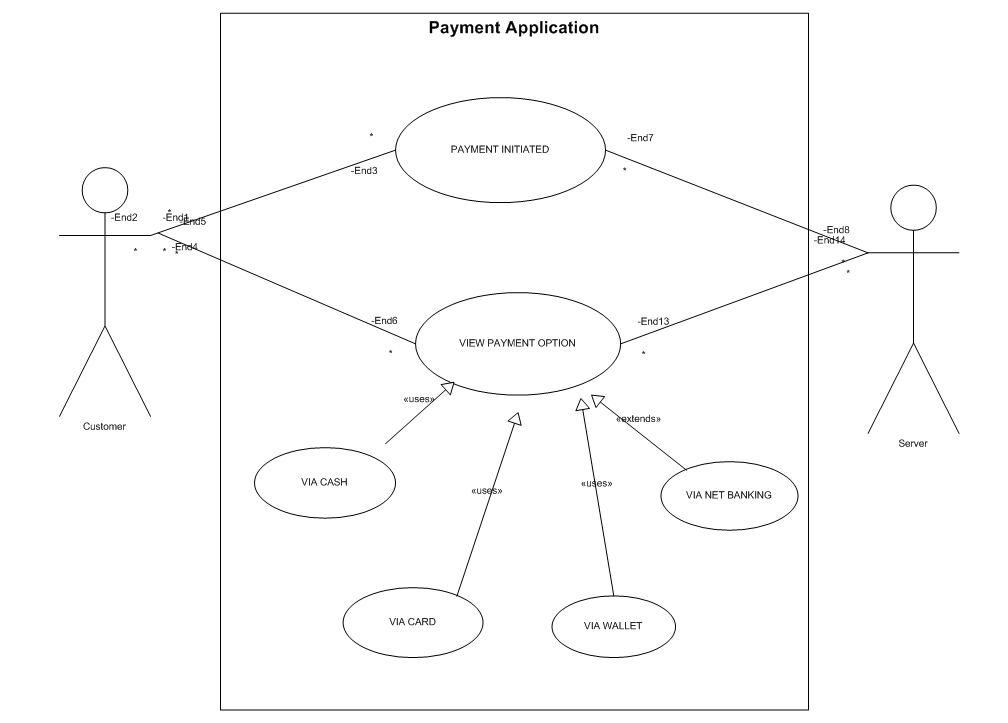
**CAPSTONE PROJECT -3**

1AnS. USE CASE DIAGRAM:



**2Ans.**

|  |  |
| --- | --- |
| **Boundary class – (Use Cases) Actors speak to the system**  **(Authentication information)** | **Customer registration** |
| * Combination of 1 actor and a use case 1 boundary class * Combination of 2 actors and a use case 2 boundary class * Combination of 3 actors and a use case 3 boundary class   All the actors should be – (Primary Actors)  Primary actors- Who initiate the use cases and interact with the  system | * Customer login * Bank server logs in * Customer logout * Bank Server Logs out |

|  |  |
| --- | --- |
| **Controller class – (Handles Users (Primary actors)**  **input and processes the data** | **Registration Contorller** |
| •Use case will consider as Controller class  system | •Login Controller  •Payment Controller  •Credential controller  •Net Banking controller  •Logout controller |

|  |  |
| --- | --- |
| **Entity Class – All Actors** |  |
| •Each actor will be considered as an  entity | •Customer  •Bank server  •Cash  •Card  •Net Banking |

**3Ans.**

|  |
| --- |
| **USER LAYER** |
| Payment Method selection boundary |
| Card Payment boundary |
| Wallet Payment Boundary |
| Cash payment Boundary |
| Net banking Payment boundary |
| **BUSINESS LOGIC** |
| 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) |

**4Ans.**

**Domain model** is similar to the entity relationship model. The tables are connected to each

other. In the below diagram, the customer table is connected to bank table, which is why

the customer is able to make payment.

Customer table is also connected to payment table, because he should make the payment. Now the

payment is done by net banking, so payment table is connected to net banking table.

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

table is connected to both net banking table and bank table, because authentication is to be performed there.

Also, the authentication table is connected to transaction table, because authentication will be done while transaction.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CUSTOMER** | | | | |
| CUSTOMER ID | CUSTOMER NAME | CONTACT DETAILS | ADDRESS | ACCOUNT NUMBER |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **BANK** | | |
| BANK NAME | LOCATION | BRANCH CODE |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **PAYMENT** | | | |
| PAYMENT ID | AMOUNT | PAYMENT DATE | STATUS |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **ACCOUNT** | | | |
| ACCOUNT NO | ACCOUNT TYPE | BALANCE | ACCOUNT HOLDER NAME |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

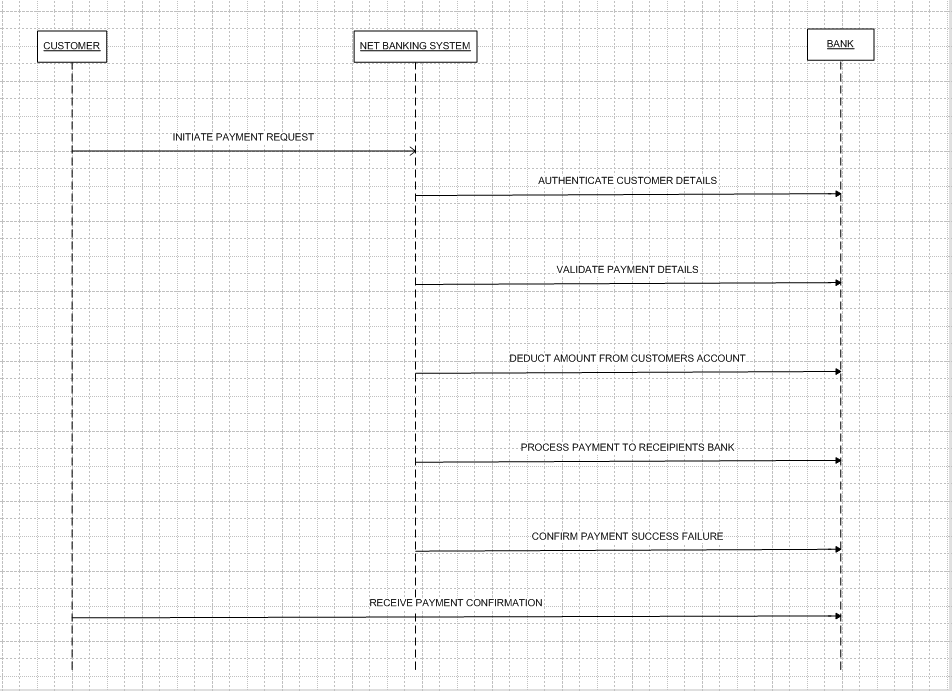
|  |  |  |  |
| --- | --- | --- | --- |
| **NET BANKING SERVICE** | | | |
| AUTHENTICATION | FUNDS TRANSFER | TRANSACTION HISTORY | ACCOUNT MANAGEMENT |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **AUTHENTICATION** | | | |
| USER NAME | PASSWORD | OTP |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **TRANSACTION** | | | |
| TRANSACTION ID | RECEIPIENT DETAILS | AMOUNT | TIMESTAMP |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

5Ans.

**Sequence diagram for payment done by Customer Net Banking:**



6Ans.

**CONCEPTUAL MODEL:**

The conceptual model helps in understanding the key concepts, their relationships, and the

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 between these entities can be described as follows:

1.Customer: This node represents the customers or users of net banking services.

2.Service awareness: Customers should be aware of the available net banking services

and their features.

3.Privacy of 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: The significance of this node is that customers should be aware

and comfortable with the underlying technology used in net banking services.

5.Trust & Support: This node indicates that the bank provides such good services that it will

help to enhance the customer’s trust.

6.Bank: This node represents a 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 their transaction secure and private.

9.Infrastructure: This component suggests that the underlying technological infrastructure,

including hardware and software systems, plays important role in enabling net banking

services.

10.Policies: This node represents the various policies and regulations that govern the

implementation and operation of net banking services, ensuring compliance and customer

protection.

**7Ans.**

**MVC** is a design pattern where, the application is divided into 3 logical parts-

Model, View and Controller.

Each of these parts will have specific responsibility.

**MODEL:** The model class is known about all the data that is needed to be displayed. This layer corresponds to the data-related logic that the user works with. It represents the data that is being transferred between View and Controller. It can add or retrieve the data from the database.

**VIEW:** The View is responsible -for presenting the data to the user for handling the user interface. The View can be a web page, 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. View refers to the model. It takes the data from the Model and renders it in a way that is suitable for the user's display or interaction.

**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 then updates the View with the new data or state.

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 actor and one use case results in two boundary class.
* Combination of three actor and one use case results in three boundary class.

2. Use case will result in controller class.

3. Each actor will result in one entity class.

Guidelines to place classes in 3-tier architecture-Presentation Layer-This layer is nothing but a user interface. View is inside this layer

Tier architecture: - has 3 layers

* Application layer
* Business logic layer
* Data base layer

Data flowing from the 3 layers explains the 3-tier architecture where the information is passed. Here in elicitation techniques 3 tier architecture works as a medium where as a BA we get to know the information shared by the requester will it fit in application layer, business logic layer or data base layer.

**8Ans.**

**Waterfall model** is useful in the situation where the project requirements are well defined and

the project goals are clear.

•Waterfall model follows sequential approach.

•In this model each phase is completed entirely and then only moved to the next

phase.

•Waterfall model relies on documentation to ensure that the project is well defined and

project team is working toward clear goals.

•Ones that particular phase has been completed and ones we move to the next

phase, we cannot go back to the previous phase to make changes.

•This model is stable for the projects when the requirements are clear.

Requirements Gathering-

First, the stakeholders are identified. In this phase, all the requirements are gathered from

the stakeholder.BA and Project Manager participates in this phase. After completing this

phase, BRD will be generated.

**Requirements Analysis**-The requirements are analysed to understand the scope of

the project. Analysing means the BA will check all the requirements, if he founds convincing

requirements then the BA will talk to the concerned stakeholder to clear it, remove the

ambiguous requirements.

**Design-**

After the requirements are cleared, Design phase starts. This has a detailed design

document that outlines the software architecture, user interface, and system components’,

ADD and solution document will be generated here. [High-level Design Doc.]

BA Collaborate with designers, architects, and developers to translate requirements into

system design.BA Ensure that the design aligns with the documented requirements and

addresses stakeholder needs.

**Development-**

The Development phase include implementation. It involves coding the software based on

the design specifications. Programmers or developer are involved in this phase. Here BA

acts as a mediator between the development team and the stakeholders.BA clarifies the

requirements, check if the development is going on right track or not.BA also participates in

scrum meetings.

**Testing-**

In the testing phase, the software is tested as a whole to ensure that it meets the

requirements and is free from defects. Testers are involved in this phase

Test documents are generated here.BA works with the testing team to ensure that the

solution meets the requirements.BA facilitate UAT.BA helps the users to know the

functionality of the system and also helps them to use the system.

**Deployment-**

Once the software has been tested and approved, it is deployed to the production

environment.BA ensures that there is smooth transition from development phase to the

production phase.

**Implementation-**

This is the final stage of waterfall model. It involves running the code for the very first time in production phase. Release manager handles this phase.

BA will Update documentation and requirements specifications to reflect changes in the

system over time Maintenance-Running the code for second time in the production phase is called maintenance. This is done by support team.

**9Ans. Conflict management:**

Conflict management is nothing but the process of identifying and addressing conflicts in a

healthy and constructive manner. It consists of strategies and techniques aimed at resolving

disputes, disagreements, or differing perspectives among individuals or groups.

By identifying the conflicts efficiently, it will in turn be helpful to reduce negative impact and

increase positive impact.

**THOMAS-KILMANN technique** is a widely used technique for understanding and managing conflict.

The Thomas-Kilmann technique helps individuals understand their preferred conflict-handling styles and provides insights into when each mode might be appropriate.

**10Ans. Reasons for project failure are:**

**Improper requirement gathering-**

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

failure.

**Lack of stakeholder involvement-**

A project can fail if the stakeholders are not participating in the process. The stakeholders

input and feedback plays very important role to meet the goals.

**Ineffective or less communication-**

If there are communication issues between stakeholders, team members then this can lead

to misunderstandings or delays in project or even can lead to project failure.

**Continuous change in the requirement-**

if the requirements keep on changing frequently, this can also lead to project failure.

Because the scope of the project will also keep on changing which will lead to project failure.

**Poor risk management-**

Poor risk management can also lead to project failure. The team fails to identify the risks and

do the risk mitigation, which can lead to unexpected challenges or delays in project. Lack

of user involvement. Lack of executive support.

**Unrealistic expectations-**

means the goals that cannot be achieved or the goals that are out of scope

**Improper planning-**

The project can fail if the planning is not done properly. The milestones, goals should be

discussed. If there is no proper planning, then team may face difficulties in addressing the

issues or to track the progress.

**11Ans.**

**The Challenges faced in projects for BA:**

**1.** Changing requirements

2. Lack of stake holders involvement

3.unclear project objectives.

4. Managing conflicts & negotiations.

5. project communication

6. Resistance to change

Co-ordination between developers and testers

7.Conducting meetings.

8.Making sure status report is effective

9.Driving clients for UAT completion

10.Making sure that the project is going on right track and delivered as per the timelines without any issues.

**12Ans.**

[ProjectID][Document Type]V[x]D[y].extension

Example- [PQ777FRDV1D1.docx] or [PQ777FRD1.1.docx]

**13Ans.** **Do’S & Dont’S**

There is no word called as “BY DEFAULT”

* Never imagine anything in terms of GUI
* Question everything in the world
* Go to the client with plain mind that is with no assumptions
* Listen to the client very carefully and after he is done, then ask questions
* Don’t interrupt the client.
* Never try to give solutions to the client right away.
* Try to concentrate only on important and required things.
* Be like a lotus in mud- if a client comes with a fancy requirement, then talk to the
* project manager first.
* Requirement hurried-project buried.
* Never criticize the stakeholder.
* Always appreciate the stakeholder even for small
* efforts.

**14Ans.**

**Packages**- it is a group of classes or use cases that are used to organize model elements.

Packages can be nested within other packages.

1. These are used as containers to organize elements.
2. It is very useful to represent system architecture.

Subsystems- it is logical grouping of related components.

It is collection of classes, packages, libraries and other sub systems that work together to

deliver a specific set of functionalities

**Subsystems**- It is logical grouping of related components.

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

**15Ans.Camel-casing** refers to the naming convention of variable, parameters or properties.

Here, multiple words are combined together.

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

with capital letters.

Ex- first Name, last Name

In BA, camel-casing is used in requirements documentation.

In requirement documentation, BA often use camel-casing to name the entities like use case,

features, user stories like validate Customer Details, calculate Interest Rate, etc

Business rules, which should be satisfied by the system use camel-casing.

While documenting business process or work flows, camel-casing can be used to individual in steps.

This will help maintain consistency in the document.

The database tables name also uses camel-casing.

**16Ans**.**Development server:**

A development server refers to a dedicated environment that is used during

the software development process.

The accesses a BA has are-

1. Read only access

2. Collaborative access.

3. Limited configuration access.

**17Ans. DATA MAPPING**

It is the process of establishing a relationship or connection between data elements a relationship or connection between data elements in two or more different data sources or data formats.

The main purpose of data mapping is-

Data integration-

While combining 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 new system(destination), 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 Transformation-

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

data transformation plays very important role which ensures that the data of legacy

system(source) is mapped correctly to the data in new system.

**18AnS.**

**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 applicati on 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.

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.