**Nurturing Process - Capstone Project3– Part -1/2**

**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**

**Answer**

 

**Q2. Derive Boundary Classes, Controller classes, Entity Classes.**

**Answer:**

**Boundary Class:**

* Handles input and output from the system, like receiving user input from a form or sending response data to a web page.
* Focuses on data translation and formatting to match the external system's requirements.
* Represented with

* A combination of a primary actor and a Use case results in a Boundary Class

**Controller Class:**

* Orchestrates the execution of a use case by calling necessary methods on entities and boundary classes.
* Contains the business logic specific to a use case, including validation and decision making.

* Represented with

* Use Case results in Controller Class.

**Entity Class:**

* Represents real-world objects or concepts within the domain, storing data attributes and related business logic.
* Usually persists data in a database**.**
* Represented with

* Each actor is an entity class

|  |  |
| --- | --- |
| Boundary Class | * CustomerLogin Boundary Class
* CustomercashPayment Boundary Class
* CustomerCardPayment Boundary Class
* CustomerNetBankingPayment Boundary Class
* CustomerWalletPayment Boundary Class
* CustomerPrintRecipt Boundary Class
* BankServerEMailconfirmation Boundary Class
* PrinterPrintsReceipt Boundary Class
* ServerAcceptsPayment Boundary Class
* ServerAcceptsPayment Boundary Class
* ServerAcceptsCashPayment Boundary Class
* ServerAcceptsCardPayment Boundary Class
* ServerAcceptsWalletPayment BoundaryClass
* ServerAcceptsNetBankingPayment Boundary Class
* ServerSendsEmailConfimation Boundary Class
 |
| Controller Class | * Login Controller Class
* Payment Controller Class
* Card Controller Class
* Cash Controller Class
* Net Banking Controller Class
* Wallet Controller Class
* Email Confirmation Controller Class
* Print Receipt Controller Class
 |
| Entity Class | * Customer Entity Class
* Payment Entity Class
* Card Entity Class
* Cash Entity Class
* Wallet Entity Class
* Net banking Entity Class
* Server Entity Class
* Bank Server Entity Class
* Printer Entity Class
 |

**Q3. Place these classes on a three tier Architecture.**

**Answer**

There are some guidelines to place the MVC Classes in the 3 tire architecture.

* Place all the Entity Classes in the Data Base Layer.
* Place all the Primary actor associated Boundary classes in the Application Layer.
* Place all the Controller Class in the Application Layer.
* Place all the Secondary actor associated Boundary Class, Governing Body influence, reusable logic in Business Logic layer.

Login CC, Payment CC, Card CC, Cash CC, Net Banking CC, Wallet CC, Email Confirmation CC,

Print Receipt CC, CustomerLogin BC, CustomercashPayment BC, CustomerCardPayment BC,

CustomerNetBankingPayment BC, CustomerWalletPayment BC, CustomerPrintRecipt BC,

**Application Layer**

BankServerEMailconfirmation BC, PrinterPrintsReceipt BC, ServerAcceptsPayment BC, ServerAcceptsCashPayment BC, ServerAcceptsCardPayment BC, ServerAcceptsWalletPayment BC, ServerAcceptsNetBankingPayment BC, ServerSendsEmailConfimation BC

**Business Logic Layer**

Customer EC, Payment EC, Card EC, Cash EC, Wallet EC, Net Banking EC, Server EC, Bank Server EC, Printer EC

**Data Base Layer**

**Q4. Explain Domain Model for Customer making payment through Net Banking**

**Answer**

**Domain Model**: A domain model is a visual representation of concepts, objects, and their relationships in a system. It is similar to ER model. Domain model has the attributes mentioned. This model focuses on capturing the behaviour of the application.

**Explanation:**

* The Customer table is connected to Bank Table , which is why customer is able to make payments.
* The Customer table is connected to Payment Table for the Customer to do payment.
* Account is in the bank so Account table is connected to Bank Table.
* The Payment table is connected to Net Banking Table as payment is done through net banking method
* The Net Banking Table and the bank table are connected to Authentication table as authentication is performed in both the places.
* Authentication table is connected to Transaction table as authentication ius done during transaction process.



**Q5. Draw a sequence diagram for payment done by Customer Net Banking**

**Answer**

A sequence diagram represents how a data flow will happen in a sequential manner in a system. It is used to show the flow of messages, events or actions taking place between the systems.



**Q6. Explain Conceptual Model for this Case**

**Answer**

This model helps in understanding the key concepts, their relationships and the overall structure of the net banking system. It serves as a foundation for designing the database schema, defining the application architecture and implementing the necessary functionalities within the system.

The relationship between entities can be described as follows.

* Customer: This node represents the customer or users of the net banking application.
* Bank: This node represents a service provider responsible for offering net banking services.
* Service Awareness: Customers should be aware of the net banking service, their features and the process of net banking.
* Privacy of data: As the service is delt with personal information of each customer and their bank accounts the privacy of the customer information is considered high.
* Technology Awareness: The customer should be aware of the underlying technology used.
* Trust and Support: This node represents the support that will be provided for the customer for any issues.

**Q7. What is MVC architecture? Explain MVC rules to derive classes from use case diagram and guidelines to place classes in 3-tier architecture**

**Answer:**

**MVC Architecture:** MVC stands for Model View Controller. It is a design architecture that separates the application into 3 main logical components Model, View and Controller. Each architectural component is built to handle specific development aspects of an application.

**Components of MVC:**

* Model
* View
* Controller

**Model:** The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. It can add or retrieve data from the database. It responds to the controller’s request because the controller can’t interact with the database by itself. The model interacts with the database and gives the required data back to the controller. It represents the enterprise data and the DB tables

**View:** The View component is used for all the UI logic of the application. It generates a user interface for the user. Views are created by the data which is collected by the model component but these data aren’t taken directly but through the controller. It only interacts with the controller**.** It helps render data to the user in specific format. It displays user interface elements.

**Controller:** The controller is the component that enables the interconnection between the views and the model so it acts as an intermediary. The controller doesn’t have to worry about handling data logic, it just tells the model what to do. It processes all the business logic and incoming requests, manipulates data using the Model component, and interact with the View to render the final output.

**Rules to derive classes from use case diagram:**

* Combination of one Actor and a Use case results in one Boundary class.
* Combination of two Actors and a use case results in two Boundary class
* Combination of three Actors and a use case results in three Boundary class.
* Use Case will result in Controller Class.
* Each actor will result in Entity class.

 **Guidelines to place classes in 3-tier architecture:**

There are three layers in MVC Architecture- Data Base Layer, Business Logic Layer, Application Layer.

* All Entity class objects will be placed in Data Base Layer.
* All Primary Actors associated boundary class will be placed in Application Layer
* All Controller Class objects will be placed in Application layer.
* If any governing body influence or reusability is there with any remaining boundary class place them in logical layer.
* All the secondary actors associated boundary classes will be placed in Business Logic Layer.

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

|  |  |  |
| --- | --- | --- |
| **Stages** | **Activities** | **Artifacts and Resources** |
| **Pre Project** | Enterprise Analysis- SWOT Analysis, GAP Analysis, Market Research, Feasibility Study, Root Cause Analysis, Decision Analysis, Decision Analysis, Strategy Analysis, Enterprise Architectural Framework, Project Scope and Business Case writings, Risk Analysis. | Business Case Documents,SOW(Statement of Work)PO(Purchase Order)Sr.BA, Business Architects, Presales Consultants.  |
| **Planning** | 1. Understands 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 Stakeholder Analysis.
5. Plan BA approach strategy (Requirements Gathering techniques, communication, requirement management, Documents to follow, Tools to use, Change request handling for the project
 | PMSr. BA |
| **Requirements Gathering** | 1. Identification and Documentation of stakeholders.
2. Client gives BRD or BA prepares BRD by interacting with Client- Brainstorming, Document Analysis, Reverse 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
5. Prioritize requirements-MoSCoW
6. Validate requirements-FURPS
 | Business Requirement documentBA, PM |
| **Requirements Analysis** | 1. Draws UML Diagrams (Use Case and Activity Diagrams)
2. Prepares Functional Requirements from Business Requirements.
3. All Architects comes up with Technical requirements. (SSD)
4. SRS will have functional requirements and Technical requirements.
5. Takes Signoff on SRS from Client. SRS is the first legal binding between the Business and the technical team.
6. BA prepares RTM from SRS before design phase starts
7. BA traces how requirements are delt in each phase of development life cycle from design till UAT
 | Functional Requirements SpecificationSSD (Supplementary Support Document)SRS (Software Requirements Specification)RTM(Requirements Traceability Matrix)BA, PM, Solution Architect, DB-Architect, NW-Architect. |
| **Design** | 1. From Use Case Diagram , Test Manager or BA will prepare Test Cases
2. Communicates with the client on design and solution documents (updates status to clients and make them understand how the solution would look like to prepare them to drive UAT)
3. BA will initiate the preparation of the End User Manuals
4. Updates RTM
5. From Use Case Diagram Solution- Architects recommends Architecture of the IT solution
6. DB Architects uses Persistence classes and comes up with ER diagrams or DB Schema.
7. GUI Designers will look into transit classes and design all possible screens for the IT Solution
 | Solution Document, Design Document-HDD,ADDBA, PM, Solution Architect, DB-Architect, NW-Architect, GUI Designer, Test Manager. |
| **Coding** | 1. BA organizes JAD Sessions.
2. BA clarifies queries of technical team during coding
3. Developers refers diagrams and Transient and code their Unit.
4. Updates End User manuals.
5. Update RTM.
6. Conducts regular status meetings with technical team and client and tuning client for participation in UAT.
 | LDD-CDD ApplicationDevelopment Team,BA,PM |
| **Testing** | 1. BA prepares test cases from the Use Case or helps the test managers in the process
2. Performs high level testing
3. Prepares client for UAT
4. Requesting test data from client
5. Updates End User Manuals
6. Updates RTM
7. Takes Signoff from Client on Client Project Acceptance form
 | Test DocumentsRTMTesting Team,BA,PM,Client |
| **Deployment and Implementation** | 1. Forwards RTM to client or PM which should be attached to the Project closure document.
2. Coordinates to complete and share End User manuals.
3. Plans and organizes training sessions for End Users.
4. Prepares lessons learned from this project(to take precautions for coming projects).
 | Project Closure documentBAPMClient |

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

**Answer**

**Conflict Management:**

Conflict can occur due to multiple reasons like goals, values, personalities, resources or communication breakdowns. It is an inevitable part in any project. Conflict Management is a process of identifying and addressing conflicts in a healthy and constructive manner. It consists of strategies and techniques aimed at resolving disputes, disagreements, differing perspectives among individuals or groups.

There are 5 steps for conflict management

* Identify Conflict
* Discuss the details
* Agree with root cause
* Check for every possible solution for the conflicts
* Negotiate the solution to avoid the future conflicts.

**Thomas-Kilman Technique:**

It is a approach used to recognize the approaches for conflict management. It is a graphical representation where the X-axis represents Cooperation with the BA from Low to High and Y-axis represents the Assertiveness(domain Knowledge) from Low to High

**Low Co-Operation and Low-Assertiveness**-Avoidance-Means ignoring the conflict.

**High Co-Operation and Low-Assertiveness**- Accommodation- The stakeholders will prioritize their needs over others.

**Low Co-Operation and High Assertiveness-** Competition-means defensive, that is standing for individual believes and trying to win.

**High Co-Operation and High Assertiveness-** Collaboration-Working together for finding solution.

**Medium Co-Operation and Medium Assertiveness-**Compromise-Where we try to compromise on a middle ground.

This method helps us understand where the stakeholder is in the current topic their knowledge on the topic and decide where they stand in this graph.

If they are in Avoidance, Accommodative or Competition mode we should try to convince them and bring them to Collaborative or Compromise mode.

High

**Assertiveness**

Low

 Low **Co-Operation** High

**Q10. List down the reasons for project failure**

**Answer**

1. **Improper Requirement Gathering:** During the Requirement gathering state if all the requirements are not understood and documented correctly it can lead to project failure.
2. **Continuous change in requirements:** If the requirements keeps changing frequently it will hinder the scope of the project which can lead to project failure.
3. **Lack of Stakeholder Involvement:** The Stakeholders inputs and feedback play a very important role in the project. If they do not actively participate in this process it can lead to project failure.
4. **Lack of Executive support:** If the Higher management doesn’t support in taking any major decisions in the project and do not negotiate with the clients can lead to project failure.
5. **Unrealistic expectations:** If the project goal is unrealistic that a huge task has to be completed within a small amount of time can cause project failure.
6. **Improper Planning:** If there is no proper planning regarding the project milestones and goals to be achieved can lead to project failure.
7. **Inadequate Risk Management:** If the project has failed to identify all the risks that can happen in the future and has not a solution for the same can cause a project failure.
8. **Poor Communication:** Communication is the key for a good project. If there is a gap in communication between the team members regarding the process or flow of the project can cause project failure.
9. **Scope Crepe:** When a project’s scope grows beyond what was originally agreed upon can cause project failure.
10. **Resource Constraints:** If there are less resources and huge number of tasks the timelines of the project cannot be met leading to project failure.
11. **Technical challenges**: It there is any system downtime, server down time or portal blocked due to password issues or no trained developers are available can cause project failure.
12. **Budget Constraints:** If the project wants more resources or the time for the project has been extended but there is no sufficient budget for the project can lead to project failure.

**Q11. List the Challenges faced in projects for BA**

**Answer**

1. **Lack of Domain Knowledge:** It is difficult for a BA to have knowledge on all the Domains which they have no experience.
2. **Lack of training:** If there is no proper training given if it a new domain or a new project it will be difficult for the BA
3. **Changing Requirements:** If the requirements keeps changing throughout the project it will be difficult for the BA to understand and document the processes.
4. **Lack of Stakeholder Involvement:** It is very important for all the stakeholders to be in the same page. It they do not turn up to the meetings and provide approvals it will be very difficult for the BA to move to the next steps.
5. **Unclear project objectives:** It is very important that each and every requirement with clear definition is provided to the BA.
6. **Managing conflicts and negotiation**: If any stakeholder is against any decision of the project the BA will have a tough time to make them accept to a correct sequence.
7. **Project Communication:** If there is no communication regarding any issue or failures in the project timeline not communicated to BA it will casus issues.
8. **Time and Resource constraints:** The project should provide a exact timeline for each requirement and correct number of resources to achieve a task.
9. **Quality Assurance and Testing:** It is the task of the BA to assist in UAT testing. At that time to provide correct guidance to users it is very important that the BA has a clear knowledge on each and every step that are followed.
10. **Documentation Management:** All the documents should be updated and reviewed always.
11. **Technology Constraint:** It is not necessary that a BA will be aware of the technologies

**Q12. Write about Document Naming Standards**

**Answer:**

The naming standards vary from project to Project. But there are some points that we should consider while naming the documents

1. All documents will be named with some standards [ProjectId][Document Type]V[X]D[Y].ext.
2. The Filenames should be short and meaningful
3. Avoid unnecessary redundancy and repletion in the file names, File path and folders.
4. Version numbers are very important for each document.
5. Avoid using special symbols and spaces in the naming format.
6. Before saving check for the document is saved with the correct extension.

**Q13. What are the Do’s and Don’ts of a Business analyst**

**Answer**

**Do’s of a Business Analyst:**

1. **Actively engage with stakeholders**-Build strong relationships with all relevant parties to understand their needs and concerns thoroughly.
2. **Clearly document requirements-**Precisely capture business needs and translate them into detailed specifications to avoid misunderstandings.
3. **Prioritize requirements-**Identify the most critical needs and focus efforts on addressing them first.
4. **Communicate effectively-**Clearly articulate findings, solutions, and potential risks to all stakeholders, ensuring everyone is on the same page.
5. **Analyse data thoroughly-**Utilize data analysis techniques to identify patterns, trends, and areas for improvement.
6. **Stay adaptable-**Be prepared to adjust plans and requirements as the project evolves and new information becomes available.
7. **Utilize visualization tools-**Create diagrams, flowcharts, and other visuals to effectively communicate complex concepts.

**Don’ts of a Business Analyst:**

1. **Ignoring stakeholder needs-**Neglecting to actively listen to stakeholders and address their concerns can lead to issues later on. Never say “NO” to Client.
2. **Make assumptions about requirements-**Do not assume you understand needs without proper clarification and validation.
3. **Fail to prioritize effectively-**Not prioritizing requirements can result in wasted time and resources on less important issues.
4. **Poor communication-**Lack of clear and concise communication can lead to confusion and project delays.
5. **Overlook data quality-**Rely on unreliable or inaccurate data when making decisions.
6. **Become overly technical-**Focus solely on technical details without considering the broader business impact.
7. **Resist change-**Be inflexible and not adapt to evolving project needs or changing market conditions.

**Q14. Write the difference between packages and sub-systems**

**Answer**

**Package:**  A Package is a grouping mechanism for related code elements within a system like Class and Interface. It is a collection of components which are not reusable in nature. Application development companies work on Packages.

**Sub-System:** A Sub-System represents a large, more functional unit within the system, often composed of multiple packages with a distinct responsibility and a set of functionalities within overall system architecture. It is a collection of components which are reusable in nature. Product Development companies work Sub-Systems.

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

**Answer**

Camel-Casing refers to the naming convention variables, parameters, documents and properties. Multiple words are joined together. In Camel-Causing the first letter of the string starts with small letter and other words first letter starts with capital letters. It is used in many programming languages that doesn't allow spaces in file names. CamelCase enables the creation of names that are more unique and have more meaning for the developer. It is commonly used in web URLs, programming and computer naming conventions. BA usually uses it while writing Use Cases, User Stories and requirements gathering.

**Q16. Illustrate Development server and what are the accesses does business analyst has?**

**Answer**

A Development Server refers to a dedicated environment that is used during the software development process. It is platform that is used by developers and testers to develop, test, debug the applications. It comprises of all the essential hardware, software, and other components used to deploy and test the software under development, including bulk storage, development platform tools , network access and high end processors. Once the development is completed the testing is done in the development server and is further moved to UAT servers or directly to the Production servers.

Accesses provided for the BA’s in Development Servers:

Read-Only-Access- Theya re not allowed to make any code changes.

Limited access will be provided to specific modules.

Limited Configuration Access: Only certain places will be given access for configuration changes for BA.

**Q17. What is Data Mapping**

**Answer**

Data Mapping is the process of matching fields from one Database table to another. It is the first step to facilitate data migration, data integration and data management tasks. Data Mapping is an essential part of ensuring that in the process of moving data from a source to a destination data accuracy is maintained. It involves identifying the source data elements, determining their meaning and structure and mapping them to the corresponding target data elements. Good data mapping ensures good data quality in Data Warehouse.

**Data Migration:** While migrating the data from the source system to the destination system, the data elements are mapped accurately into the new system. Required techniques are applied to covert the data into the format that is required by the new system.

**Data Integration:** While combining the data from different sources, it ensures that the data is properly matched.

**Data Transformation:** Data transformation means converting the data from one format to another.

**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**

**Answer**

API stands for Application Programming Interface. It is asset of rules and protocols that allow different software applications to communicate with each other. API allows sharing of only necessary information and keeps the internal system details hidden, which helps the system security.

For the above scenario,

**Establish API communication**- set up API communication between your application and other application to exchange data.

**Data formatting**- while sending the data from one application to other, convert the date format from dd-mm-yyyy to mm-dd-yyyy. While receiving the data from other application, parse the data and extract the date, month and year and re-arrange them accordingly.

**Data Validation**- Perform data validation and ensure that the converted date remains in valid format.