

NAME: MBEVO MITCHEL WAWIRA

ADM NO: 1049085

DATE: 06/11/2024

UNIT CODE: CMT 310

UNIT TITLE: OBJECT ORIENTED PARADIGM AND SOFTWARE DEVELOPMENT

ASSIGNMENT TITLE: CASE STUDY ASSIGNMENT ON OBJECT-ORIENTED PARADIGM, OOAD, AND UML

LECTURER’S NAME: MR. JULIUS SIRMA

**QUESTION 1:**

**Question 1: Understanding Object-Oriented Concepts**

**a) Benefits of an Object-Oriented Approach for Developing a CRM System (3 Marks)**

1. **Modularity and Maintainability**

Object-oriented means that the CRM system is modularized into separate, manageable pieces-or objects-such as "Customer," "Order," and "Account." Each of these modules maintains its own data and functions in a way that is easily maintainable or updatable independently without affecting other parts of the system. Also, separation of concerns keeps the system organized and makes debugging easier.

2. **Code Reusability**

OOP allows objects and classes to be reusable within a system or even across systems. For example, a class like "Customer" could be reused multiple times within the CRM system, let's say, at places involving sales and customer support, without having to redefine customer attributes or functions every time. Additionally, such reuse decreases development time and effort.

3. **Flexibility and Scalability**

OOP allows the addition of new features or modifications of existing ones without requiring big rewritings: new types of customers, orders, and other entities would easily be added by inheriting their properties and methods. Later on, such a system would be easier to scale when the business of the client is growing.

**b) Three Possible Classes in the CRM System and Their Interactions (5 Marks)**

**1. Customer Class**

Purpose: Deals with specific customer information.

Attributes: `name`, `email`, `phone`, `address`.

Methods: `registerCustomer()`, `updateDetails()`, `deleteAccount()`.

Relationships: The Customer class would relate to or interact with the AccountManager class in terms of managing login and authentication. It would further relate to the Order class regarding maintaining orders placed by each customer.

**2. Order Class**

Purpose: Store the details in an order and the status.

Attributes: 'orderID', 'orderDate', 'customerID', 'orderStatus'.

Methods: createOrder(), updateOrderStatus(), viewOrderDetails().

Interactions: The Order class shall interact with the Customer class to assign a customer for every order, and with the Inventory class to check the availability of an item at the time of placing an order.

**3. AccountManager Class**

Purpose: It regulates the user's access and authentication.

Attributes: username, password, role.

Methods: login(), resetPassword(), manageRoles().

Interaction: The class AccountManager will interact with the Customer and Order classes in order to restrict the access to modifications regarding sensitive information, or the placing of the order, only by authorized users.

**Question 2: OOAD Process**

**a) OOAD Phase Steps to Ensure the CRM System Meets Client Requirements (4 Marks)**

**1. Requirements Gathering**

The interviews with the stakeholders will be held at the very outset, to understand in fair detail what is required by the client to be done by the system in question, considering also the features and its security-related aspects. This will explain what the proposed CRM system will be doing, in which customer details are kept securely and management of orders is held.

**2. System Analysis**

Identify the major components in this system of CRM: customer management, order processing, and account handling. This step is a part of defining what kind of objects the system needs and how these objects would relate.

**3. Design Using UML Diagrams**

Draw UML diagrams, which are Class Diagrams and Use Case Diagrams to depict what goes into the system and the relationship between different components. This ensures that everything has been represented, and logical flow is sensible in the design phase.

**4. Client Review and Feedback**

Present the preliminary design and prototypes to the client in order to verify that it meets the expectations of the client. This feedback loop helps in catching misunderstandings or additional requirements early.

**5. Implementation and Testing**

Develop the system and test each module for functionality and security. The testing phase here will make sure that the CRM works as expected and meets the client's standards.

**b) Applying Design Patterns for Scalability and Maintenance in CRM (4 Marks)**

**1. Singleton Pattern**

It ensures through the Singleton pattern that there is, at any moment in time, only one instance of something, such as a database connection. That minimizes memory use and removes conflicts. In CRM, this pattern can be used to manage the database access layer efficiently.

**2. Factory Pattern**

The Factory pattern enables the creation of objects based on certain requirements. In a CRM, this could be utilized to instantiate different kinds of customers, for instance, individual versus corporate, without modifying the core customer class.

**3. Observer Pattern**

The Observer pattern provides the ability for subjects, or objects, to be notified automatically whenever a related object-the observable one-changes. Consider, for example, a CRM in which the system automatically notifies the other modules, say Sales or Support, when a customer updates his profile.

**Question 3: UML Diagrams**

**a) Use Case Diagram for CRM System User Interactions (4 Marks)**

**1. Actors**

Sales Representatives: Add new customers, view customer orders, and update information.

Administrators: Providing administrative functions like user account management, permissions, and data access.

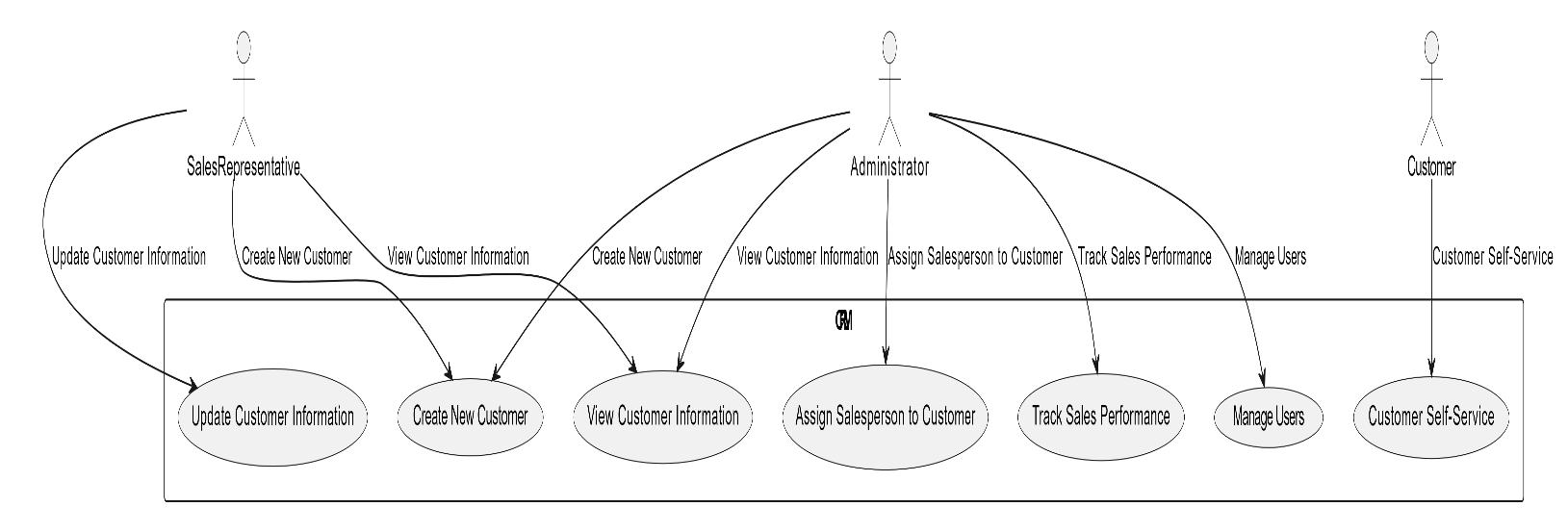
Customer: To view own profile and order history.

**2. Use Cases**

- The Sales Representatives do something "Add Customer" and "Update Order."

- The Administrators do something "Manage Accounts" and "Set Permissions."

- The Customers do something "View Profile" and "Track Orders."



**b) Activity Diagram for Adding a New Customer (3 Marks)**

1. Start: Sales Representative logs into CRM.

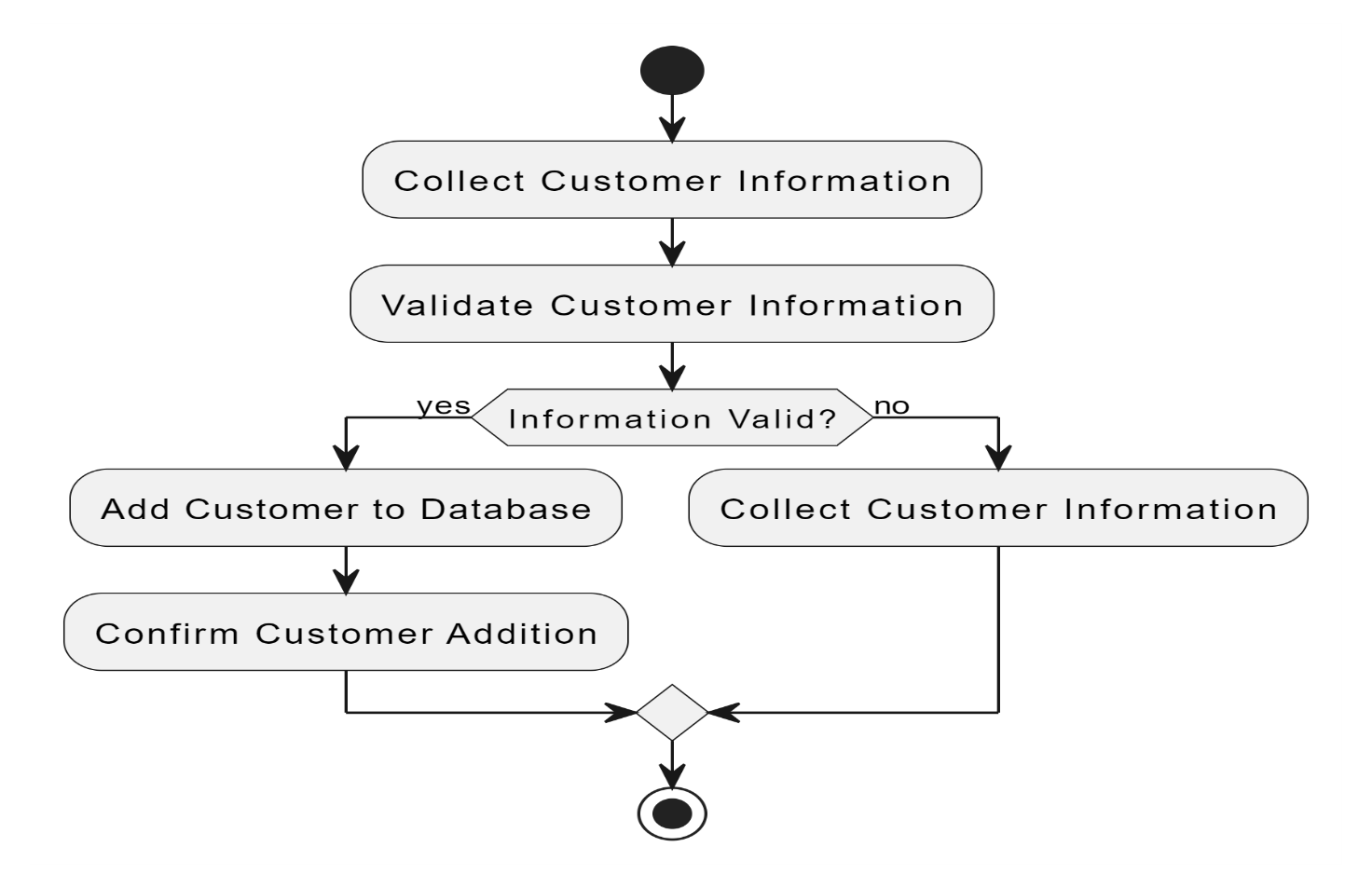
2. Input Customer Details: Sales Representative inputs details like Name, Contact Information, and Address

3. Validate Information: System checks for errors or missing information.

4. Save Details: The CRM saves the new customer record.

5. Confirmation: System sends a confirmation message or email.

6. End: Process finishes.

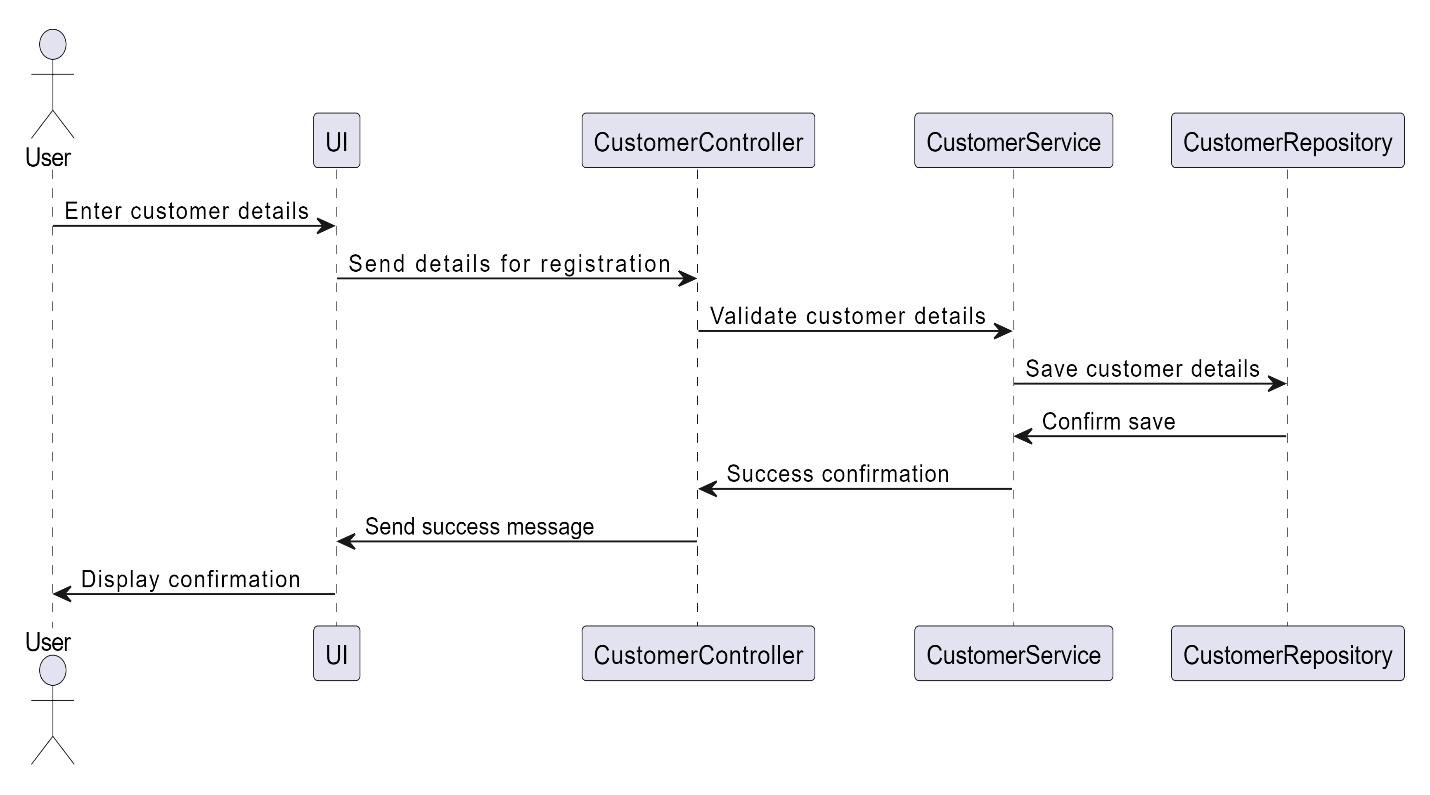


**c) Using Interaction Diagrams for Customer Registration Process (3 Marks)**

1. Customer Object sends the registration information to AccountManager.

2. AccountManager checks details and communicates to the Database for storing the record of a new customer.

3. Database confirms the successful registration, and the confirmation is reflected back to Customer Object by AccountManager.



**Question 4: Ethical Issues in Software Development**

**a) Ethical Issues regarding Data Privacy and Security in Designing CRM System (4 Marks)**

**1. Data Privacy**

Data protection laws, such as the GDPR, will be followed by encrypting data, giving access only to a small group, and allowing the user the right to show or delete the data.

**2. Data Security**

Sensitive information, say customer contact information, should be securely authenticated, encrypted, and audited on a regular basis for possible breaches in security.

**3. Transparency and User Consent**

Clearly articulate how the customer's data is to be used. For instance, an opt-in for marketing communications enables a customer to have some sense of control about their data.

**4. Design Impact:** Because these are ethical considerations, design impacts of the CRM system are mainly oriented towards secure data handling. This could mean integrating encryption, role-based access, and clear privacy settings to protect users' information in the design.