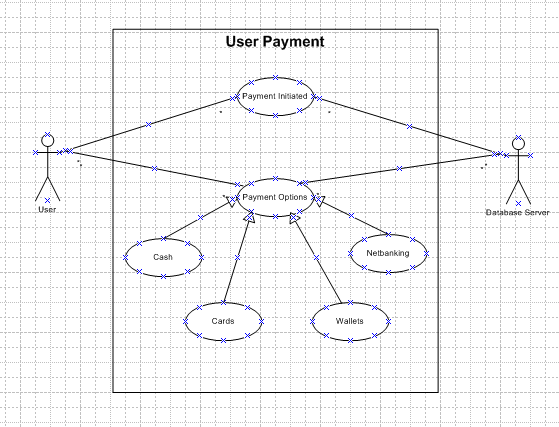
**Nurturing Process - Capstone Project3– Part -1**

Question 1) A customer can make a payment either by Card or by Wallet or by Cash or by Net banking. Draw a Use Case Diagram.

Answer) Use Case Diagram: 

Question 2) Derive Boundary Classes, Controller classes, Entity Classes.

Answer) Boundary Class: A Boundary Class is used to handle the interactions between the system and external actors.

Controller Class: This acts as an intermediary between Boundary and the Entity Class.

Entity Class: Represents the core data and business logic of the application.

Here, All Actors would become an Entity Class.

All the Use cases become the Boundary Class.

The Actor and the Use case relationship where there is no third party involved becomes a controller class.

* Boundary Class – Payment Option Boundary Class

Card Payment Boundary Class

Wallet Payment Boundary Class, Cash Payment Boundary Class, Net Banking Payment Boundary Class.

* Controller Class – Payment initiated controller class.

Card Payment controller class

Wallet Payment controller class

Cash Payment controller class

Net Banking Payment controller class

* Entity Class – Customer Entity Class

Payment Entity Class

Card Entity Class

Wallet Entity Class

Server Entity Class

Question 3) Place these Classes in 3 tier Architecture.

Answer) Three Tier Architecture divides the Application into three logical layers.

Application Layer: The topmost or the presentation layer which handles the User interface.

Business Logic Layer: Middle layer of the Architecture acts as an intermediary between the Presentation and the Database layer. This layer contains the core logic of the application. All the reusable or frequently changing components like the Governing body rules, Payment gateways, RBI rules for banks.

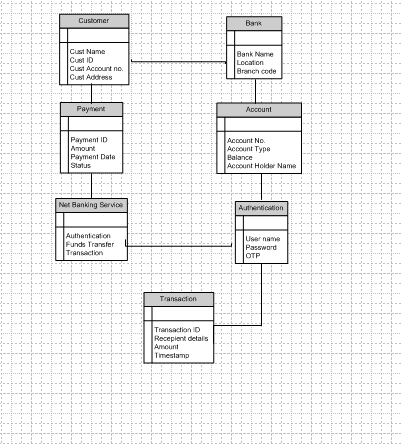
Database Layer: Bottom most layer of the Application responsible for storing and retrieving data Ex: Database components connecting to databases like MySQL database, Oracle Database.

|  |
| --- |
| **User Layer** |
| Payment Method Selection Boundary |
| Card Payment Boundary |
| Wallet Payment Boundary |
| CashPayment Boundary |
| Net Banking Payment Boundary |
| **Business Logic Layer** |
| Payment Controller |
| Card Payment controller |
| Wallet Payment controller |
| Cash Payment controller |
| Net Banking Payment controller |
| **Data Tier** |
| Customer Entity Class |
| Payment Entity Class |
| Card Entity Class |
| Wallet Entity Class |
| Bank Account Entity Class |

Question 4) Explain Domain Model for customer making payment through Net Banking.

Answer) Domain Model is a conceptual representation that defines the structure relationships behaviour of the entities with in a specific problem domain.

Its a visual representation connecting the relationships between different components, tables and entities in a database.



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

Answer) A Sequence diagram is a type of interaction diagram that shows how objects interact in a particular sequence of time.

It is widely used to visualize the flow of messages and events in a system making it easier to understand the dynamic behaviour of the system it shows how data flows sequentially in a more orderly manner.



Question 6) Explain Conceptual Model for this Case.

Answer) A Conceptual model is a high level representation of a system that helps in understanding, visualizing and communicating the essential aspects of a domain.

It’s a high level representation of a system focusing on business concepts and their relationships.

Helps visualizing the over all structure and flow of the process.

Key elements of the Conceptual Model are

Entities – Customer, Product, Order & Payments etc.

Attributes – Customer ID, Name, Email, Phone no.

Relationships – A customer makes a payment, A customer places 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.

Answer) The Model View Controller framework is an architectural pattern that separates an application into three main logical components Model, View, Controller.

Model: Represents the Busines logic layer and the data of the application.

View: Represents the Presentation layer or the application layer.

Controller: Acts as an intermediary b/w model and view.

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 2 boundary classes.
3. Combination of three actors and a use case results in three boundary classes.
4. Use Case is a controller class.
5. Actor is an Entity.

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

Answer)

|  |  |  |  |
| --- | --- | --- | --- |
| **Stages** | **Activities** | **Artifacts** | **Resources** |
| Pre Project | Enterprise Analysis – SWOT Analysis, GAP Analysis, Market Research, Feasibility Study, Root cause Analysis, Decision Analysis, Strategy Analysis, Enterprise Architectural Frameworks, Project Scope and Business Case Writing, Risk Analysis. | Business Case Document  SOW (Statement of Work)  PO (Purchase Order) | Sr. BA |
| Planning & Estimations & Assessment | BA Conducts Stakeholder Analysis.  Plans BA approach strategy (Req. Gathering techniques, Communication, Req. Management, Documents to follow, Tools to use, Change request handling methodology) for the Project |  | Sr. BA |
| Requirements Gathering | Stakeholders Identify and Document.  Client gives BRD or BA prepares BRD by interacting with Client Brainstorming, Document Analysis, Reverse Engineering, Interviews, Workshops, Focus Groups, Observation, Questionnaires.  Prototyping can be used by BA to make the Client to give more specific requirements.  Sort the gathered requirements (avoiding duplicate reqs, grouping into similar functionality or into modules)  Prioritize requirements – MOSCOW  Validate requirements - FURPS |  | BA |
| Requirement Analysis | Draws UML Diagrams ( Use case and Activity diagrams)  Prepare Functional Requirements from Business requirements.  All Architects come up with Technical Requirements (SSD)  SRS wil have Functional Requirements and Technical Requirements.  Takes Signoff on SRS from the Client, the first legal binding doc between the Business and the Technical team.  BA Prepares RTM from SRS before design phase starts.  BA traces how requirements are delat in each phase of development life cycle from Design till UAT. | Function Requirements Specification.  SSD ( Supplementary Support Document)  SRS ( Software Requirements Specification)  RTM (Requirements Traceability Matrix) | BA |
| Design | From the Use case diagram Test Manager or BA will prepare test cases.  Communicates with the Client on the design and Solution documents (updates status to Client & make them understand how the solution would look like to prepare them to drive UAT)  BA will initiate the preparation of end user manuals.  Updates RTM.  From Use case diagram Solution Architect recommends Architecture of the IT Solution. | Solution Document  Design Document – HDD - ADD | BA |
| Coding | BA organizes JAD sessions.  BA clarifies queries of technical team during coding.  Developers refer diagrams and Transient class of BA and code their unit.  Update RTM.  Conducts regular status meeting with technical team and client and tunes client for participation in UAT. | LDD – CDD Application | BA |
| Testing | BA Prepares test cases from Use cases or assists test manager to do so  BA performs high level testing  BA prepares Client for UAT.  Test data is requested by BA from Client.  Updates end user manuals.  Updates RTM.  Take signoff form the Client on Client Project Acceptance form. | Test concerning Documents  Application with less errors. | BA |
| Deployment and Implementation | Forwards RTM to Client or the PM which should be attached to the Project Closure Document.  Coordinates to complete and share End User manuals.  Plans and Organizes Training sessions for end users.  Prepares lessons learnt from this Project (to take precautions for coming Projects) |  |  |

Question 9) What is conflict management? Explain using Thomas – Kilmann technique

Answer) Conflict Management is the Process of resolving conflicts or disagreements between individuals or groups with in an organization in a constructive manner.

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

5 Steps of conflict management are:

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

Question 10) List down the reasons for project failure.

Answer) The reasons for Project failure are:

1. Improper requirement gathering.
2. Continuous change in requirements.
3. Lack of user involvement.
4. Lack of executive support.
5. Unrealistic expectations.
6. Improper Planning.
7. Poor Planning
8. Technical Challenges

Question 11) List the Challenges faced in projects for BA.

Answer) The challenges faced in Projects for BA are

1. Unclear or Changing requirements
2. Managing Stakeholder Expectations
3. Scope Creep & Scope Management.
4. Time and resource constraints
5. Quality Assurance & Testing
6. Documentation & knowledge Management.
7. Technology constraints and complexity.

Question 12) Write about Document Naming Standards.

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

[ProjectID][Document Type]V[X]D[Y].ext

Question 13) What are the Do’s and Don’ts of a Business analyst.

Answer)

|  |  |  |
| --- | --- | --- |
| **S.No.** | **DO’S** | **DONT’S** |
|  | Go to the Client with a Plain mind with no assumptions. | Never Say NO to the Client. |
|  | Listen carefully and completely until the Client is done and then ask queries. | There is no word as “By default” |
|  | Try to extract maximum leads to the solution from the Client himself. | Never imagine anything in terms of GUI. |
|  | Question the Existence of existence | Don’t interrupt the Client when he is giving you the problem. |
|  | Concentrate on important requirements | Never try to give solutions to the Client straight away with your previous experience and assumptions. |
|  | Consult an SME for Clarification in requirements. | Don’t be washed away by add on functionalities. |

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

Answer)

|  |  |
| --- | --- |
| **Packages** | **Sub Systems** |
| Collection of components which are not reusable in nature. | Collection of components which are reusable in nature. |
| Application development companies work on Packages. | Product development companies work on Sub Systems. |
| Smaller and more focused in scope. | Larger and encompass multiple packages or modules. |
| Manage dependencies at a class/component level. | Manage dependencies at a higher level, defining boundaries and interfaces between different parts of the system. |

Question 15) What is camel-casing and explain where it will be used ?

Answer) Camel Casing is a naming convention used in computer programming and is characterized by removing spaces between words and capitalizing the first letter of each word except for the first word. The name Camel casing is derived from the appearance of the resulting string, which resembles the humps of a camel.

By using camel casing developers can create meaningful and readable names that are easier to understand and follow coding standards. It promotes consistency within the codebase and improves collaboration among the team members.

It is used for naming variables, functions and identifiers.

Question 15) Illustrate Development server and what are the accesses does business analyst has?

Answer)

* A Development server refers to a dedicated environment or server that is used during the software development process.
* It provides a platform for developers and testers to build, test and debug applications before they are deployed to a production environment.
* The development server typically replicates the target production environment to ensure compatibility and accurate testing.
* As a BA we have only limited access only like Read only access, Collaborative access, Limited Configuration Access.
* If a BA joins a Project then we will be able to access the Public Area, Code and test areas of the development server.

Question 16) What is Data Mapping ?

Answer)

* Data mapping is the process of connecting data from one sources 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 we are moving data between different systems or databases to ensure that the data stays consistent and accurate.
* The purpose of data mapping is to ensure that data can be accurately and affectively transferred converted or transformed between different systems databases or formats. It involves identifying the source data elements determining their meaning and structure and mapping then to the corresponding target data elements.

Question 17) 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 or Application Programming Interface is a set of rules and tools that allows different software applications to communicate and interact with each other.

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

To establish API Communication we first need to set up API communication between our application and the other application to exchange data.

Data Formatting: When sending date data from our application to the other application, convert the date from the dd-mm-yyyy format to the mm-dd—yyyy format. This can be achieved by extracting the day, month & year components from the date and rearranging them according to the target format.

Data Parsing: When receiving date data from the other application, parse the mm-dd-yyyy formatted date into our application’s dd-mm-yyyy format. Again, we will need to extract the day , month & year components and rearrange them accordingly.

Data Validation: Perform data validation and ensure that the converted date remains valid after the format conversion. Check for edge cases, such as invalid dates or date ranges that might be affected by the format conversion and handle them appropriately.