**Capstone Project 3– Part -1**

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

Answer-

Boundary Classes- 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 dependant on the application logic. It remains same if there is there is any modification in the business logic. View class is the data required by the query. View class is represented as boundary class ad it also known as Form Class. In this actor speaks to the system and vice-versa through the boundary. Ex. Payment option boundary, card payment boundary.

Boundary Classes

Controller classes- Controller class is working based on the user’s command. It understands the command and request given by user through the boundary and form class. Whatever the user sends the request for something then it always go through the controller. The Controller is responsible for intercepting the request 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. It acts as intermediaries between boundary and entity class. Ex. Payment initiated controller, card payment controller.

Controller classes

Entity Classes- All Model class represent as Entity Class. The model Class known about all the data that need to be displayed. It is model who aware about all operations that can be applied to transform the class. It only represents the data of application. The model represents enterprise data and the business rules that govern access to and update of this data. Entity class represent core data and business logic of the application. Ex. Customer, Payment

Entity Classes

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

Answer-

**User Layer**- User Layer refers to the Presentation Layer, topmost layer of the architecture. It also known as “Presentation layer” It handles user interface (UI) component such as screens, pages

Boundary Classes

Payment method selection Boundary

Card Payment boundary

**Business Logic**- middle layer of the architecture. It acts as intermediary between the presentation layer and the data Storage layer. This layer contains the core logic of the application

Controller classes

Payment controller

Wallet controller

Card payment controller

**Data tier**- Bottom most layer of the architecture. It responsible for storing and retrieving data.

Entity Classes

Customer (Entity class)

Payment (Entity Class)

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

Answer-

Domain modelling is known as conceptual modelling. A conceptual model is depicts the concept (idea, thing, object) that are easily identifiable in the problem description. It defines structure relationship and behaviour of entities within specific problem domain.

|  |
| --- |
| Customer |
|  | Customer ID |
|  | Customer NameCustomer detailsAddressAccount details |

|  |
| --- |
| Bank |
|  |  |
|  | Bank NameLocationBranch code |

|  |
| --- |
| Account |
|  |  |
|  | Account NoAccount typeAccount Holder NameBalance |

|  |
| --- |
| Payment |
|  |  |
|   | Payment IDAccountPayment detailsstatus |

|  |
| --- |
| Net Banking Service |
|  |  |
|  | AuthenticationFund transferTransaction HistoryAccount Management |

|  |
| --- |
| Authentication |
|  |  |
|  | UsernamePasswordOTP |

|  |
| --- |
| Transaction |
|  |  |
|  | Transaction IDRecipient detail AmountTimestamp |

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

Answer-

Sequence Diagram- The Sequence diagram is used primarily to show the interaction between classes in the sequential order in which those interaction occur. A sequence diagram can map a scenario describe by a use case in step detail to define how classes collaborate to achieve your application goal.



Q6. Explain Conceptual Model for this Case - 4 Marks

Answer-

A **Conceptual Model** represents high-level business entities and their relationships without focusing on implementation details. It Provide clear and simplified view of the domain, making it easier to understanding

For the given scenario, we can model the **Customer** and **Payment** as entities with the following relationships:

1. Entity- Customer, Product, Payment and Order
2. Attributes- customerID, name, email, PhoneNumber
3. Relationship- For Example -

A **Customer** can place order

A **Customer** can make a **Payment.**

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

Answer-

**Model-View-Controller (MVC)** is a **software architectural pattern** that separates an application into three interconnected components

Model, View, Controller

Model- All Model class represent as Entity Class. The model Class known about all the data that need to be displayed. It is model who aware about all operations that can be applied to transform the class. It only represents the data of application. The model represents enterprise data and the business rules that govern access to and update of this data.

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 dependant on the application logic. It remains same if there is there is any modification in the business logic. View class is the data required by the query. View class is represented as boundary class ad it also known as Form Class. In this actor speaks to the system and vice-versa through the boundary.

Controller- Controller class is working based on the user’s command. It understands the command and request given by user through the boundary and from class. Whatever the user sends the request for something then it always go through the controller. The Controller is responsible for intercepting the request 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. It acts as intermediaries between boundary and entity class.

MVC Architecture Rules- 1) Combination of one Actor and a use case result in one boundary class.2) Combination of Two Actors and Use case result in two boundary class 3) Combination of Three Actors and Use case result in Three boundary class. But Only one primary actor is to be consider with a use case 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:

In **3-Tier Architecture**, the classes are placed in the following layers:

1. Presentation Layer (User Layer) → View Classes

Example is OrderScreen, PaymentUI

1. Business Layer → Controller Classes

Example is OrderManager, PymentProcessor

1. Data Layer → Model Classes

Example is OrderDB, PaymentDB

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

Answer-

|  |  |  |
| --- | --- | --- |
| Stage | Activities | Artifacts & Resources |
| Pre Project | Understand business needs and goals.Identify stakeholders.Conduct feasibility analysis. | Business CaseFeasibility ReportStakeholder List |
| Planning | Assist in defining project scope.Identify risks and constraints.Support project charter creation | Project CharterRisk RegisterHigh-Level Requirements |
| Project Initiation | Facilitate kick-off meetings.Define key business objectives.Establish communication plan. | Stakeholder RegisterCommunication PlanProject Roadmap |
| Requirements Gathering | Conduct stakeholder interviews and workshops.Document business, functional, and non-functional requirements | Business Requirements Document (BRD)Functional Requirements Specification (FRS) |
| Requirement Analysis | Analyze requirements for feasibility.Identify gaps and ambiguities.Prioritize requirements based on business needs | Use Case DiagramsProcess Flow DiagramsRequirement Traceability Matrix (RTM) |
| Design | Collaborate with designers and architects.Validate design against business needs.Ensure usability and workflow efficiency | WireframesUI/UX Mock-upsSystem Design Documents |
| Development | Clarify business requirements for developers.Validate system components align with requirements.Conduct interim reviews | Updated RTMFunctional Design DocumentsChange Request Documents |
| Testing | Define test scenarios and validate test cases.Participate in system testing and defect triaging.Ensure requirements coverage | Test CasesTest PlanDefect Logs |
| UAT | Support stakeholders during UATGather feedback and document issues.Ensure system meets business objectives | UAT PlanUAT Test ResultsSign-off Document |

Q9. What is conflict management? Explain using Thomas – Kilmann technique – 6 Marks

Answer-

Conflict Management- Conflict management is the process of resolving conflicts or disagreement between individuals or group in a constructive manner.

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

5 Steps of conflict management

1. Identify the conflict
2. Discuss the details
3. Agree with root problem
4. Check for every possible solution for the conflict
5. Negotiate the solution to avoid future conflict

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

Answer-

1. Poor Planning- Lack of a structured project plan leads to missed deadlines, budget overruns, and unclear roles and responsibilities. Effective planning includes defining milestones, tasks, and dependencies
2. Unclear objective and requirements - When business goals and project requirements are not clearly defined, teams struggle with misalignment, leading to rework and delays. A well-documented requirement ensures clarity
3. Inadequate Risk Management - Ignoring potential risks (technical, financial, or operational) can result in major disruptions. A risk management plan helps identify, assess, and mitigate risks before they escalate
4. Poor Communication - Ineffective communication between stakeholders, teams, and management leads to misunderstandings and misinterpretations, affecting project progress and decision making
5. Scope Creep - Uncontrolled changes or additions to project requirements without proper assessment result in extended timelines, increased costs, and resource exhaustion
6. Lack of stakeholder Engagement - If key stakeholders (e.g., clients, users, sponsors) are not actively involved, the project may fail to meet expectations or face resistance during implementation
7. Resource Constraints - Insufficient human, financial, or technical resources limit project execution, leading to delays, low-quality deliverables, and team burnout
8. Technical Challenges - Issues such as outdated technology, system incompatibility, poor software architecture, or lack of technical expertise can lead to project failure

Q11. List the Challenges faced in projects for BA – 6 Marks

Answer-

Here are some challenges which is faces during Project For BA-

1. Unclear requirements and changing requirements in Project - One of the biggest challenges for a BA is dealing with vague or frequently changing requirements, leading to scope issues and rework. Proper documentation and stakeholder collaboration help mitigate this.
2. Managing stakeholder expectations - Different stakeholders may have conflicting priorities and unrealistic expectations. A BA must ensure clear communication, manage expectations, and align business goals.
3. Scope Creep and Scope Management - Continuous addition of new features without proper evaluation leads to project delays and budget overruns. A BA must enforce scope control using a Change Management Process.
4. Time and resources Constraint - Limited time and resources can impact requirement gathering and analysis. Prioritizing tasks and using Agile/Scrum methodologies can help manage constraints.
5. Quality Assurance and Testing - If requirements are not well-defined, testing teams may struggle with validation. BAs must ensure that test cases align with business needs and acceptance criteria.
6. Documentation and Knowledge management - Proper documentation (e.g., BRD, FRS, Use Cases) is essential for project continuity. Lack of knowledge transfer can create dependency issues.
7. Technology constraints and Complexity - BA often face challenges in aligning business needs with technical feasibility due to system limitations, integration issues, or outdated technology. Close collaboration with technical teams is crucial.

Q12. Write about Document Naming Standards – 4 Marks

Answer-

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

Example is We have project with the ID “PROJ123,” and we are working with Requirement Specification Document

Project ID : PROJ123

Document Type : REQ

Version : 1.0

Date : 2025-01-31

The Document Identifier could be : PROJ123-REQ-1.0-2025-01-31

Q13. What are the Do’s and Don’ts of a Business analyst – 6 Marks

Answer-

|  |  |  |
| --- | --- | --- |
| Sr. No. | Do’s | Don’ts |
| 1. | Consult an SME for clarification in requirements | Never Say No to the client |
| 2. | Go to the client with plain mind with no assumptions.Listen carefully and completely until the client is done and the BA Ask queries | There is no word as “by default” |
| 3. | Try to extract maximum leads to the solution from the client itself. | Never imagine anything in term of GUI.  |
| 4. | Concentrate on the important requirements. | Don’t interrupt the client when he is giving you the problem.  |
| 5. | Question the existence of existence/Question everything. | Never try to give solutions to the client straight away with your previous experience and assumptions. |

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

Answer-

Packages - A **Package** is a logical grouping of related classes, components, or diagrams to improve organization and modularity.

It is collection of components which are not reusable in nature.

Example: Application development companies work on packages.

sub-systems - A **Subsystem** is a self-contained module that represents an independent unit within a system, having well-defined functionalities.

It is collection of components which are reusable in nature.

Example: Product development companies work on Subsystem.

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

Answer-

Camel casing is naming convention used in computer programming. Its is used for naming variable, functions and identifiers

Example:

CamelCase : camelCaseexample

 customerName

 customerID

In Camel casing first words start with lower case letter and each subsequent word begins with an upper letter.

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

Answer-

A **Development Server** is an environment where software developers build, test, and debug applications before deploying them to production. It allows for continuous development and integration without affecting live users.

**Development Server** used by developers for coding, testing, and debugging

A **BA access** to the development server is typically limited only. but they may have access to certain components for requirement validation and collaboration.

Requirement Validation - Verify that the development aligns with documented business requirements

View Development Builds - Access to early builds to ensure the system meets functional expectations

Database Access - Limited access to verify data integrity and business rules.

Q17. What is Data Mapping 6 Marks

Answer –

Data mapping is the process of connecting data from one source to another. **Data Mapping** is the process of linking or transforming data from one format, source, or structure to another. It ensures that data from different systems is correctly transferred, integrated, and used efficiently.

Its like creating a guide or map that shows how data in one place corresponds to data in another place.

It ensures **data consistency** across systems, Facilitates **data migration, integration, and transformation**

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

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 10 Marks

Answer-

An API or Application Programming Interface is a set of rules and tools that allows different software applications to communicate with each other.

 It defines the method and data formats that applications can use to request and exchange information.

The application follows the date format DD-MM-YYYY

It receives order data from a **US-based application** where the date format is MM-DD-YYYY

To ensure consistency, the system must **convert the date format** before storing or displaying the data

The US-based system sends data through an API in **JSON format**

The backend system processes the request and convert DD-MM-YYYY to MM-DD-YYYY

The conversion can be handled using **backend programming**

After conversion, the formatted date is **stored in the database** and **sent in the response.**