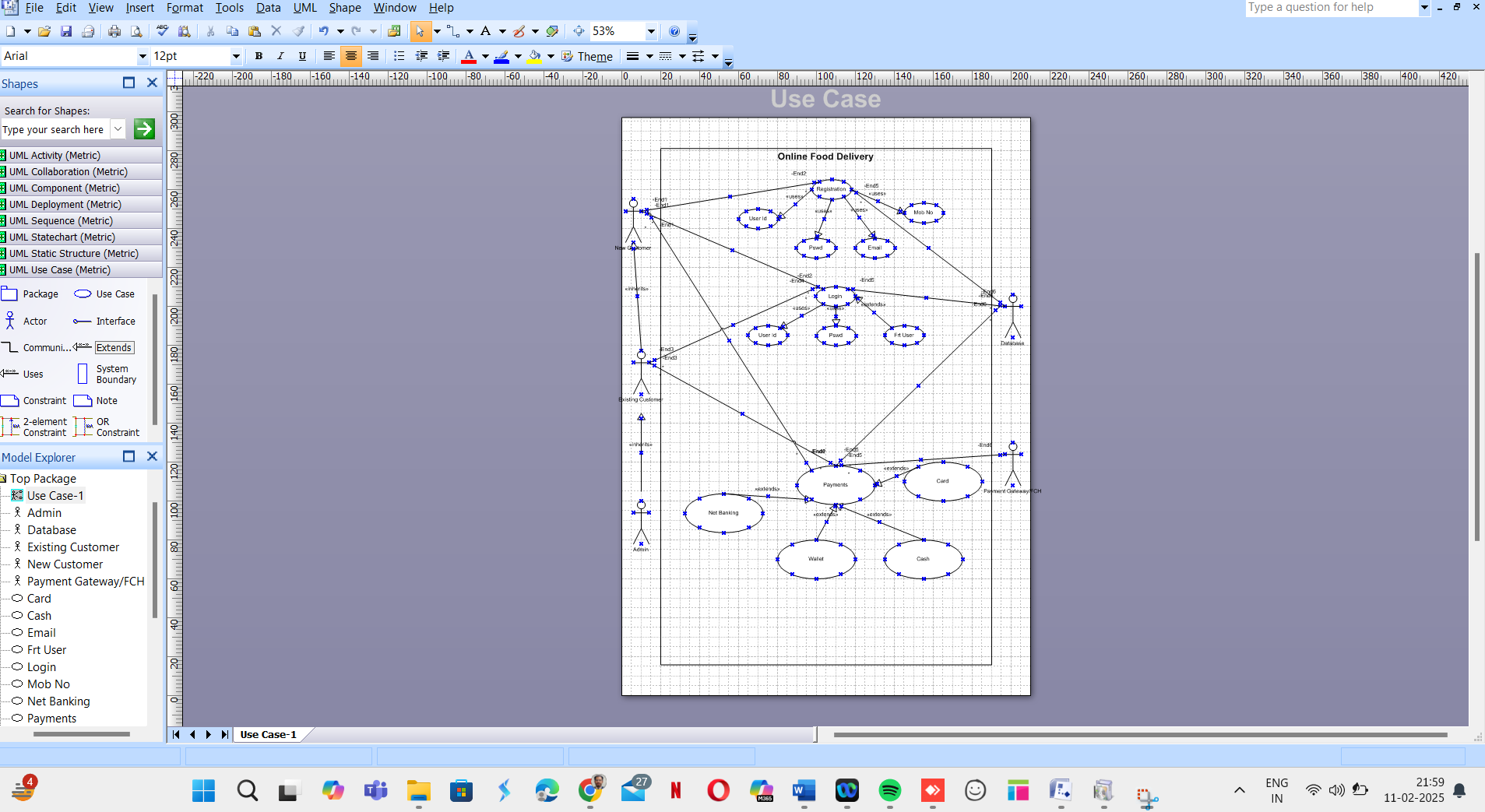
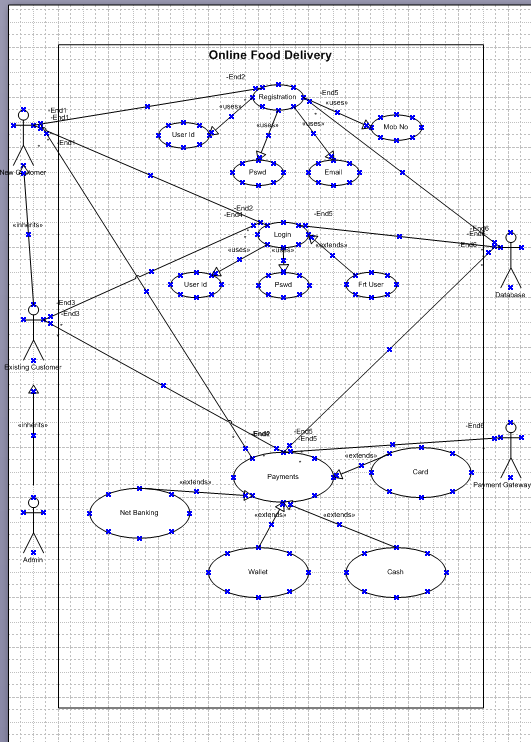
**Capstone Project 3**

**Q1 Draw a Use Case Diagram - 4 Marks**

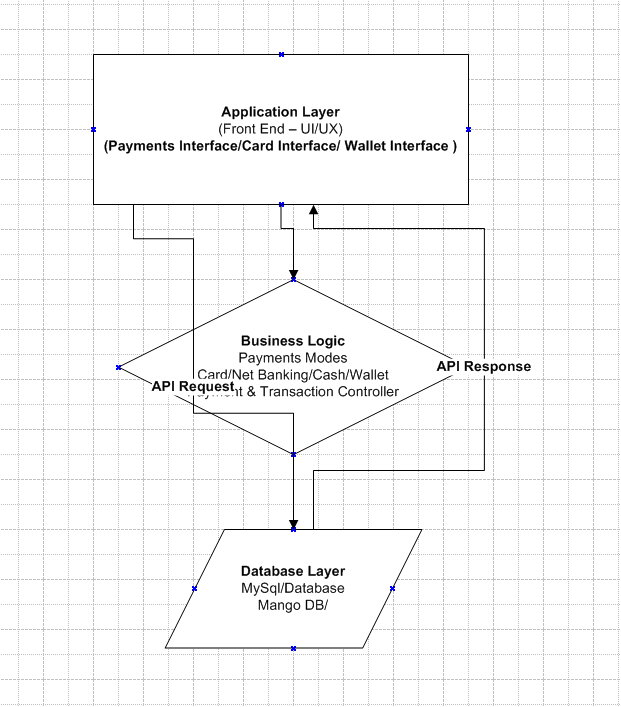
****

****

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

* Boundary Classes: Payment Interface, Card Interface, Wallet Interface, Net Banking Interface
* Controller Classes: Payment Controller, Transaction Controller
* Entity Classes: Customer, Payment, Transaction, Bank Details

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

****

**The Three tier Architecture could be broken into below**

* Presentation Layer: Payment Interface, Card Interface, Wallet Interface
* Business Logic Layer: Payment Controller, Transaction Controller
* Data Layer: Customer, Payment, Transaction, Bank Details

**Q4. Explain Domain Model for Customer making payment through Net Banking - 4 Marks**

The domain model represents objects involved in net banking payments, such as:

* Customer initiates payment
* Net Banking Interface facilitates transactions
* Transaction Record maintains details
* Bank Server verifies and processes the payment

**Q5. Draw a sequence diagram for payment done by Customer Net Banking - 4 Marks**

* Customer selects Net Banking payment method
* Payment system requests authentication from the Bank
* Bank authenticates and processes payment
* Confirmation sent to the payment system and customer

**Q6. Explain Conceptual Model for this Case - 4 Marks**

A conceptual model abstracts key entities and their relationships. Entities include Customer, Payment, Transaction, and Bank. Relationships:

* Customer makes a Payment
* Payment is processed through Net Banking
* Transaction is recorded

**Q7. What is MVC architecture? Explain MVC rules to derive classes from use case diagram and guidelines to place classes in 3-tier architecture**MVC (Model-View-Controller) is a design pattern that separates an application into three interconnected components:

* Model: Manages data and business logic (Entity Classes)
* View: Handles UI representation (Boundary Classes)
* Controller: Processes input and coordinates Model and View (Controller Classes)

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

* **Requirement Gathering:** Conducts stakeholder analysis, documents requirements
* **Design:** Assists in preparing functional specifications
* **Implementation:** Ensures alignment with requirements
* **Testing:** Validates system functionality
* **Deployment:** Provides user training and documentation
* **Maintenance:** Supports post-deployment changes

**Q9. What is conflict management? Explain using Thomas-Kilmann technique**  
Conflict management involves resolving disputes effectively. The **Thomas-Kilmann technique** outlines five strategies:

* Competing
* Collaborating
* Compromising
* Avoiding
* Accommodating

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

* Poor requirement gathering
* Lack of stakeholder involvement
* Scope creep
* Poor project management
* Inadequate testing

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

* Unclear requirements
* Stakeholder conflicts
* Frequent changes
* Communication gaps
* Tight deadlines

**Q12. Write about Document Naming Standards**

* Use version control (v1.0, v2.0)
* Include document type (FRD, BRD)
* Follow consistent format (YYYY-MM-DD)

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

* Gather clear requirements
* Maintain good communication
* Validate requirements

**Don’ts:**

* Assume requirements
* Ignore stakeholder feedback
* Skip documentation

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

* **Packages**: Logical grouping of classes
* **Sub-systems**: Independent components with defined functionalities

**Q15. What is camel-casing and explain where it will be used**  
Camel-casing capitalizes the first letter of each word except the first (e.g., customerPayment). Used in variable and method naming.

**Q16. Illustrate Development server and what are the accesses a BA has?**  
Development servers host test environments. BAs may have:

* Read access to test data
* Access to logs and reports
* Limited debugging permissions

**Q17. What is Data Mapping?**  
Data mapping links data fields between source and destination. Used in migration and integration.

**Q18. What is API? Explain how you would use API integration in the case of your application**  
An API (Application Programming Interface) allows different systems to communicate. In this case, the system must handle date format conversion between **dd-mm-yyyy (local)** and **mm-dd-yyyy (US)** by mapping and transforming data formats correctly.