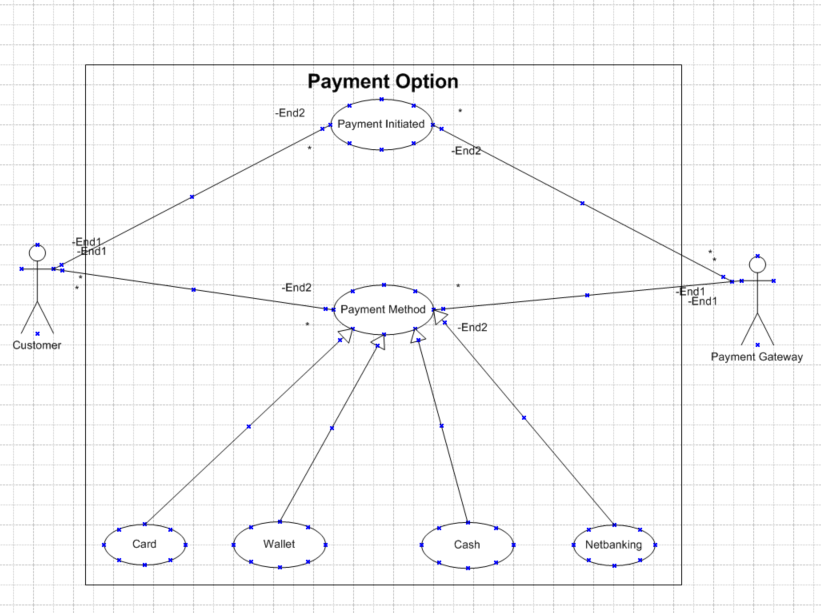
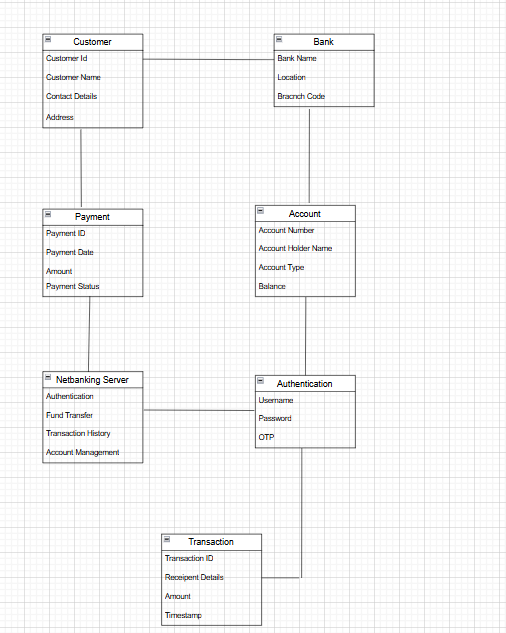
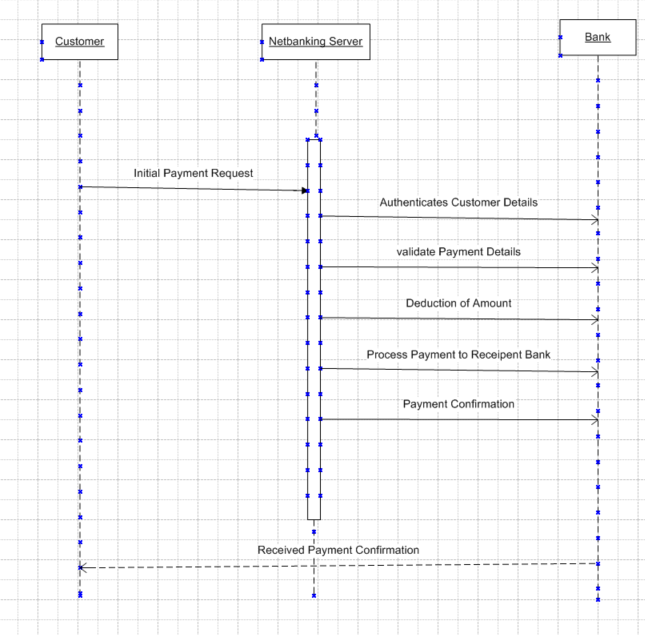
**Capstone Project Prep 3- Part 1**

**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  
  
**Solution:**   
**  
  
Question 2:** Derive Boundary Classes, Controller classes, Entity Classes  
  
**Solution:   
Boundary Class:** Boundary Class used to handle interactions between the system and external actors.  
**Example:** PaymentOptionBoundary  
CardPaymentBoundary  
 WalletPaymentBoundary  
 CashPaymentBoundary  
 NetbankingPaymentBoundary   
**Controller Class:** CC acts as intermediate between boundary class and entity class.   
**Example:** PaymentinitiatedController  
 CardPaymentController  
 WalletPaymentController  
 CashPaymentController  
 NetbankingPaymentController  
   
**Entity Class:** EC represents the core data and business logic of the application  
**Example:** Customer  
 Payment  
 Card  
 Wallet  
 Cash  
 Server   
  
**Question 3:** Place these classes on a three tier Architecture  
  
**Solution:** 3-Tier Architecture has 3 layers Application Layer, Business Logic Layer and Database Layer.

|  |
| --- |
| **Application Layer** |
| PaymentMethodsSelectionBounday |
| CardPaymentBounday |
| WalletPaymentBoundary |
| CashPaymentBoundary |
| NetbankingPaymentBoundary |
|  |
| **Business Logic Layer** |
| PaymentController |
| CardController |
| WalletController |
| CashController |
| NetbankingController |
|  |
| **Database Layer** |
| Customer(Entity Class) |
| Payment(Entity Class) |
| Card (Entity Class) |
| Wallet (Entity Class) |
| Cash (Entity Class) |
| Netbanking (Entity Class) |

**Question 4:** Explain Domain Model for Customer making payment through Net Banking  
  
**Solution:**   
**Domain Model:** Domain Model is also known as conceptual model. A conceptual model depicts the concepts (ides, thing or object) that are easily identifiable in the problem description.   
  
> A Domain Model is a conceptual representation that defines the structure, relationship and behaviour of entities within a specific problem domain.   
  
   
  
**Question 5:** Draw a sequence diagram for payment done by Customer Net Banking  
  
**Solution:   
Sequence Diagram:** The sequence diagram is used primarily to show the interactions between classes in the sequential order in which those interactions occur. A sequence diagram can map a scenario describes by a use case in step-by-step detail to define how classes collaborate to achieve the application goal.  
  
> A sequence diagram is a type if diagram used in software engineering and systems design to illustrate how process operates with another process and in what order.   
  


**Question 6:** Explain Conceptual Model for this Case  
  
**Solution:   
Conceptual Model:** 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 provides a clear and simplified view of the domain, making it easier to understand   
> Key elements of conceptual model are:   
  
1. Entities: Customer, Product, order & payment  
2. Attributes: Customer ID, Customer Name, Email, Phone number  
3. Relationship: Customer Places an order  
  
> A conceptual model for the payment process done by customer using net banking provides a high-level understanding of the key concepts and their relationship involved in the payment transaction. It helps in visualizing the overall structure a flow of payment process.   
  
**Entities:** Customer and Payment  
**Attributes:** Customer ID, Password, Customer Name, Contact Details, Address, Recipient details  
**Relationship:** Customer using net banking to transfer the fund  
  
**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  
  
**Solution:**   
**MVC Architecture:**To identify classes from use case diagram, we apply MVC rules on each use case to derive classes.   
  
> MVC stands from Model, View and Controller  
  
**Model:** The Model class knows about all the data that need to be displayed. It is model who is aware about all the operations that can be applied to transform that class. The mode represents enterprise data and the business rules that govern access to and updates of this data. This represents database, tables in the database. All model classes are represented as Entity Classes.   
  
  
  
  
  
  
  
  
 Entity Class, Database classes,   
 Persistent class (Back-end designers)  
  
**View:** The View represents the presentation of the application. The view class refers to the model. It uses the query methods of the model to obtain the contents and renders it. View class is the data required, by the query. View class is represented as Boundary Class or Form Class.   
  
  
  
  
  
   
   
  
  
 Boundary Class  
  
**Controller:** Whenever the user sends a request for something then it always go through the controller. The controller is responsible for intercepting the requests from view and passes it to the model for the appropriate action. Once the action is taken on the data, the controller is responsible for directing the appropriate view to the user. In GUIs, the views and the controllers often work very closely.



**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 two Boundary classes.   
 3. Combination of Three Actors and a Use Case results in Three Boundary Classes and so on…

**Note:** Only one primary actor is to be considered with a use case.   
4. Use case will result in a controller class

5. Each Actor will result in one entity class  
  
**Question: Explain BA contributions in project (Waterfall Model – all Stages)**

**Solution:**

|  |  |  |
| --- | --- | --- |
| **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 Framework, Project Scope and Business case writing, Risk analysis | Business Case, SOW (Statement of Work), PO (Purchase Order), Sr. BA, Business Architects, Presales Consultants |
| **Planning & Estimations & Assessment** | 1. Understand Assumptions and Constraints along with Business Rules and Business Goals 2. Plan Packages for Big Projects 3. Understand the project plan from PM 4. BA conducts Stakeholder Analysis 5. Plan BA approach strategy (Req. gathering techniques, communication, Req. mgmt., Documents to follow, Tools to use, Change Request Handling methodology for this Project) | PM, Sr. BA |
|  |  |  |
| **Requirements Gathering** | 1. Stakeholders identify and document 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 give more specific requirements 4. Sort the gathered Requirements (avoiding duplicate Requests, grouping into similar functionality or logical Requests) 5. Prioritize requirements – MoSCoW 6. Validate Requirements – FURPS | BRD (Business Requirements Document), BA, PM |
| **Requirements Analysis** | 1. Draws UML Diagrams (Use case and Activity Diagrams) 2. Prepares Functional Requirements from Business Document 3. All Architects come 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 binding Doc between the Business and the Technical Team 6. BA prepared RTM from SRS before Design phase starts (BA is the owner of RTM) 7. BA tests the requirements are dealt in each phase of development life cycle from Design till UAT | Functional Requirements Specification, SSD (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. Solution-Architect designs Content on the design and Solution documents (updates Status to Client and stakeholders) | Solution Document, Design Document – HDD – ADD |
| **Coding** | 1. BA organizes JAD Sessions 2. BA clarifies queries of Technical Team during Coding 3. Developers refer Diagrams and Transient (Controller Classes) of BA and code their unit 4. Update End user manuals 5. Update RTM 6. Conducts regular Status meetings with technical team and the Client and tuning Client for participation in UAT | LDD – CDD, Application, Development Team, BA, PM |
| **Testing** | 1. BA- Prepares Test Cases from Use Cases or assists Test Manager to do so 2. BA performs high-level testing 3. BA prepares Client for UAT 4. Test Data is requested by BA from Client 5. Updates End User Manuals 6. Updates RTM 7. Take signoff from Client on Client Project Acceptance | Test Concerning Documents, Application with fewer errors, Testing Team, BA, Client |
| **Deployment and Implementation** | 1. Forwards RTM to Client or the 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. Takes Lessons learned from this project (to take precautions for coming projects) | N/A |

**Question 9:** What is conflict management? Explain using Thomas – Kilmann technique  
**Solution:**   
**Conflict Management:** Conflict Management refers to the process of handling and resolving conflicts or disagreements that rise between individuals or group within an organization.   
  
> Thomson Kilmann Conflict Mode Instrument (TKI) is a widely used technique for understanding and managing conflict.   
  
> This technique helps individuals understand their preferred conflict management styles and provides insights into when each mode might be appropriate.   
  
> Effective conflict management involves recognizing the existence of conflicts, actively listening to the concerns of all parties involved, seeking common ground and working towards collaborative solutions that meet the needs of everyone.  
  
**Options of Conflict Management:** There are 5 options of conflict management as follows:   
 1. Competing

2. Avoiding

3. Accommodating

4. Collaborating  
 5. Compromising  
  
**Steps to Conflict Management:** There are 5 steps to Conflict Management as follows:   
 1. Identify Conflict

2. Discuss the details

3. Agree with root problem

4. Check for every possible solution for the conflicts

5. Negotiate the solution to avoid the future conflicts.

**Question 10:** List down the reasons for project failure  
**Solution:** The following are the reasons for any project failure:  
 **1. Improper Requirement Gathering:** Inadequate or unclear collection of project requirements from stakeholders leading to misunderstanding and incorrect implementations.

**2. Continuous Change in Requirements:** Frequent change requests from stakeholders that disrupts development progress and increase budget.

**3. Lack of User Involvement:** Minimal or no participation from end users in the project, resulting in a final product that may not meet the criteria and project failure.

**4. Lack of Executive Support:** Absence of commitment or guidance from management causing resources constraints and project delays sometimes project failure.

**5. Unrealistic Expectations:** Setting impractical goals or requirements like timelines, budget, scope will lead poor quality outcomes and project failure.

**6. Improper Planning:** Improper Planning or inadequate project roadmap, resource allocation or risk management by the Project Manager resulting in inefficiencies and missed deadlines followed by project failure.  
  
**Question 11:** List the Challenges faced in projects for BA **Solution:** The following points would act as challenges faced in the project by a Business Analysts.  
**1. Ambiguous or Changing Requirements:** Requirements that are unclear, incomplete or frequently changing can cause delays, rework and confusion in project execution.

**2. Stakeholder Management:** Managing multiple stakeholders with differing priorities and expectations or requirements can be challenging.

**3. Lack of Stakeholder Involvement:** minimal participation from key stakeholders can lead to misaligned project goals, incomplete requirements and a product that does not meet business needs.

**4. Unclear Project Objectives:** When project goals and acceptance criteria are not well defined, it will become difficult to meet the requirements and project success.

**5. Project Communication:** Poor communication among the team and stakeholders can result in misunderstanding, missed requirements and project failures.

**6. Time and Resource Constraints:** Limited time, budget or resources can create project inefficiencies which effect the quality of deliverables and overall project success.

**7. Resistance to Change:** Stakeholders or end – users may resist new processes or systems, making it difficult to implement solutions smoothly and achieve business requirements or objectives.

**Question 12:** Write about Document Naming Standards

**Solution:** Every document related to the project must have a consistent and standard naming format.   
  
> This pattern must be followed for every document related to the entire project.   
> Document naming standard as follows:   
 **[PorjectID][Document Type]V[X]D[Y].ext  
  
Example:** ABC123BRDV1D1.docx  
 ABC123BRDV1D2.docx

**Question 13:** What are the Do’s and Don’ts of a Business analyst  
**Solution:** The following are the Dos and Don’ts as BA

1. Never say NO to client
2. Ther is NO word called “By Default”
3. Never imagine anything in terms is GUI
4. Question the existence of existence / question anything in the world

Example: what client says is not always correct

1. Consult and SME for clarifications in requirements
2. Every problem of client is unique. No two problems of different client are same, may be approach, technology, place of use, local laws may be varied to make them to be different.
3. Go to client with a plain mind with no assumptions. Listen carefully and completely until client is done and then you can ask your queries.
4. Do not interrupt client when they giving you the problem.
5. Try to extract the leads to solution from the client itself.
6. Never try to give solutions to client straight away with your previous experiences and assumptions.
7. Try to concentrate on the important and truly required requirements.
8. Don’t be washed away by add on functionalities or don’t imagine solutions on screen basis.

**Question 14:** Write the difference between packages and sub-systems

**Solution:   
Packages:**   
Collection of components which are not reusable in nature.

**Sub-systems:**

Collection of components which are reusable in nature.

|  |  |  |
| --- | --- | --- |
| **Event** | **Packages** | **Sub-systems** |
| Granularity | Smaller and more focused in scope | larger and encompass multiple packages or modules. |
| Dependency Management | Managed 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  
 **Solution:** Camel casing is a naming convention used in computer programming and 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.

**Example:** getEMPId();turnLeftAndSlowDown();

turnLeftAndThenRight();

slowDownAndStop();

**Question 16:** Illustrate Development server and what are the accesses does business analyst has?  
  
**Solution:**   
**Server:** A server is a temporary is a system that provides resources, services or programs to other systems or other computers.   
  
**Development Server:** 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.   
  
**Business Analyst Access:**

1. Read – Only Access
2. Collaborative Access
3. Limited Configuration Access

**Question 17:** What is Data Mapping  
  
**Solution:** Data mapping is the process of establishing a relationship or connection between data elements in two or more different data sources or data formats.

> It involves defining how data from one source corresponds to or transforms into data in another sources.

> Data Mapping is commonly used in data integration, data migration and data transformation processes.

> The purpose of data mapping is to ensure that the data can be accurately and effectively transferred, converted or transformed between different systems, databases or formats.

> It involves identifying the source data elements, determining their meaning and structure and mapping them to the corresponding target data elements.

**Question 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

**Solution:   
  
API:** API stands for Application Programming Interface, is a set of rules and protocols that allows different software applications to communicate and interact with each other.   
  
> It enables data exchange and functionality sharing between systems without requiring direct access to their internal code.   
  
**Setup API Communication:**

> Will set up API communication between our application and the other application to exchange the data.  
  
**Data Formatting:**

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

**Data Parsing:**   
 > When receiving date data from the other application, parse them mm-dd-yyyy formatted date into our application’s dd-mm-yyyy format.

> We must have to again extract the day, month and year data components and rearrange them accordingly.

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