CSE 522 Fall 2016
Object Oriented Analysis and Design
Project - Final Phase
Project Report

Project:
E-COMMERCE B2C PLATFORM

Team Members:
Anna Jonet Joseph
Sunil Kunjappan Vasu

Project Description

In this project we have developed an e-commerce B2C platform, for let's say ABC company, which would enable the customers to purchase goods online. The company has multiple physical stores and plan to expand its online business. Hence we plan to create a website that would help the company to provide online shopping experience to their customers.

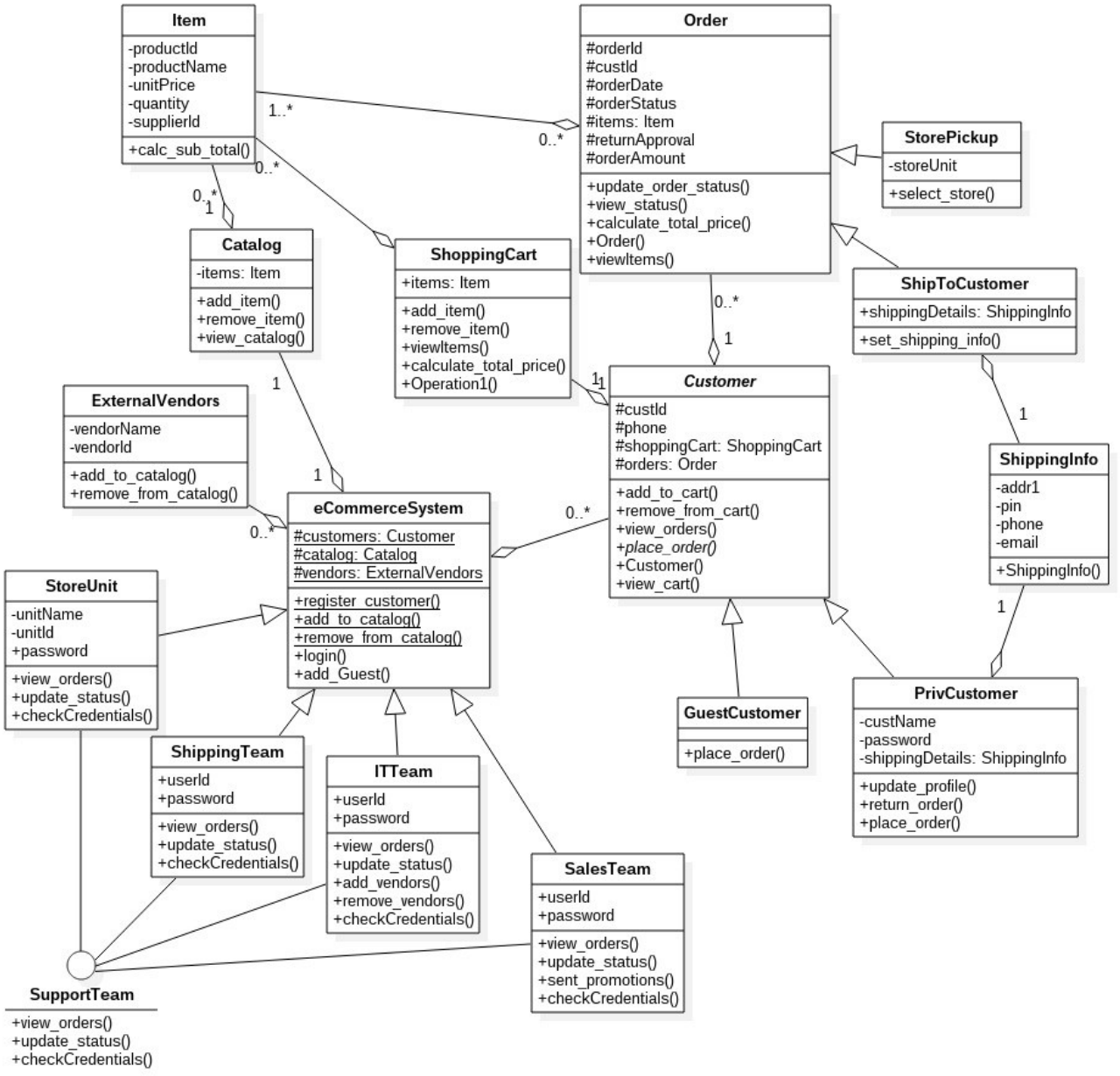
The website provides the ***customers*** with a ***catalog*** from which the customer can make a purchase. The customer can select item from the catalog and add to ***shipping cart*** and proceed to checkout. The catalog includes all the ***items*** at the physical ***Stores Units*** and items that are sold by the ***external vendors***. The site offers two mode of fulfillments for their customers – Ship-to-customer and Store-pickup. The customers are of two types – ***Privileged and Guest customer.*** Privileged customers are the ones who are registered in the website. Guest customer are the ones who perform purchases without registering. Each of the Store Units would keep track of the ***orders*** placed in that particular physical store. The company has the following support teams - ***IT team, Sales Team, Shipping Team*** – which takes care of specific tasks.

The system should allow **access** to anyone who wishes to **create** an account in the website and become a privileged customer. If anyone who wishes not to **register** should also be able to perform online shopping as a Guest Customer. The customer can **browse** through the catalog and do online shopping. There is also a **return** system which helps the customer to return the goods that are purchased within a specific time period. The IT team would be responsible to **add and remove** external vendors from the system in addition to overall maintenance task of the system. The registered external vendors can access the system and update/delete the records for each of the items that are sold by that vendor.

In Store-pick up the customer can **place orders** for items and these items should be **picked-up** by the customer within a specific time period. If the item is not picked up, then the orders is **canceled**. Each of the physical stores would **manage** the online orders for store pick up. Once these orders are ready for pick up the corresponding physical store would change the order status to "Ready for Pickup", once its picked-up they **modify** the order status to "Picked-up" in the system. And **cancel** the order in the system, if they are not picked up after certain time period. In Ship-to-customer the customer can order items from catalog and these items would be **shipped** to the shipping address provided by the customer. The Sales team are in charge of **returns and refunding** to the customer. The return can be requested by a privileged user only. The request for return placed by the customer must be approved by the Sales Team to be processed for return. Sales Team would also **send promotional, discount mails** to the companies privileged customers. They also **maintain the stock information** at each of the Physical stores The Shipping Team **manages the shipping** of orders. They **update** the shipping status [Packing, Shipping, Delivered] of the order in the system. They **manage return** of the goods pickup once the return is approved by the Sales Team.

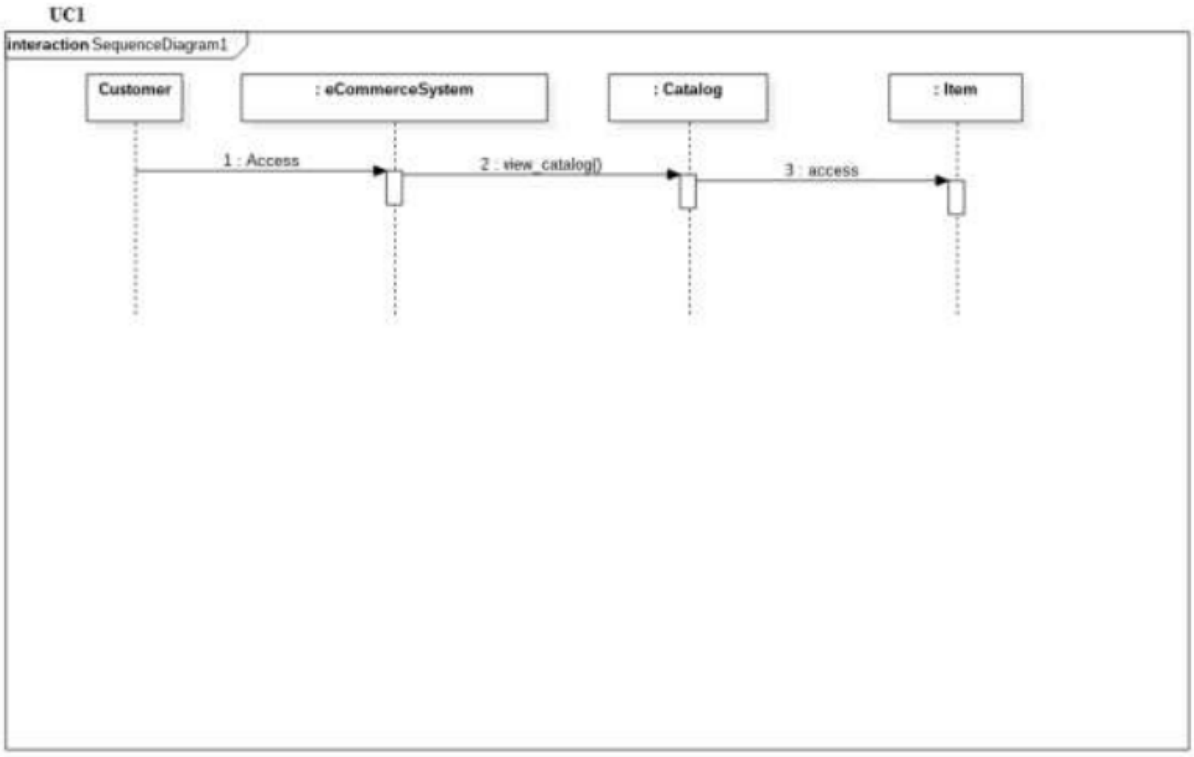
The system is designed in such a way that it work on Desktop. The front end should be catchy and should be in terms with the current theme and style. The overall system should be easy and efficient for the customers. There should be good security features so that the privileged customers' information is secure and unwanted disclose of this can be avoided.

Class Diagram

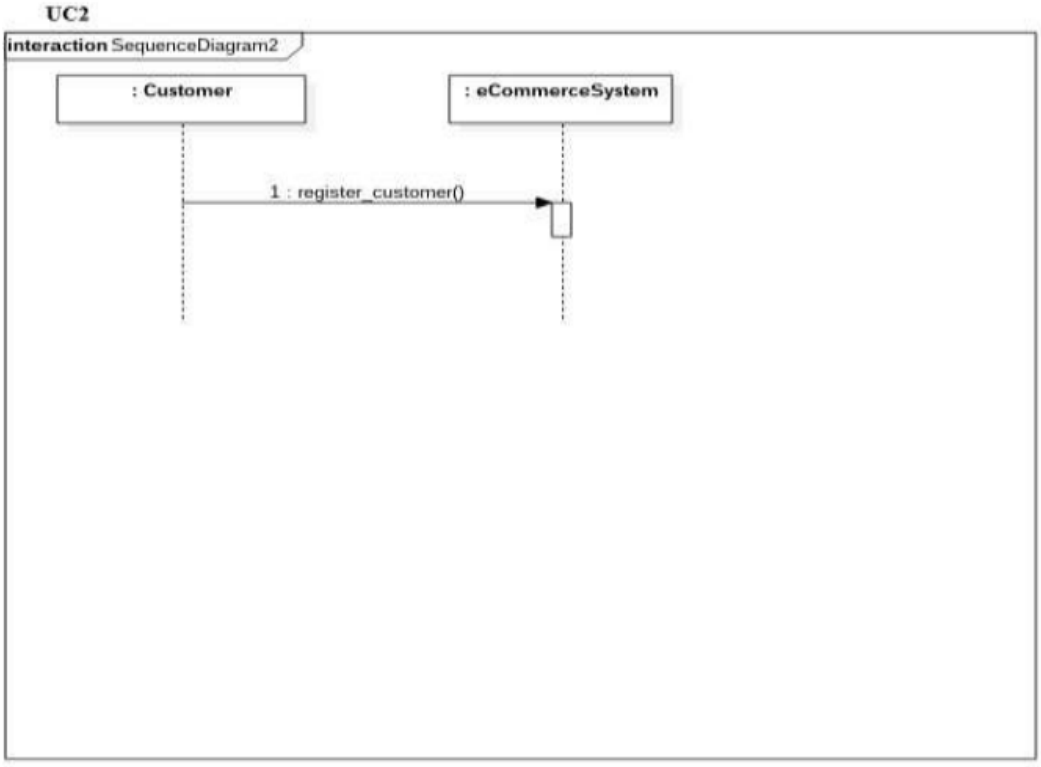


Use-Cases (Design-Time Sequence Diagrams)

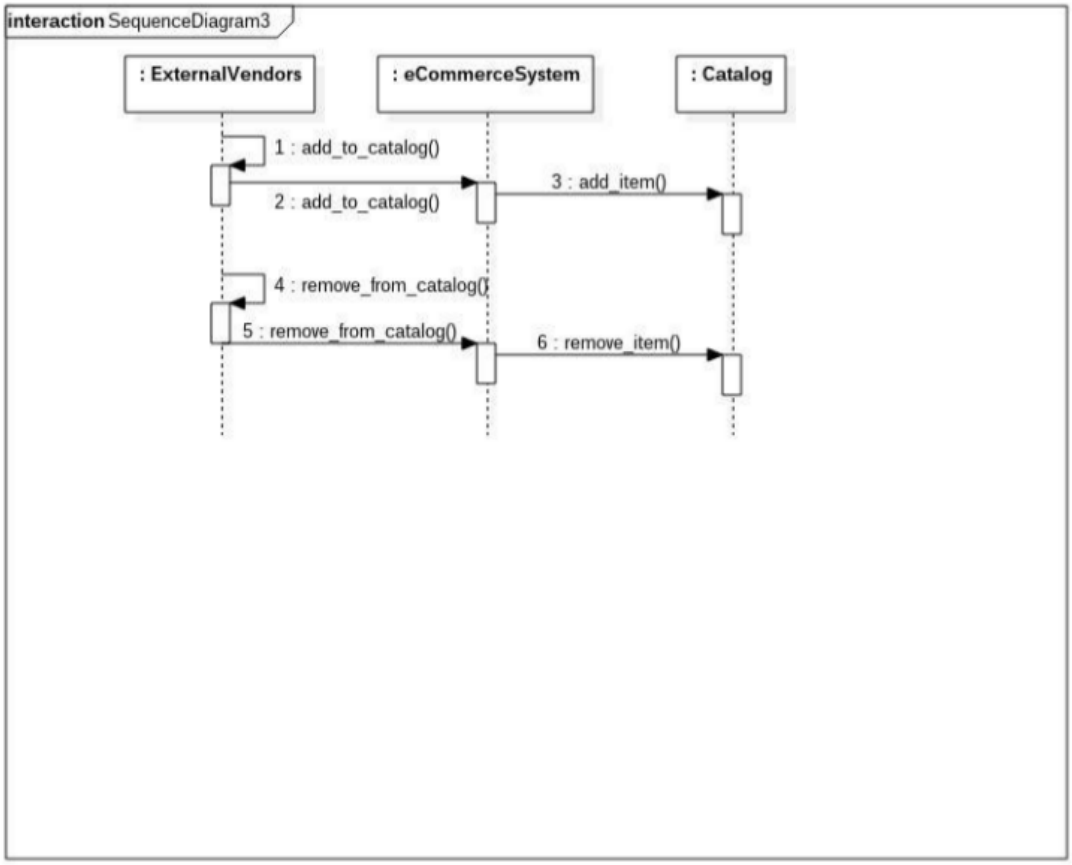
a) Use Case 1: Customer Browse through the catalog of items [UC1]



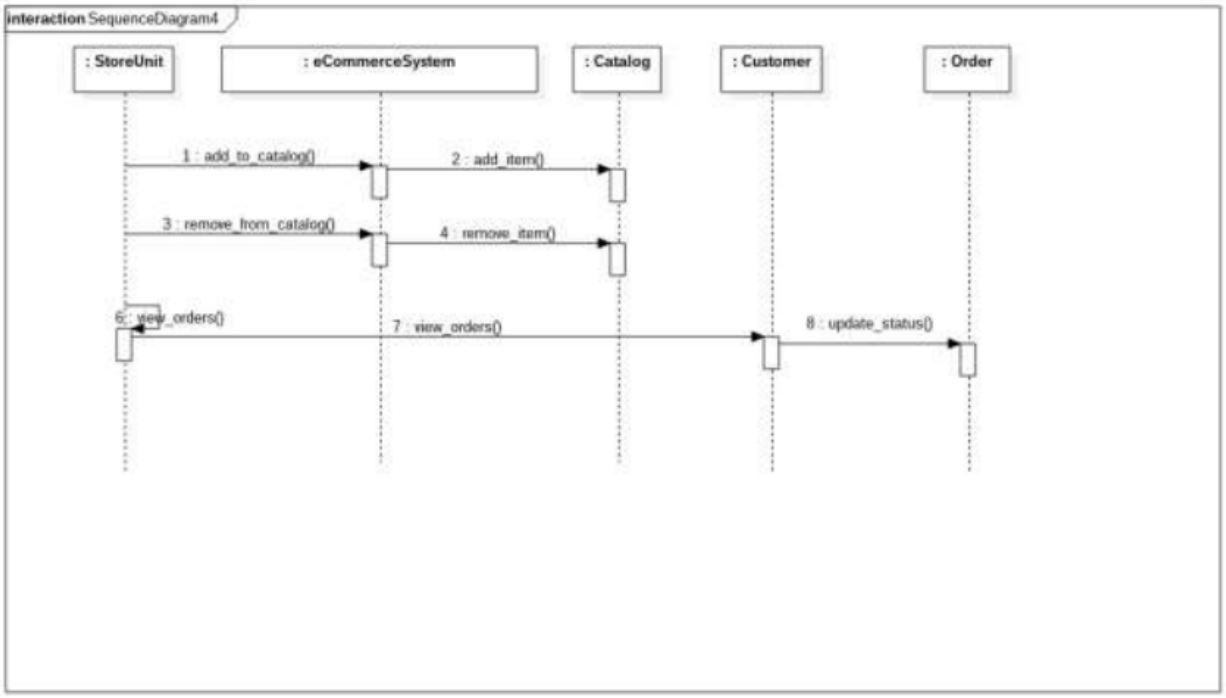
b) Use Case 2: Customer Create an Account online [UC2]



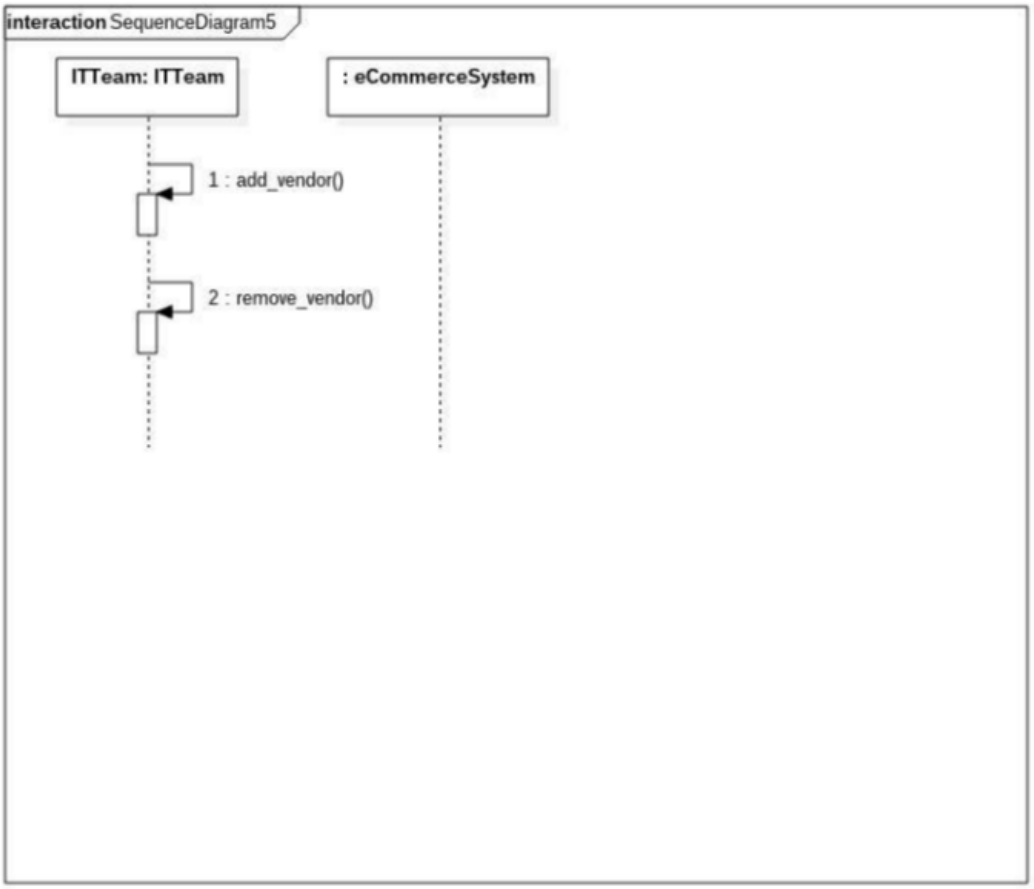
c) Use Case 3: External Vendors Manages items sold by them. [UC3]



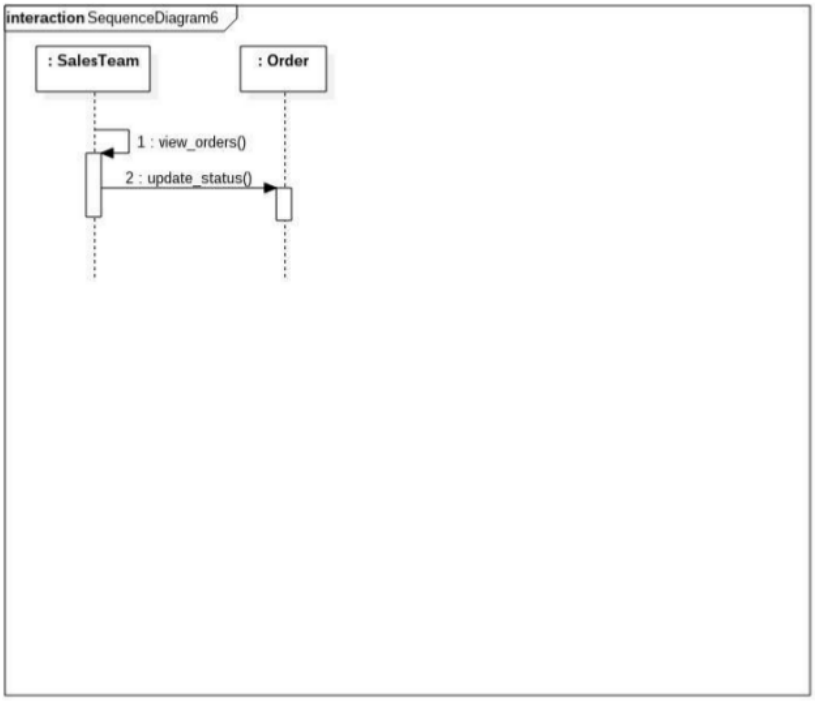
d) Use Case 4: Units Manages Orders placed at physical Store. [UC4]



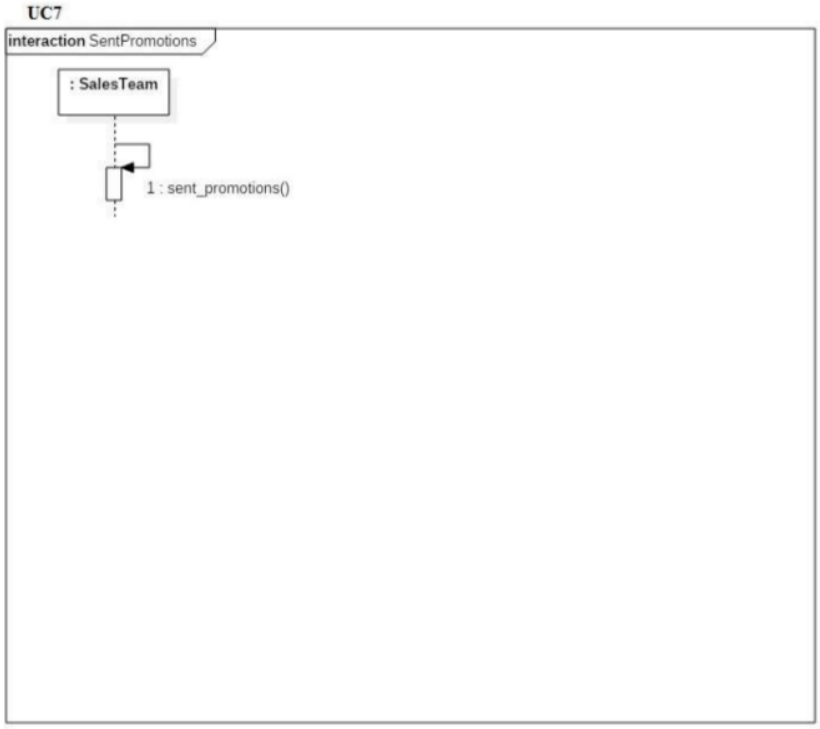
e) Use Case 5: IT Team Add/Remove external vendors in the system. [UC5]



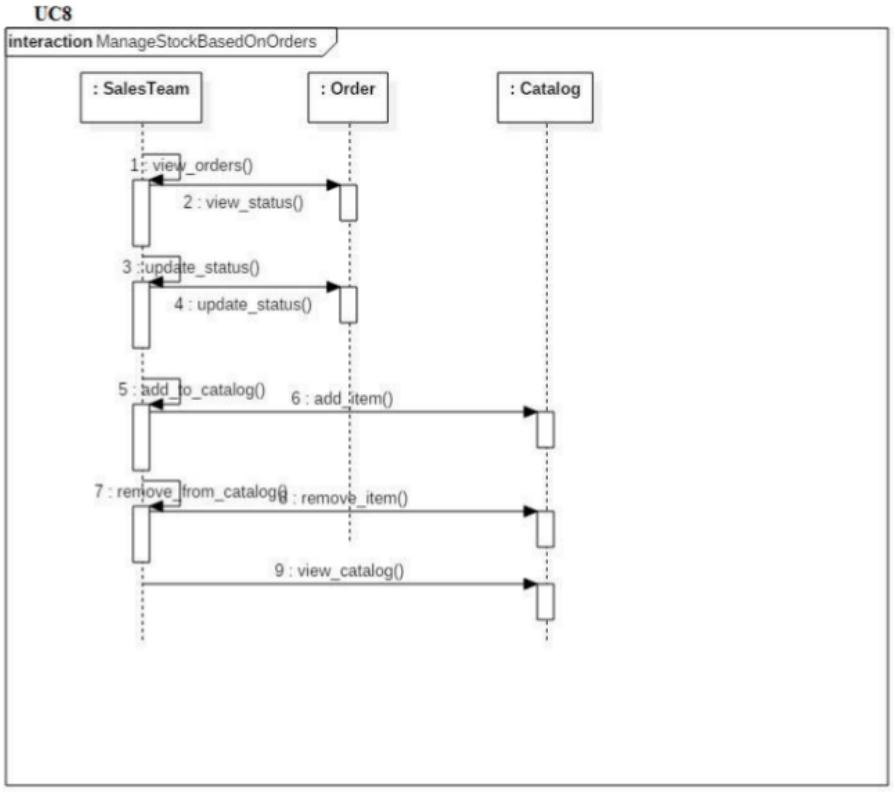
f) Use Case 6: Sales Team performs approval of returns [UC6]



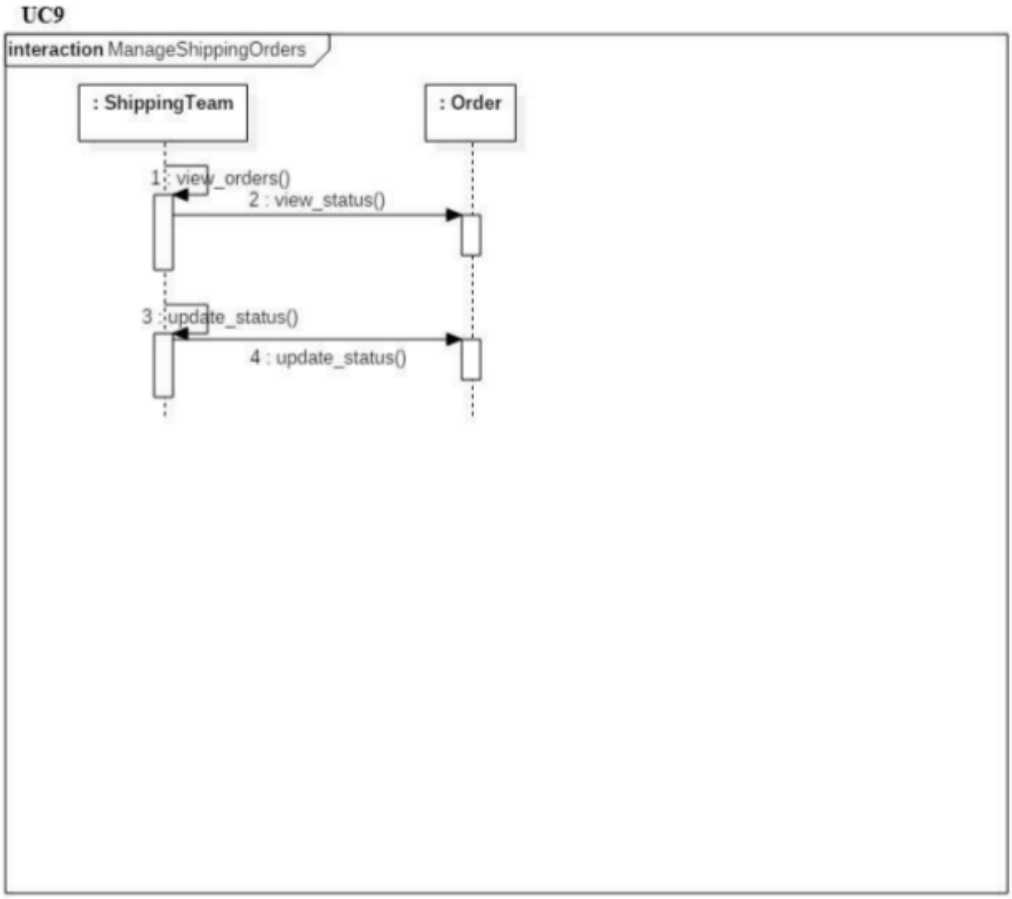
g) Use Case 7: Sales Team Sent promotions, discount to privileged Customer. [UC7]



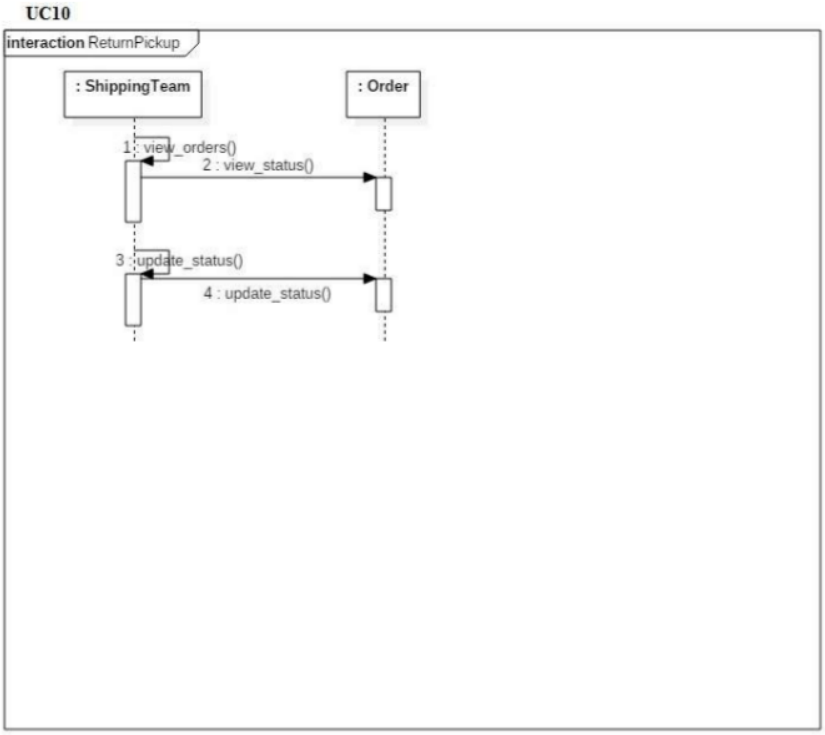
h) Use Case 8: Sales Team maintain the stock of physical store in the system. [UC8]



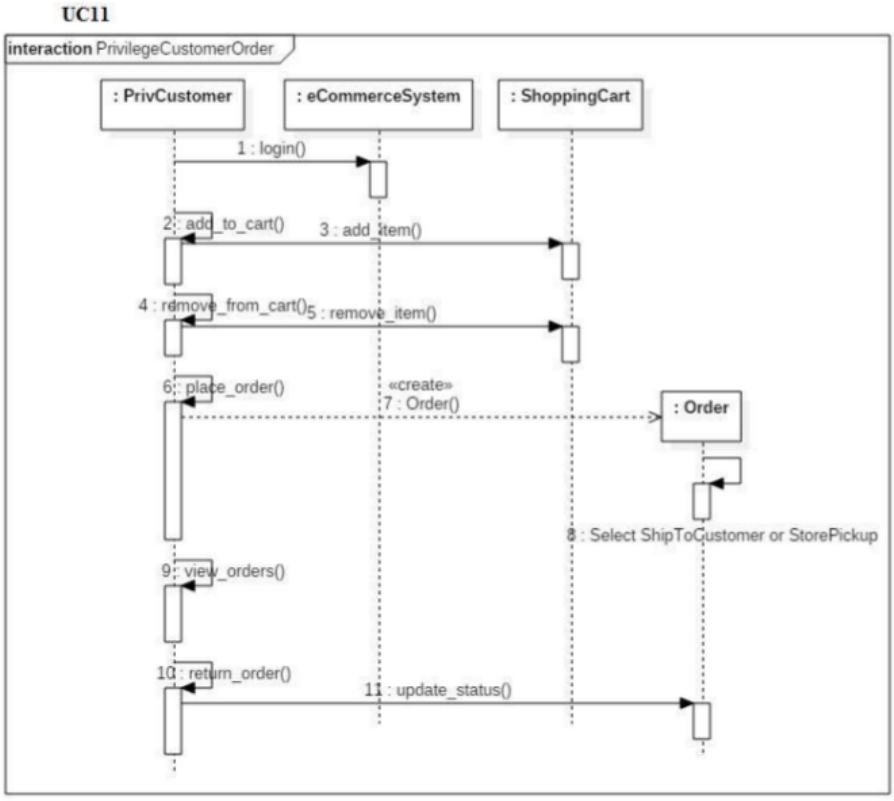
i) Use Case 9: Shipping Team Manages the shipping to orders. [UC9]



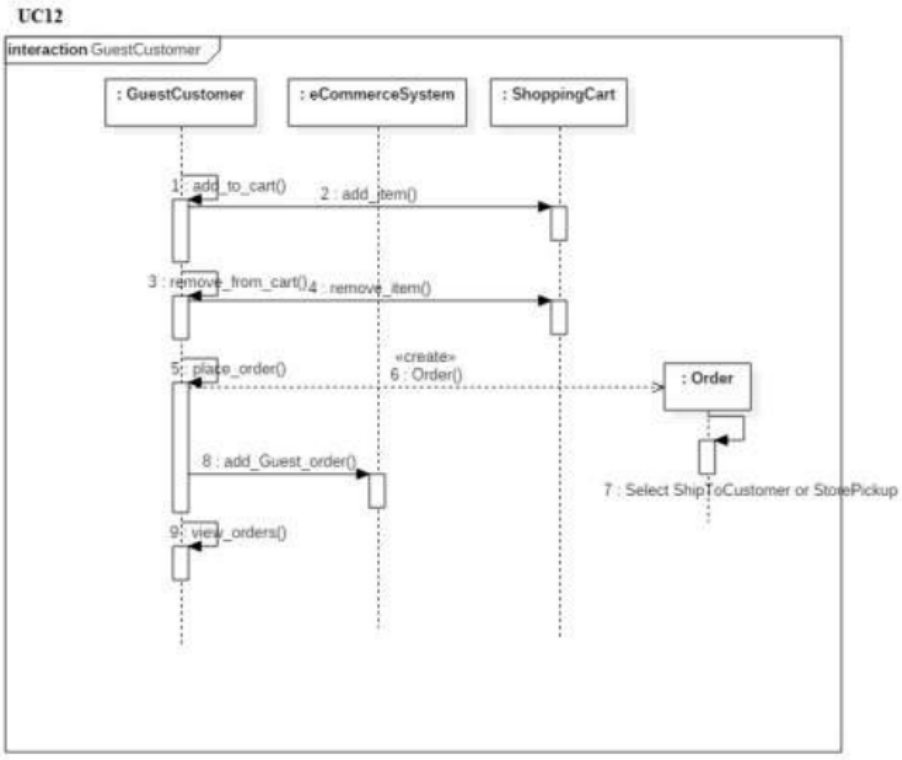
j) Use Case 10: Shipping Team Manages the return pickup [UC10]



k) Use Case 11: Privileged Customer Login to system and order/return items [UC11]



l) Use Case 12: Guest Customer Order without logging in. [UC12]



Use-Case Table

	eCommerceSystem	Catalog	Item	ExternalVendors	ShoppingCart	Customer	Order	StorePickup
UC1		view_catalog()						
UC2	register_customer()							
UC3	add_to_catalog() remove_from_catalog() view_catalog()	view_items()		add_to_catalog() remove_from_catalog()				
UC4	add_to_catalog() remove_from_catalog()	add_item() remove_item()				view_order()	get_order_status()	
UC5				get_vendor_id()		view_order()	get_order_status()	
UC6						get_custId() get_orders()	update_order_status() get_orderId()	
UC7								
UC8		add_add() remove_items() view_items()					view_status() update_status()	
UC9						get_orders()	view_status() update_status() get_orderId() update_order_status()	
UC10						get_orders()	view_status() update_status() getOrderId() update_order_status()	
UC11	Login() view_catalog()		calc_sub_total()		add_item() remove_item()	add_to_cart() remove_from_cart() view_order() place_order() return_order()	update_status() calculate_order_total() add()	select_store()
UC12	view_catalog()		calc_sub_total()		add_item() remove_item()	add_to_cart() remove_from_cart() view_order() add_to_cart() view_cart())	calculate_order_total()	select_store()

	StoreUnit	ShippingTeam	ITTeam	SalesTeam	GuestCustomer	PrivCustomer	ShipToCustomer	ShippingInfo
UC1								
UC2								
UC3								
UC4	view_orders()							
UC5			add_vendors() remove_vendors()					
UC6				view_orders() update_status()				
UC7				sent_promotions()				
UC8				view_order() add_to_catalog()				
UC9		view_order() update_status()						
UC10		view_order() update_status()						
UC11						place_order() return_order()	set_shipping_info()	
UC12					place_order()		set_shipping_info()	

Remark

We have completed most of the use cases as stipulated in the Use case diagram. But the e-commerce system is a complex system with much dependencies on the front-end as well as the back-end, hence we think that a User Interface is a requirement for this project. At present the project is driven by the driver classes and all the test are verified. This project is in the phase where it is ready to be interfaced with a User Interface which would enhance this project further.

The project was done using use-case driven methodology. This methodology helped us to ensure completeness and robustness at each phases of development. Use-case methodology provided us with a overall complete blue print of how the project would look like from the initial phase itself. We were able to identify the class relationships, interfaces even the complexity of each classes even before we started the coding. The idea of starting implementation with the riskiest class helped us from getting into a situation where we would need to change the complete class structure at a near final phase. In short this approach ensured that the riskiest classes which can causes greater changes in class structures are in place at the initial coding phase itself which helped to ensure robustness to the project. This methodology also helped us identify and rectify errors in our initial implementation ideas without causes much interruption in the project development.

In short use-case driven methodology led to project development which accommodated scope for error corrections at each phase and still be able to complete the project on time with minimal structural variations in classes.