# Problem 1

This is the best approach because it separates the implementation from the consumer and allows the consumer to instantiate classes in any combination. The purpose of the Abstract Factory is to facilitate the creation of linked object families using a standardized interface that does not rely on the predefined classes of the underlying framework. Product families may be swapped out easily using this approach, provided that the factory object's class is only referenced once in the application, during its instantiation. By creating a new physical instance of the abstract factory at a new location, the application may totally replace the whole family of products.

Request for
Customer Details
Details of
Customer
Architectural context diagram
for the video store system
Customer
Request fo

# Problem 2

Loose coupling is achieved by the usage of the chain of responsibility pattern while designing software programs. A customer's request is sent to a chain of resources that work to fulfill the request. Given that, in this scenario, we want the chain to bypass through three various forms of patterns, this is a good fit for our needs.

## UML Diagram:

Diagram

Description automatically generated

# Problem 3

Imagine you work for a shipping firm and are tasked with building a software component that will be in charge of determining shipping prices for a customer. The firm ships to locations both locally (within the same city) and nationally (within the same country), as well as internationally. There are two parts to the shipping price:

1. Handling: Company personnel handle local shipments, a third-party contractor handles domestic shipments, and certain airline employees handle international shipments.

2. Delivery: Cars may be utilized for local locations, trains can be used for domestic addresses, and aircraft can be used for international addresses

SOLUTION:

Step1: Develop a type representing your clientele.

Step2: Let's start coding the class's characteristics and methods to meet those specifications.

Let's begin to formulate an answer to this issue by considering the factors that our software uses to determine the shipping cost:

CITY comes first, then COUNTRY.

Graphical user interface, text, application, letter, email

Description automatically generated

# Problem 4

A picture containing table

Description automatically generated

**The component structure for the system:**

