**CAPSTONE PROJECT PREPARATION-3-PART-1/2**

Q.1) Draw Use case diagram?

Ans.1**) (In next Page)**



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

Ans.2)

**Boundary Classes**:

|  |  |
| --- | --- |
| Boundaries are objects that interface with system actors: user-interface, database gateway, server etc.  Here system actor should be primary actors  Primary actors are those who initiate the use case and interact with the system | Customer Registration  Customer login  Bank server login  Customer logout  Bank server logout |

**Controller Classes**:

|  |  |
| --- | --- |
| Controller class handles user input and process the data. It mediates between the boundaries and entities  Use case will be considered as the controller class. | Registration controller  Login controller  Payment controller  Credentials  Net banking  Email  Logout |

**Entities**:

|  |  |
| --- | --- |
| Entities are object representing system data (All actors)  All actors will be considered as one entity | Customer  Bank server  Database |

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

Ans.3)

|  |  |
| --- | --- |
| **Application Layer** | Customer Registration  Customer Login  Bank Server login |
|  | |
| **Business Logic Layer**  **(Primary actors associated with boundary class)** | Customer  Bank Server |
|  | |
| **Data Layer(All the entity class)** | Customer  Bank server  Cash  Card  Net banking |

In the Three-tier architecture, the application layers handle the user interface, The business logic layer manages the business logic and coordinates between the other tiers, The data tier handles data storage and retrieval.

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

Ans.4) Domain model is very much similar to Entity-Relationship diagram, where tables are connected to each other.

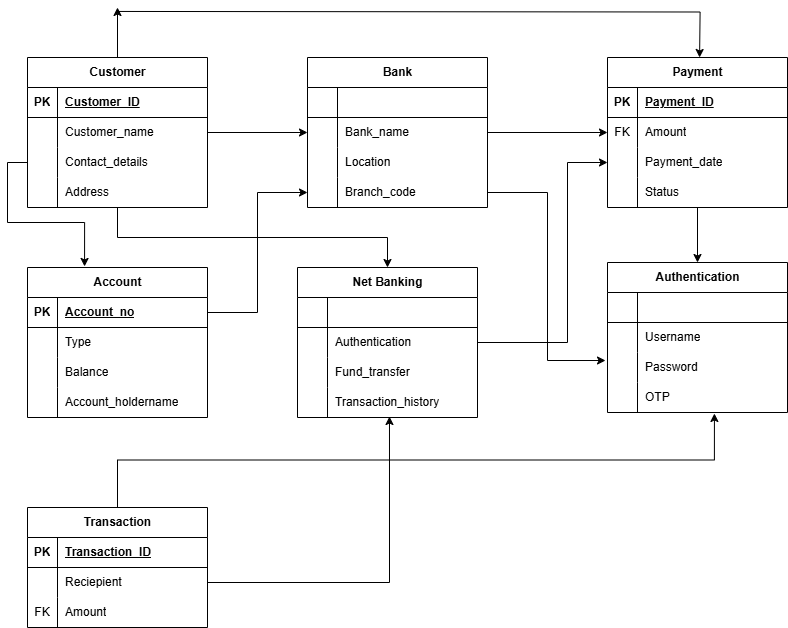
In below diagram, Customer ID is connected to bank, Account, Payment because he has an account in the bank and he has to do the payment.

Payment table is connected to the bank, Authentication, Net banking as when a customer makes a payment by Net banking, authentication must be done.

The account is in the bank so the account table is connected to the bank table.

Authentication table is connected to the Transaction table as when payment is done the transaction Id is there.

**ER diagram do not have attributes inside the box while the domain models have attributes mentioned inside the box**.

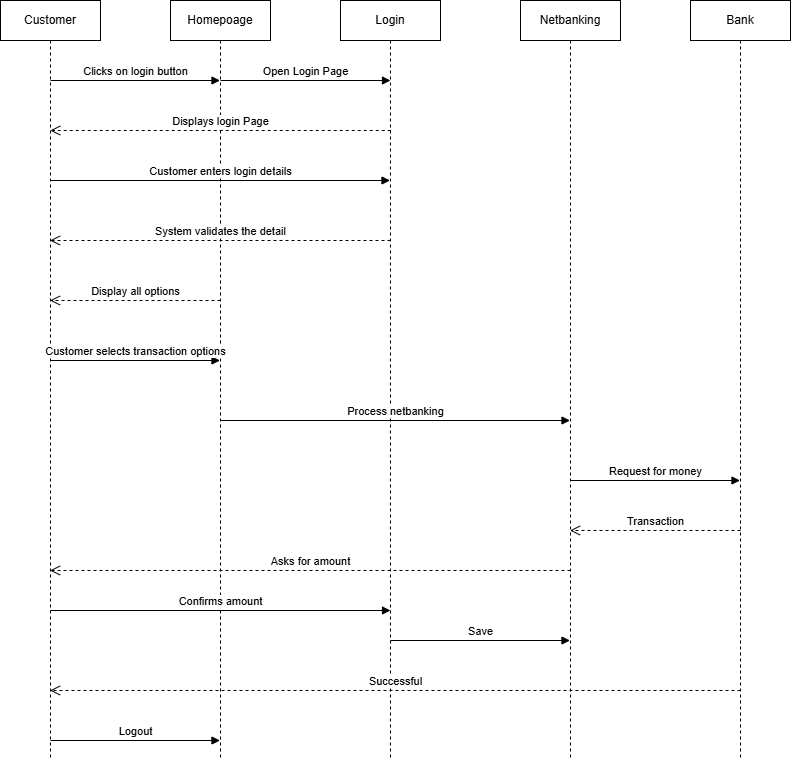


Q.5) Draw a sequence diagram for payment done by Customer Net Banking?

Ans.5) Sequence diagram shows how the objects in the system interacts and communicate with each other with time to achieve tasks.

It is used to show the flow of messages, events or actions between the objects of the system. It helps to visualize the behavior of the system.

This diagram shows the process in detail.



Q.6) Explain Conceptual Model for this Case?

Ans.6) The conceptual model helps in understanding the key concepts, their relationships, and overall structure of the net banking payment system.

It serves as a foundation for designing the database schema, defining the application architecture and implementing the necessary functionalities within the system.

The relationships among these entities can be described as follows

1. Customers: This node represents the customers or users of net banking services.
2. Service awareness: - Customers must be aware about the service of the net banking system and their features.
3. Privacy data: - The importance/significance of this node is to protect the privacy and confidentiality of customer data in the context of net banking.
4. Technology awareness: - This significance of the node is that customers should be aware and comfortable with underlying technology used in the net banking.
5. Trust & Support: - This node indicates that the bank provides such good services that will help to enhance customer trust.
6. Bank: - This node represents the service provider responsible for offering net banking services.
7. Online information: - This aspect highlights the importance of providing accurate and up to date online information about net banking services to customers.
8. Security & Privacy: - The bank should adapt the security policies which will help the customers to keep their data related to the transaction secure and private.
9. Infrastructure: - This component suggests that the underlying technologies infrastructure, including hardware and software systems.
10. Policies:- This node represents policies and regulation that govern the implementation and operation of net banking services, ensuring compliance and customer protection.

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

Ans.7) MVC is design pattern where, the application is divided in three logical Parts-Model, View and controller.

Each of these parts has different responsibility.

**Model**:

The Model represents the data and business logic of the application. Model is responsible for multiple tasks like managing data, performing data, validation, implementing business rules and handling data access operations.

Model does not depend on how the data is presented or how the users interacts with the application.

The Model class is known about all the data that is needed to be displayed. This layer corresponds to the data related logic that users work with. It represents the data that is being transferred between View & Controller. It can add or retrieve data from database.

It responds to the controller request because the controller cannot interact with the database itself. The Model interacts with the database and give the requested data.

All the Model classes are nothing but entities class. Model class is represented by entity class.

**View**:

The View is responsible for presenting the data to the user. It handles the user-interface. The View can be a webpage, a desktop application window or any other form of user interface. It receives input from the user and passes it to the controller for processing. It represents the presentation of the application. It takes the data from the Model and renders it in a way that is suitable for the users display or interaction.

For rendering the data, it is uses query method. View doesn’t depend upon application logic. View class are represented as boundary class.

**Controller**:

The Controller acts as an intermediary between the Model and the View. It receives input from the user (via the view), processes the input by invoking the appropriate methods in the model and updates the view with new data or state.

The Controller handles user interactions, interprets user input and translates it into instructions for the Model or the View. It coordinates the flow of data between Model & View, ensuring that they remain separated and independent of each other.

Whenever user requests for anything, then request first goes to the controller, Controller works on the user request.

Takes input from the client/user

It interacts with Model & View

Controller class represents use case

Controller class acts as mediator between model & database.

Controller cannot get data directly from the database, so controller interacts with model.

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

1. Combination of one actor and one use case results in one boundary class.

Combination of two actors and one use case results in two boundary class.

Combination of three actors and one use case results in three boundary class.

1. Use case will result in controller class.
2. Each actor will result in one entity class.

**Example**: Online shopping application with the following use case: -

**Model Class**:

Customer, Payment, Net-banking, Card, Cash

**View Class**:

Login view, Payment option view, Net banking, Bank selection view, Credentials, Payment amount view, Payment confirmation view, Logout view.

**Controller Class**:

Login controller, Payment option controller, Net banking controller<bank selection controller, Credentials controller<payment amount controller<Payment confirmation controller, Logout controller.

**Guideline to place classes in three tier architecture**:

**Presentation Layer**:

The layer is nothing but user interface.

View is inside this layer

This class directly interacts with the user.

Presentation layer is responsible for displaying information and also receiving the input from the user.

**Application Layer**:

This layer is nothing but the business logic.

Model and controller are inside this layer.

Controller handles user input, process the request and coordinates interaction between the model and view.

**Data Layer**:

Classes those are responsible for data access and storage are in this layer.

It contains the classes which interacts with database, files and other data sources.

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

Ans.8) Waterfall model is useful in the situation where the project requirement are clear and well defined. It follows a sequential approach. In this model each phase is completed entirely and then moved to the next phase.

**Requirements Gathering**:

First the stakeholders are identified

BA and Project owner participates in this phase. After completing the phase BRD will be generated.

**Requirement Analysis**:

The requirements are analyzed to understand the scope of the project.

BA will check all the requirements and if he found any conflicting requirements then the BA will talk to the concerned stakeholders and will ask to clear it.

BA will prepare functional requirements called(FRS)- Functional requirement specification

Technical team will prepare the non-functional requirements called(SSD)-Supplementary support document.

BA will combine FRS & SSD to form SRS (Software requirement specification)

BA will prepare RTM based on SRS.

**Design**:

After the requirements are completed design phase starts. It contains detailed design document that outlines the software architecture, user interface and system components.

HDD (High level design document), ADD (Application design document)

BA collaborates with designers, architects and developers to translate requirement into system design.

BA ensure the design meets the stakeholders needs.

**Development**: -

This phase includes writing software codes based on design specification.

Developers and programmers are involved in these phase.

BA acts a mediator between development team and stakeholders.

BA clarifies the requirement and participates in the Scrum meeting.

**Testing**:

In the testing phase, software is tested to ensure it is free from defects.

BA facilitate UAT. And helps user to know the functionality of the system

**Deployment**:

BA ensures the smooth transition of application from development to deployment phase while getting in production environment.

**Implementation**:

While code is run for first time in this phase, BA will update the documentation and requirement specification to reflect changes over time.

Maintenance:

It is done by support team.

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

technique?

Ans.9) Conflict can occur due to various reasons such as differences in goals, values, personalities, resources or communication breakdown. It is important to resolve these conflicts to promote growth and learning and achieve project goal.

Conflict management is nothing but the process of identifying and resolving conflicts in a healthy and constructive manner.

It consists of strategies and techniques aimed at resolving conflicts, disputes, disagreements or differing perspective among individuals or groups.

Hence it can be said as process or skill aimed to find creative ways to handle the disagreement.

Thomas-Kilmann technique approach is widely used to recognize the approaches for conflict management.

**High Assertiveness and High Cooperativeness**- Collaboration means working together to find solution.

**High Assertiveness and Low Cooperativeness**-Competition means defensive, that is standing for your individual

**Low Assertiveness and High Cooperativeness**-Accommodation-Stakeholder will prioritize their needs over others.

**Low Assertiveness and Low Cooperativeness**-Avoidance-means ignoring conflict.

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

Ans.10) Reasons for Project failure are:

**Improper requirement gathering**:

If the requirements of the project are not properly gathered correctly, then this can lead to project failure.

**Lack of Stakeholder requirement**:

A project can fail, If the stakeholders are not participating in the process.

**Less Communication**:

If there is misunderstanding between stakeholders and team members due to ineffective communication this can lead to project failure.

**Continuous change in the requirements**:

If the requirements keep on changing frequently this can lead to project failure, because the scope of the project will keep on changing.

**Poor risk management**:

Poor risk management can lead to project failure.

Lack of user involvement

Lack of executive support.

**Unrealistic expectations**:

If the goals given are out of scope, then this can lead to project failure.

Improper Planning:

Project can fail if planning is not done properly, milestones and goals can be discussed.

**Insufficient resources**:

Project can fail due to lack of technical knowledge and lack of finances.

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

Ans.11) Here are some challenges faces by BA:

Lack of training

Obtaining sign-off on the requirement

Change management

Co-ordination between developers and testers.

Conducting meeting

Making sure status report is effective

Driving clients for UAT completion

Making sure that project is going on the right track and delivered as per timelines.

Gathering clear requirements can be challenging

Scope creep: Change in requirements and scope during project lifecycle can lead to scope creep.

Managing the stakeholder with conflicting interests

Lack of domain knowledge

Poor communication between stakeholders and BA can affect the process of gathering the requirements.

Technical complexity

Q.12) Write about Document Naming Standards?

Ans.12) All documents will be named using some standards like

[Project ID] [Document Type]V[x]D[y].ext.

Example: -PQ786BRDV1D2.docx

PQ786BRD1.2.docx

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

Ans.13)

Never Say ‘NO’ to the client

There is no word called as ‘BY DEFAULT”

Never imagine anything in terms of GUI

Question everything in the world

Go the Client with plain mind i.e with no assumption

Listen to the client very carefully and after he is done, then ask question.

Don’t interrupt the client

Never try to give solution the client right away

Try to concentrate only on important and required things.

Be like lotus in the mood-If the client tries to give fancy requirements then talk to the project manager first.

Requirement Hurried-Project buried.

Never criticize the stakeholder.

Always appreciate the stakeholder even for small efforts.

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

Ans.14)

**Packages**: - It is group of classes or use case that are used to organize model elements.

Packages can be nested with other packages.

These are used as containers to organize elements.

It is very useful to represent system architecture.

**Subsystem**: - It is logical grouping of related components.

It is a collection of classes, packages, libraries and other sub system that work together to deliver a specific set of functionalities.

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

Ans.15) Camel casing refers to the naming convention of variable, parameters or properties. Here, multiple words are combined together

In camel casing, the starting letter starts of the first word starts with small letter and other words first letter starts with capital letter.

Ex: firstName, lastName

Camel casing is used for requirements documentation, In req documentation, BA often use camel casing to name the entities like use case, features, user stories etc.

The database tables name also uses camel casing.

Camel casing is used requirement document also, to name the functional or non-functional requirements.

By using camel casing in the documents, it helps maintain consistency in the entire document and also increases readability.

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

Ans.16) A development server refers to a dedicated environment that is used during the software development process.

It provides platform for the developer and testers to build, test, develop and debug the application.

The accesses BA has:

Read-only-BA may be granted access to the development server. This will allow them to use user interface of the application, navigate through the features and also they will be able to observe the behavior of the application.

Limited access:

Depending on the project needs, the BA will be granted limited access to the specific modules in the application.

Limited configuration access: - Means have the authority to make changes in certain areas of application where they have the access.

Q.17) What is Data Mapping?

Ans.17) The databases contains multiples tables in it.

There may come a scenario, where we need to map the data from one table to another.

Hence Data mapping is a process to establish connection between multiple data sources.

The purpose of the data mapping is to ensure that the data is accurately transferred or converted into different format.

The main purpose of data mapping is:

**Data integration**: -

While combing the data from different sources, it ensures that the data is properly matched.

**Data migration**: -

While migrating the data from legacy system(source) to the view system(destination), the data elements are mapped accurately into the system.

Required techniques are applied to convert the data into format that is required by the new system.

**Data transformation**: -

Data transformation means converting the data from one format to other.

In data mapping data mapping plays very important role which ensures that data of the legacy system(source) is mapped correctly to the data in the new system(destination).

Q.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?

Ans.18) API stands for Application Programming Interface.

It is a software intermediary that allows the two applications to communicate with each other.

It is the set of rules, protocols and tools that define how different software application should interact 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.

Do 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.

Perform Data Validation and ensure that the converted date remains in a valid format.

………………………………………… End……………………………………………………