Capstone project 3

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



Q2. Derive Boundary Classes, Controller classes, Entity Classes. - 4 Marks

Answer: Q2. Derive Boundary Classes, Controller Classes, Entity Classes (4 Marks)

Boundary Classes:

Payment option boundary

Card payment boundary

Wallet payment boundary

Cash payment boundary

Net Banking payment boundary

Controller Classes:

PaymentController: Manages the payment process.

Entity Classes:

Customer: Represents a customer.

Payment: Represents a payment transaction.

CardDetails: Details of card payment.

WalletDetails: Details of wallet payment.

CashDetails: Details of cash payment.

NetBankingDetails: Details of net banking payment.

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

Answer

|  |  |
| --- | --- |
| Application Layer | Customer RegistrationCustomer LoginBank Server Login |
| Business Logic Layer | Payment Controller |
| Data layer | Customer Payment Card Details Wallet Details Cash Details NetBanking Details |

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

Answer

A Domain Model represents the entities, their attributes, and the relationships between them in a specific context. For a customer making a payment through Net Banking, the domain model outlines the key entities involved and how they interact to facilitate the payment process.

Key Entities:

1. Customer
2. Attributes
3. Payment
4. Attributes
5. NetBankingDetails
6. Attributes
7. PaymentMethod
8. Attributes
9. Account
10. Attributes

Relationships:

Customer initiates a Payment.

Payment is processed through a specific Payment Method (Net Banking).

Payment includes NetBankingDetails.

Customer has an Account used for the payment.

|  |
| --- |
| customer |
| Customer IDNameEmailPhonenumberprocessed through |

|  |
| --- |
| Payment |
| Payment IDAmountDateStatus |

initiates

|  |
| --- |
| Netbanking |
| Transaction IDAmountBank nameAcc numIfsc code |

|  |
| --- |
| Net banking details |
| Bank detailsAccount numberIFSC code |

|  |
| --- |
| Payment method |
| Method idMethod name |

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

Answer

Customer

Homepage

Loginpage

Net banking

Bank

Open Login page

Clicks on login button

Display Login page

Customer enters login details (username and password)

System vailidates the details

Display all options

Customer selects transaction option

Request for money

Process

Log out

Successful transaction

Save

Confirm amount

Ask for money

Transaction

Q6. Explain Conceptual Model for this Case

Answer

A Conceptual Model provides a high-level overview of the key entities and their relationships in a system, without delving into technical details. For the case of a customer making a payment through various methods (Card, Wallet, Cash, Net Banking), the conceptual model identifies the main components and their interactions.

Key Components: Customer,Payment, Payment Methods, PaymentDetails

Relationships:

1. Customer initiates a Payment.
2. Payment uses one of the PaymentMethods.
3. PaymentDetails are associated with the Payment to provide the necessary information for the selected payment method.
4. Customer has an Account used for making payments through Net Banking.

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:

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible project.

Components: model, view, controller

MVC Rules to Derive Classes from Use Case Diagram

1. Identify Use Cases
2. Derive Model Classes
3. Derive View Classes
4. Derive Controller Classes

Guidelines to Place Classes in 3-Tier Architecture

Presentation Layer (User Interface)- view classes

Business Logic Layer- controller classes

Data Access Layer- model classes

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

Answer:

A waterfall model is very old and traditional model in IT industries. It is a progressive

implementation of the projects which is divided into different phrases of SDLC.

Business Analyst will verify if the product is delivered as per the requirements, and it is meeting the business need. Maintenance: Once the implementation is done the team has to give support by installing patches, handling change requests, etc

Stages in Waterfall Model are as follows:

Requirement Gathering and Analysis, Designing, Coding, Testing, Deployment, Maintenance

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

Answer:

Conflict management is the use of techniques to resolve disagreements or control the level of discord. Conflict occurs whenever people disagree. The disagreement could be over their perceptions, ideas, values, motivations, or desires. Researchers Kenneth Thomas and Ralph Kilmann developed a model for resolving conflicts. This model is known as the Thomas-Kilmann model.

This model is based on two dimensions of conflict management: Assertiveness and Empathy. Based on these two dimensions, there are five conflict resolution strategies: Competing, Avoiding, Accommodating, Collaborating and Compromising.

Q10. List down the reasons for project failure

Answer:

1. Poor planning
2. Inconsistently defined resources
3. Unclear objectives
4. Lack of detail control
5. Lack of transparency
6. Lack of communication
7. Change of direction
8. Unrealistic expectations
9. Lack of monitoring
10. Unrealistic due dates

Q11. List the Challenges faced in projects for BA

Answer

*  Lack of training
*  Obtaining Sign-off on requirement
*  Coordination between developers and testers
*  Conducting meetings
*  Preparing effective status reports which satisfies all the project stakeholders.
*  Driving client for UAT completion
*  People Management (Coordination between different people and different teams)

Q12. Write about Document Naming Standards

Answer:

File Naming Standards are used to save the file with particular name or format. This is important in Sharing and keeping track of data files.

Following are the best standards in Naming Convention.

1. It should be Named Consistently.

2. File names should be short (<25 characters)

3. Avoid special characters or spaces in a file name.

4. Use Capital and Underscores instead of spaces or slashes.

5. Use date format as per ISO 8601: YYMMDD

6. Include a version number.

7. Write down naming convention.

We must consider following naming conventions:

* Date of Creation
* Short Description
* Work
* Location
* Project name or number
* Sample
* Analysis
* Version Number

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

Answer:

* Never Say No to client
* Never Imagine anything in terms of GUI
* Question Everything
* Consult an SME for clarifications in requirements
* Every problem of client is unique
* Do not interrupt the client, when he/she is giving you the problem
* Should not be hurry
* Should know what the Scope of the Project is

Q14. Write the difference between packages and sub-systems

Answer

|  |  |  |
| --- | --- | --- |
| Aspect | Packages | Sub-Systems |
| Definition | Grouping of related classes/interfaces | Larger component encompassing multiple packages |
| Purpose | Organize and modularize code | Break down complex systems into manageable parts |
| Scope | Smaller, focused scope | Broader, major functional areas |
| Example | java.util, java.io packages | user Management, Order Processing sub-systems |
| Structure | Contains classes, interfaces, sub-packages | Composed of multiple packages and components |
| Encapsulation | Encapsulate related elements | Encapsulate significant functionality |

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

Answer:

Camel-casing is a convention in programming where the first letter of the first word is written in lowercase, and the first letter of each subsequent word is written in uppercase, with no spaces between words. This style helps improve readability and distinguishes individual words in a variable or function name.

Examples of Camel-Casing:

* userName
* firstName
* accountBalance

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

Answer: A development server is dedicated environment or server that is used during software development process. It provides a platform for developers and testers to build, test and debug applications before they are deployed for production environment. As a BA, we have only limited access only.

Q17. What is Data Mapping 6 Marks

Answer: Data Mapping is the process of connecting data fields from one system or database to corresponding fields in another system or database. This ensures accurate transfer and consistency of data between different systems.

Key Components:

Source and Target: Identifying data fields in the source system and matching them with corresponding fields in the target system.

Transformation Rules: Applying any necessary transformations to the data to ensure it fits the format and requirements of the target system.

Consistency: Ensuring that data remains consistent and accurate across different systems.

Automation: Using tools and software to automate the mapping process, reducing errors and saving time.

Validation: Verifying that the mapped data is accurate and meets the requirements of the target system.

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 (Application Programming Interface) is a set of rules, protocols, and tools that allows different software applications to communicate with each other.

Using API Integration to Handle Different Date Formats:

1. Hire an API integration developer

2. Create a project within the API provider system

3. Receive the API key and authorization token

4. Integrate the API framework for the app

5. Use API request instances and methods