## Capstone Project 3 – Part 1

*A customer can make a payment either by Card or by Wallet or by Cash or by Net banking.*

**Question 1: Draw a Use Case Diagram.**



**Question 2: Derive Boundary Classes, Controller classes, Entity Classes.**

* **Boundary Class:** It ensures the interactions with user and external system. In our use case, “Select Payment Method” and “Payment gateway interfaces” act as Boundary Class.
* **Controller Class:** The controller class hold the logic and handles the flow of operation. It is an intermediatory between entity and boundary class. Here “Payment Controller”, “Card payment controller”, “Payment Wallet Controller” and “Net banking payment controller” are Controller Classes.
* **Entity Class:** The entity class represents the data and business logic of the application. Payment, Card, Payment wallet, Net banking and COD are Entity Classes.

**Question 3: Place these classes on a three tier Architecture.**

The three layers of 3 tier architectures and their classes are:

* Application Layer: Select Payment Method class
* Business Logic Layer:
	+ Payment Controller class.
	+ Wallet Controller class.
	+ Card Controller class.
	+ Net banking controller class.
* Data Layer: User (Customer) entity class, payment entity class (card, wallet, net banking, COD).

**Question 4: Explain Domain Model for Customer making payment through Net Banking.**

A domain model is a visual representation of the key concepts (objects) and their relationships within a specific area of knowledge (domain). It shows the relationships between entities.



**Question 5: Draw a sequence diagram for payment done by Customer Net Banking.**

A sequence diagram is a type of interaction diagram that shows how objects interact with each other in a particular scenario.



**Question 6: Explain Conceptual Model for this Case.**

A conceptual model is a high-level representation of a system or domain, focusing on the key concepts and their relationships, without getting into implementation details. It serves as a foundation for understanding and communicating the core ideas of a system.

Key elements of this conceptual model are:

* Entities: User and Payment server.
* Attributes: Customer Name, Id, contact details, amount, order ID, etc.
* Relationships: Customer placing an order.

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

MVC architecture (Model – View – Controller) is a set of rules to identify classes from use-case diagrams. MVC rules are applied on each use case to derive classes.

The 3 components of MVC architecture are:

* **Model:** Applications data and business logics are represented in Model layer. All Model classes are represented as Entity Classes.
* **View:** It handles the presentation of data. It is responsible for displaying the data to user.
* **Controller:** It works as an intermediatory between the Model and View. This receives the user input and updates the model accordingly and then selects the appropriate view to display to the user.

The MVC rules to derive classes from use case diagram are:

* Combination of one actor and one use-case results in one boundary class.
* Combination of two actor and one use-case results in two boundary class.
* Combination of three actor and one use-case results in three boundary class.
* Use case will result in a controller class.
* Each Actor will result in one entity class.

Guidelines to place classes in 3-tier architecture are:

* Place all Entity Classes in DB layer.
* Place primary actor associated boundary class in application layer.
* Place Controller class in application layer.
* If governing body influence or Reusability is there with any of remaining boundary classes, place them in business logic layer else place them in application layer.

**Question 8: Explain BA contributions in project (Waterfall Model – all Stages).**

BA contribution in a Waterfall-model project are as follows:

|  |  |  |
| --- | --- | --- |
| Stages | Activities | Artifacts |
| Pre Project | Enterprise Analysis, SWOT Analysis, Gap Analysis, Feasibility Analysis, Root Cause Analysis, Business Case writing, Risk Analysis.  | Business Case, SOW. |
| Planning, Estimation & Assessment | * Understand project plan from PM
* Conduct stakeholder Analysis
* Plan BA approach strategy (Requirement gathering techniques, Communication, Documentation, Tools to be used, handle change request).
 |  |
| Requirement Gathering | * BA prepares BRD by interacting with client. (Requirement gathering techniques).
* Prototyping.
* Sorting the requirements.
* Prioritize requirements. (MoSCoW)
* Validating Requirements. (FURPS)
 | BRD |
| Requirements Analysis | * Draw UML diagrams. (Use Case and Activity Diagrams)
* Prepares functional requirements.
* Taking signoff on SRS from client.
* Prepare and Own RTM.
* Trace the requirement through each phase of SDLC
 | FRD, RTM |
| Design | * Help testing team in preparing test cases from Use Case Diagrams.
* Prepare End User Manual.
* Update RTM.
 | Solution Document |
| Coding | * Organize JAD session.
* Update end user manual.
* Update RTM.
* Conduct regular meetings with stakeholders and technical team on status of the project.
 |  |
| Testing | * Assist testing team in preparing test cases.
* Perform functional testing.
* Prepare client for UAT.
* Update RTM.
* Update End User Manual.
* Take sign-off from the client.
 |  |
| Deployment & Implementation | * Shares RTM and End User Manual to client and PM.
* Plan and organizes training sessions for End User.
* Prepares lesson learned from this project.
 |  |

**Question 9: What is conflict management? Explain using Thomas – Kilmann technique.**

Business Analysts work with multiple stakeholders and this often leads to conflicts regarding requirements, priorities and solutions. The process of managing these conflicts in called Conflict Management.

5 Steps to manage any conflict are:

* Identify the conflict.
* Discuss the details.
* Agree to the root problem.
* Check for every possible solution.
* Negotiate the solution to avoid future conflict.

There are 5 options to manage a conflict:

* Competing
* Avoiding
* Accommodating
* Collaborating
* Comprising

Thomas-Kilmann technique is a framework for understanding and understanding these conflicts.

Cooperativeness

Assertiveness

**Question 10: List down the reasons for project failure.**

The main reasons for project failure are:

* Improper Requirement gathering.
* Continuous change in requirements.
* Lack of user involvement.
* Lack of executive support.
* Unrealistic expectations.
* Improper Planning.

**Question 11: List the Challenges faced in projects for BA.**

The challenges faced by BA in a project are:

* Lack of training.
* Obtaining Sign-off on requirements.
* Change Management with respect to time and budget.
* Co-ordination between developers and testers.
* Conducting meetings.
* Making sure status report is effective.
* Driving clients for UAT completion.
* People Management.
* Making sure overall project health is in good shape and is delivered in time without any issue.

**Question 12: Write about Document Naming Standards.**

Document naming are very crucial part of the project. Every document is named according to the project and the standards followed by company. A general document naming standard is:

***[projectID][Document\_type] V[x]D[y]. ext.***

**Question 13: Do’s and Don’ts of a BA.**

The Do’s and Don’ts of a BA are:

* Never say NO to a client.
* There is no word “By Default”.
* Never imagine anything in terms of GUI.
* Question everything.

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

|  |  |
| --- | --- |
| Packages | Subsystems |
| A package is a grouping of related classes, interfaces, or other packages in object-oriented design. | A sub-system is a collection of related components that function as a smaller system within a larger system. |
| Used mainly in software development to organize code and manage dependencies. | Represents a functional part of a larger system, focusing on a specific domain or task. |
| These are non-reusable and used by application development companies. | These are reusable and used by Product development companies. |

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

Camel-casing is a naming convention in which multiple words are written without spaces, and each word (except the first in lowerCamelCase) starts with a capital letter.

Camel casing is used in:

* In ERD, camel casing is used to name tables and columns.
* While naming requirement documents, writing variables and system attributes.
* Naming end points and system interactions while defining API’s.
* While creating UML diagrams.

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

A Development Server is an environment where software applications are built, tested, and modified before deployment. It is mainly used by developers to write, debug, and test code.

A Business Analyst have:

* Read Access to Db: View dB schema, tables, etc.
* Access for requirement verification to check if features are as per business requirement.
* Partial access to create input data to test.
* Access to view logs and track errors.
* Test application UI before UAT.

**Question 17: What is Data Mapping?**

Data Mapping is the process of linking data fields from one source to a corresponding field in another system, database, or file. It is essential for data integration, migration, transformation, and analysis.

Key aspects of data mapping are:

* Source data.
* Target Data.
* Mapping rules.
* Transformation logic.

**Question 18: What is API?**

**Explain how you would use API integration in the case of your application Date format is dd-mm-yyyy and it is accepting some data from Other Application from US whose Date Format is mm-dd-yyyy.**

An API (Application Programming Interface) is a set of rules and protocols that allows different software applications to communicate with each other. It acts as a bridge between systems, enabling them to exchange data and functionality without exposing the internal workings of the application.

Key aspects of API are:

* Endpoint: The specific URL where the API receives requests.
* Request: The data sent to the API, usually in JSON or XML format.
* Response: The data returned by the API after processing the request.
* Methods (HTTP Methods)
* Authentication: Security mechanism (e.g., API keys, OAuth, JWT) to control access.

If our application accepts dates in dd-mm-yyyy format but is receiving data from a US-based application where the date format is mm-dd-yyyy, we need to ensure proper data transformation before storing or processing the data.

Steps to convert are:

* Identify API data format.
* Implement Data Transformation in backend using coding languages.
* Modify response in our system.
* Document the conversion.