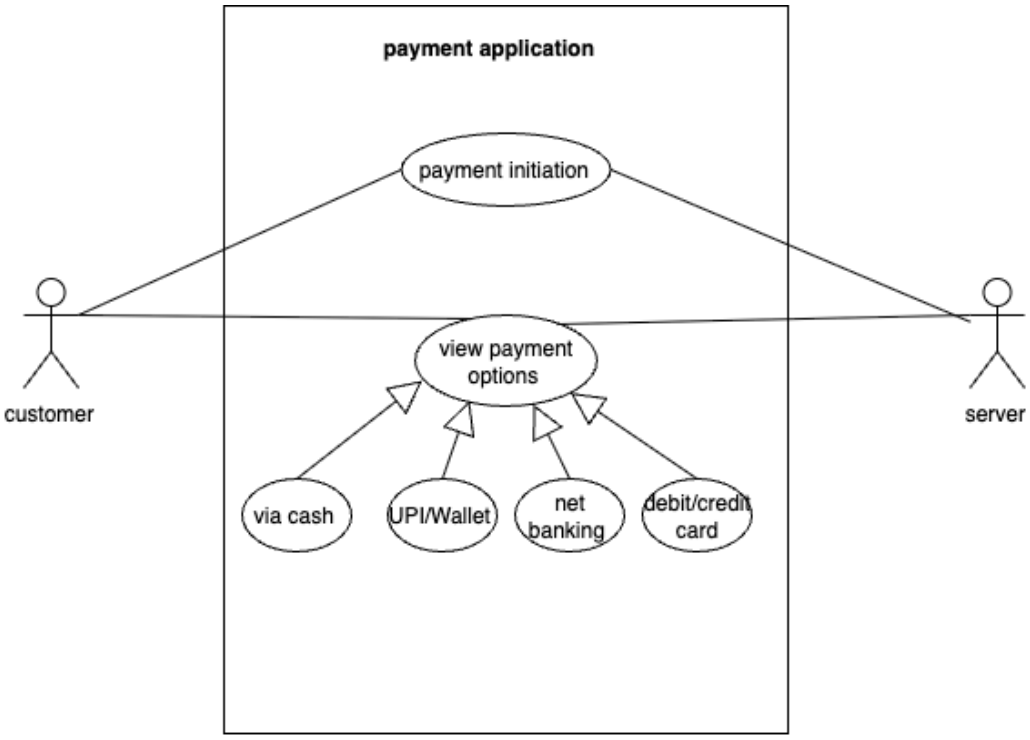
Capstone project 3 part 1

Q 1. Draw a Use Case Diagram.?



Q 2. Derive Boundary Classes, Controller classes, Entity Classes.?

Boundary class

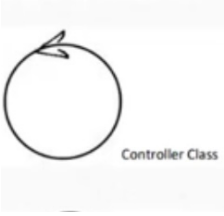
Boundary class used to handle interactions between the system and the external actors.

Example: login page, booking form

Controller class

Controller class acts as intermediaries between boundary class and entity class.

Example: login controller, order controller

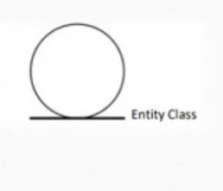


∫∫∫≤˜˜≤≤˜˜≤˜mn

Entity class

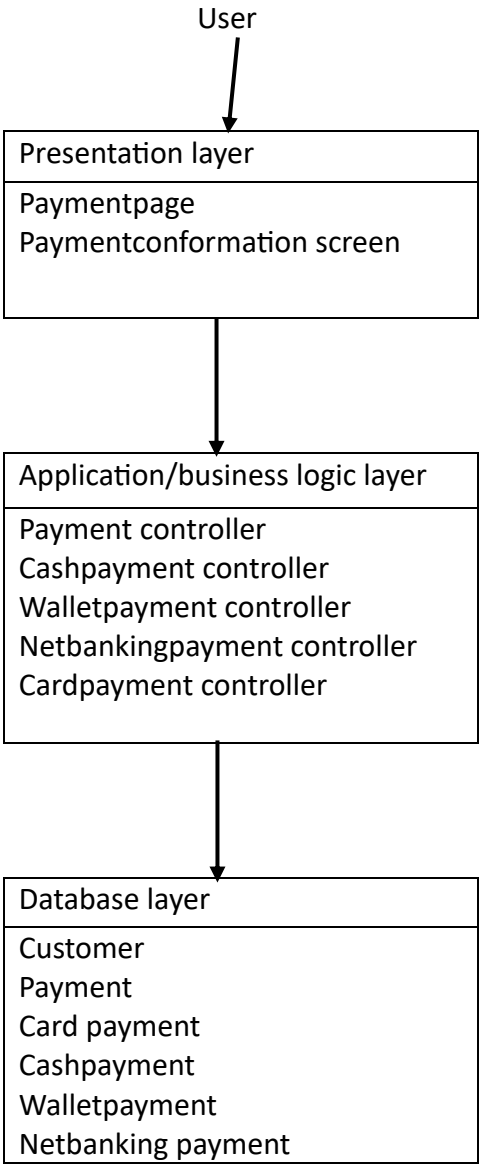
Entity class represents the core data and business logic of the application

Example: user, ticket, product



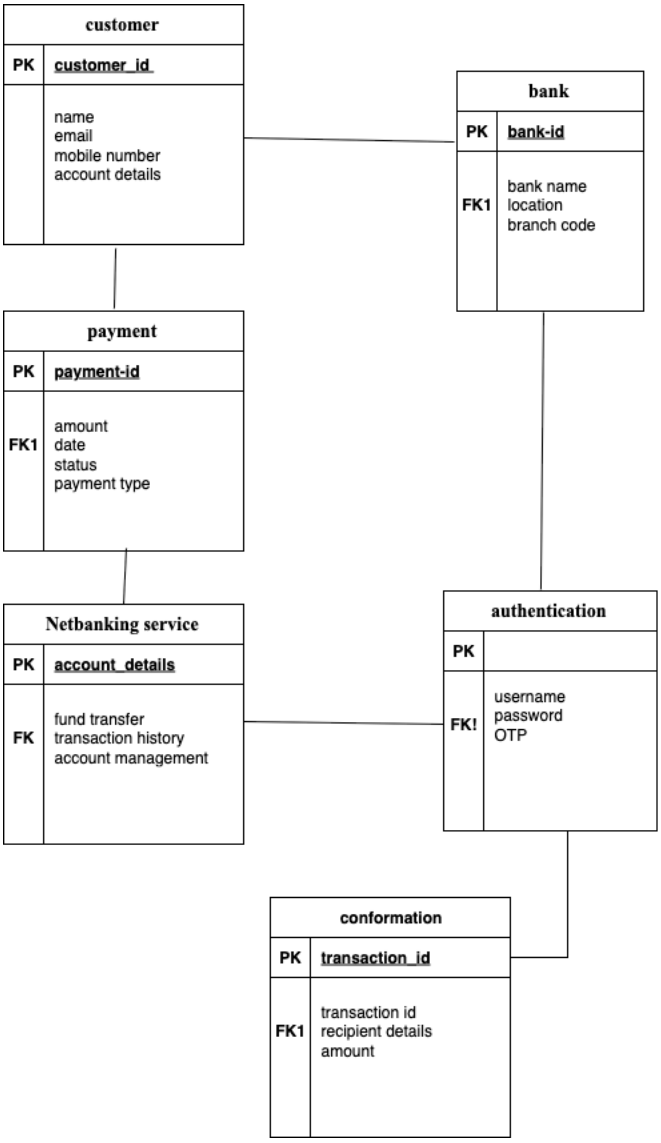
Q 3. Place these classes on a three tier Architecture.?

3-tier architecture is a software design pattern that organizes an application into three logical layers



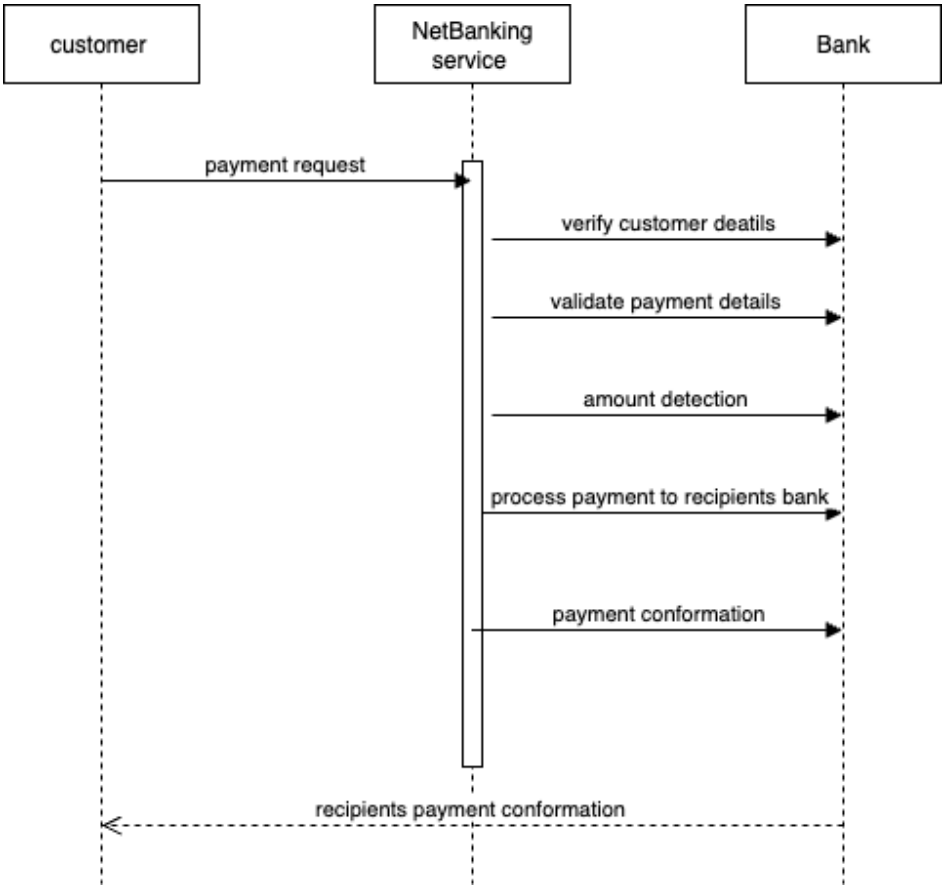
Q 4. Explain Domain Model for Customer making payment through Net Banking

A Domain Model is a visual representation of real-world objects (entities) in your system, including their attributes and relationships**.**



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

A Sequence Diagram is a type of UML (Unified Modeling Language) diagram. that shows how objects or components interact with each other in time and what order.



Q 6. Explain Conceptual Model for this Case.?

A Conceptual Model is a high-level representation of the system that shows entities and their relationships, without focusing on technical implementation

It provides a clear and simplified view of domain, making it easier to understand.

Key elements of conceptual model
Entities – customer, product, order, payment
Attributes – customer id, mail, phone no
Relationships – customer place a order

Q 7. What is MVC architecture?

The Model-View-Controller (MVC) frameworks is an architectural pattern that separates an application into three logical components Model, View, Controller.

Model – represents the application or business logic layer
View – represents the presentation layer
Controller – represents the database layer

Rules to Derive Classes from Use Case Diagram

Model (entity class)
Derived from nouns in a use case descriptions
Represents core business objects

Example: “customer makes a payment ” – customer, payment

View (boundary class)
Derived from system interactions or user interface screens
Linked to actors in the use case diagram

Example : paymentform, loginpage

Controller class
Derived from actions or use cases
One controller per use case

Example: paymentcontroller

Guidelines to Place Classes in Three-Tier Architecture

Tier	Classes type	Role
Presentation layer	Boundary class (view)	Handle user input/output
Application layer	Controller class	Handle logic,control flow
Database layer	Entity class(model)	Handle data, rules

Example: for use case "customer makes payment via net banking "

Customer- entity class database layer
Payment – entity class Database layer
Netbankingpayment entity database layer
Payment page- boundary class presentation layer
Payment controller- controller application layer

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

Stage	Activities	Resources
Pre project	Identify business problem Initial stakeholder discussions	Business case Stakeholder list
Planning	Participate in project scoping Define BA roles and responsibilities Estimate efforts	BA work plan
Requirements gathering	Conduct interviews, workshops Gather functional and non functional requirements	BRD and FRD Stakeholder requirements
Requirements analysis	Analyze requirements Identify gaps Prioritize requirements	Use case diagrams RTM Process flow
Design	Translate requirements to functional specifications Validate UL/UX mockups Review design	Wireframes Data flow diagrams
Development	Clarify requirements to developers Handle requirements change request	Clarifications logs Updated requirements documents
Testing	Review test case Support QA testing Perform requirements traceability	Test case RTM
UAT	Conduct UAT testing Support end users Feedbacks and bugs	UAT test scripts UAT sign-off Feedback reports

Q 9. What is conflict management? Explain using Thomas – Kilmann technique.?

Conflict Management is the process of handling disagreements or clashes between individuals or teams in a constructive and effective way.

The goal is to minimize negative impact and maximize positive outcomes such as better communication, innovation, or decision-making.

Thomas Kilmann technique is a widely used tool for assessing conflict resolution style and guiding individuals in selecting appropriate strategies to manage conflicts.

1. **Assertiveness-** the extent to which you try to satisfy your own concerns
2. **Cooperativeness –** the extent to which you try to satisfy the other person's concerns

5 conflict handling styles

Competing

High assertiveness, low cooperativeness. you pursue your own concerns at the expense of others

When quick and decisive action needed

Collaborating

High assertiveness, high cooperativeness. You work together to find a win-win solution

When both parties concerns are important and long term solutions are needed

Compromising

Moderate assertiveness and cooperativeness. Each party gives up something to reach a middle ground

When time is limited and both sides need to gain something

Avoiding

Low assertiveness and cooperativeness. You sidestep or postpone the conflict

When the issue is trivial or to cool down emotionally charged situations

Accommodating

Low assertiveness, high cooperativeness. You yield to the other party needs.

When preserving harmony is more important than the issue itself

Q 10. List down the reasons for project failure.?

A project failure occurs when a project does not meet its goals or expectations.

Most project failures are caused not by technical problems but by people and process related issues.

The reasons for project failure

- **Poor planning**
  Unrealistic schedules, budget or resource allocation
- **Unclear requirements**
  Requirements are vague, incomplete or misunderstood
- **Lack of stakeholder involvement**
  Poor involvement from business users or clients
- **Scope creep**
  Uncontrolled changes or additions to project scope without proper review
- **Poor communication**
  Team members or stakeholders are not informed
- **Technical challenges**
  Tools, platform, or frameworks are unsuitable
- **Inadequate risk management**
  Risks were not identified, tracked properly
- **Unrealistic deadlines**
  Pressure to deliver too fast causes burnout and quality issues
- **Lack of user training**
  End users don’t accept or know how to use the system properly
- **Lack of skilled resources**
  Team lacks necessary technical or domain
- **Budget overruns**
  Costs exceed initial estimates due to poor tracking or changes

Q 11. List the Challenges faced in projects for BA.?

Challenges faced by business analyst

- **Unclear or changing requirements**
  Stakeholder often don't know exactly what they need or keep changing requests
- **Lack of stakeholder engagement**
  Stakeholders re not available or don't actively participate in discussions
- **Poor communications**
  Miscommunications between technical and business team leads to gaps
- **Scope creep**
  Continuous additions to scope without proper impact analysis or approvals
- **Incomplete documentation**
  Time pressure or lack of clarity results in missing requirements
- **Limited domain knowledge**
  BA may not fully understand the business domain
- **Inadequate testing involvement**
  BA may not be involved enough in testing/UAT to ensure requirements are met
- **Stakeholder conflicts or politics**
  Internal politics can affect decisions and prioritization
- **Technical limitations**
  Requirements may not be feasible due to system or budget constraints
- **Tool/process limitations**
  Lack of access to modeling tools or standardized documentation processes

Q 12. Write about Document Naming Standards.?

A document numbering standard is a systematic approach to assigning unique identifiers to various document created and used throughout the development process.

This makes it easier to organize, search and manage documents

Why use naming standards
Avoid confusion between versions

Improve searchability and tracking

Ensure consistency across the team

Prevent duplication or overwriting files

Typical elements in a naming convention

project code/name - BANKAPP

document type - BRD,FRD, UAT

version number - v1.0, v2.1

date - yyyy-mm-dd

status - FINAL, REVIEW, DRAFT

Q 13. What are the Do's and Don'ts of a Business analyst.?

DO's	**DON'T's**
Consult an SME for clarifications in requirements	Never say NO to the client
Go to client with a plan mind with no assumptions. listen carefully and completely until client is done	There no word as "By default'
Try to extract maximum leads to the solutions from the clients himself	Never imagine anything in terms of GUI
Concentrate on the important requirements	Don't interrupt the client when he is giving you the problem
Question the existence	Never try give solutions to the client straight away with any previous experience
Always use 5W1H	Banned word for BA is "I KNOW"

As a BA inputs are requirements and outputs are use case and activity diagrams, flowcharts

Q 14. Write the difference between packages and sub-systems.?

Package	Sub system
Package is folder that groups related classes or files	Sub system is a small system inside a bigger system
Used for keeping code organized	Used for dividing big systems into smaller working parts
Smaller and more focused	Larger and more comprehensive
Represented by a folder icon in class diagrams	Represented like a package but treated as a complete module
It may contain classes, interface and other packages	It may contain multiple packages, components and interfaces
Example: a package for login, payment or report	Example : a sub system for billing, inventory, HR system

Package helps organize code within a project

Ex: application development companies work on packages

Sub systems help break down a complex system into independent, functional units.

Ex: product development companies work on sub systems

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

Camel casing is a naming convention used in computer programming
It is used for naming variables, functions, and identifiers

- The first word starts with a lowercase letter
- Each new word starts with a capital letter
- There are no spaces or underscores

Example: userName, totalAmountPaid

Why is camel casing important
Improves readability of code
Keeps naming consistent across teams and files
Makes your code look professional

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

A development server refers to a dedicated environment or server that is used during the software development process.

A development server is an environment where the development team builds and tests the application before it moves to QA testing , UAT or production

It is used for writing and testing code, unit testing by developers and initial testing

As a BA we have only some accesses

Business analyst has read only or review access(UI/data)

Read access – to view deployed screens, forms or test UI

Test/review access – to validate if development aligns with requirements

Screenshot capture – BA can take screenshot to compare with BRD/FSD

Download reports/UI – if reports are generated, BA may validate outputs and data

Q 17. What is Data Mapping.?

Data mapping is the process of connecting data from one source to another.

It's like creating a guide or map that shows how data in one place corresponds to data in another place

This is especially important when you are moving data between different systems or database to ensure that the data stay consistent and accurate

It prevents data loss or mismatch

Tools used for data mapping

- Excel or google sheets
- SQL Queries
- ETL tool like Talend, informatica, Apache Nifi

Example

Source system	target system	mapped field
customerName	FullName	same data, different name
Date of birth (DOB)	BirthDate	foarmat may differ

Where data mapping is used

System integration – to connect two applications

Data migration – to move data from old to new system

Reporting & analytics – to bring data from multiple sources together

ETL process (extract, transform, load)- to clean and transform data before storing

Q 18. What is API.?

API (application programming interface) is a set of rules that allows two different software systems to communicate with each other

It defines the methods and data formats that application can use to request and exchange information

Example: you use an app to book a ticket – the app uses an API to connect with an airline system

Your food app uses an API to show the restaurant menu and track delivery

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

your application accepts date in dd-mm-yyyy format
US application sends dates in mm-dd-yyyy
Both systems need to exchange customer data

Steps to handle API Integration with different date formats

1. Receive API data
   Use HTTP GET or POST to fetch the data from the US system

2. Identify date format
   Look for date fields like orderdate, birthdate, etc

3. Convert date format
   Use date converter function to change mm-dd-yyyy to dd-mm-yyyy

4. Store or process data
   Once converted, store or use in your system in your required format

5. Send response
   If needed, send back conformation in expected format