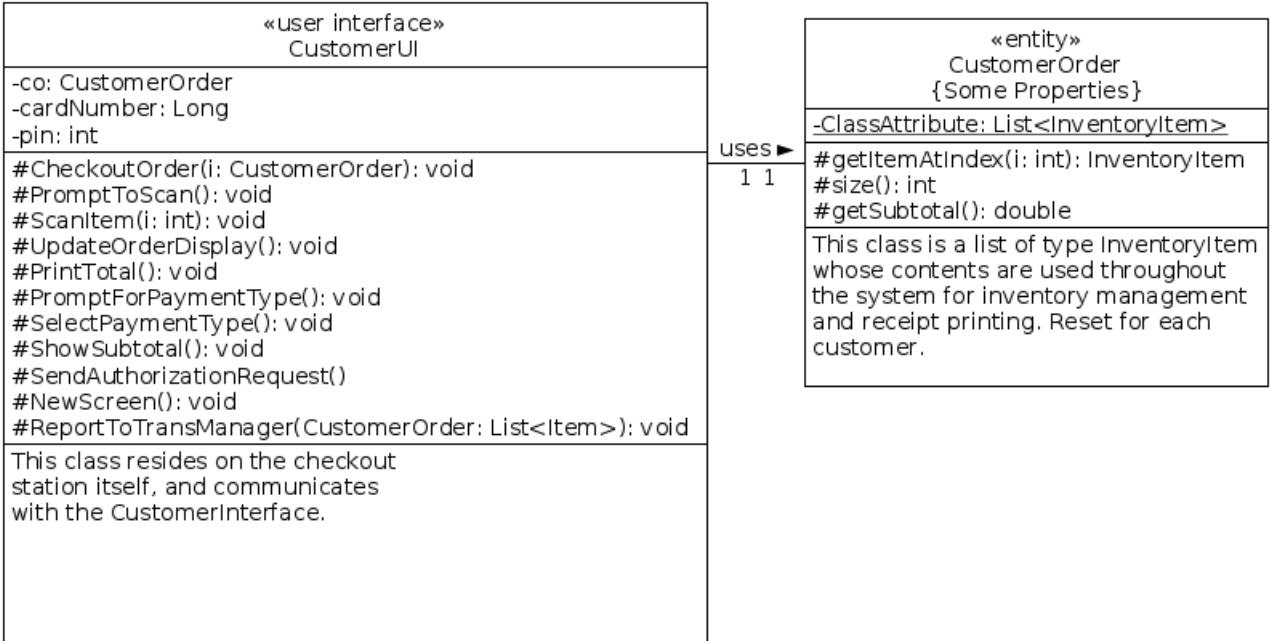
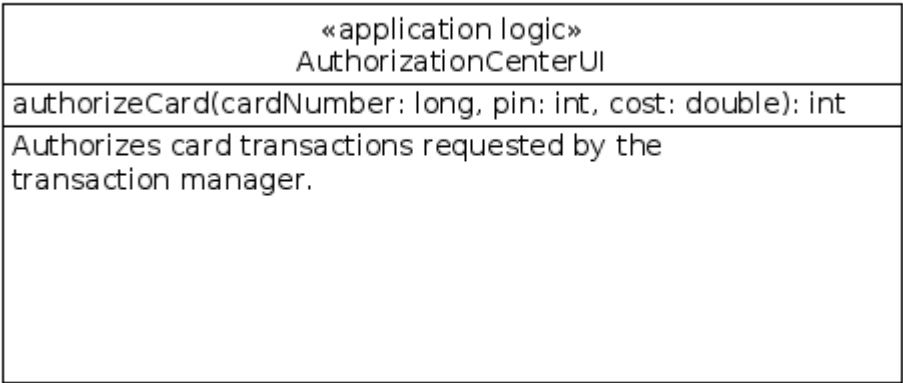
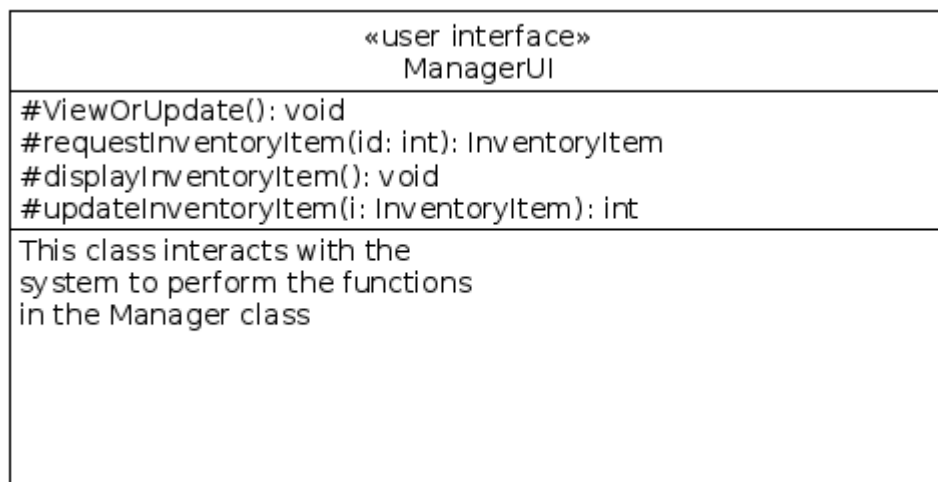
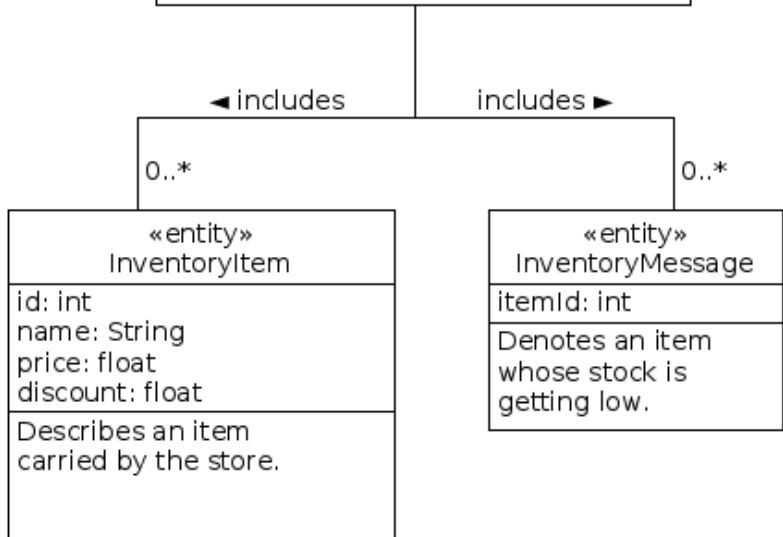
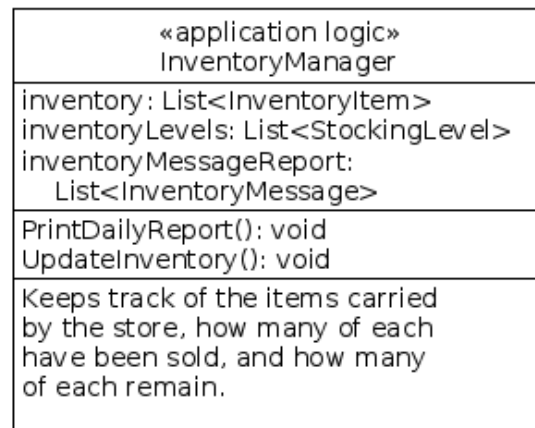
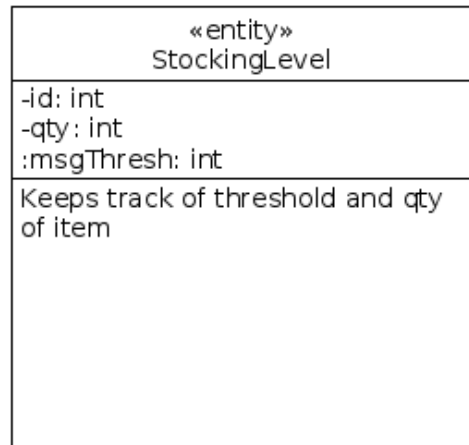


Static Models



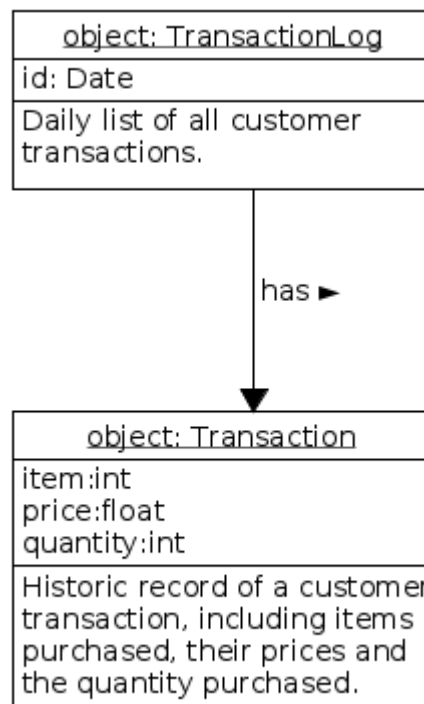


«entity» Printer
#Print(s: string): int
Printer object sends signal to printer to print, and returns success/error code

«user interface» RestockerUI
#PromptForScan(): void #RestockItem(i: int): int #UpdateQuantity(i: int): void #AddNewItem(i: InventoryItem): void
This class initiates the restock use case associated with the actor Restocker This class interacts with the rest of the system to fulfill the restock use case

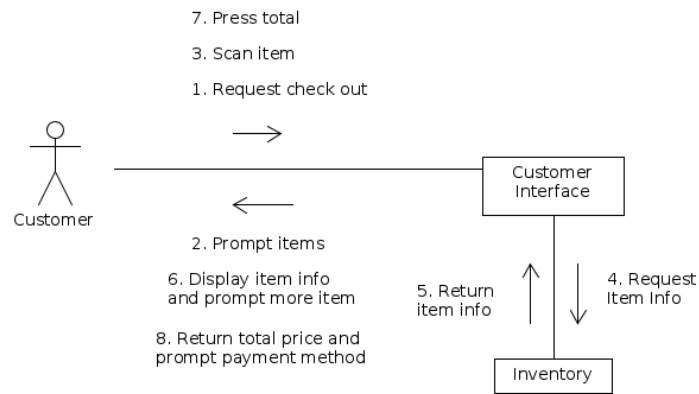
«control» Timer
#PrintDailyReport(): void #PrintInventoryMessage(): void
At midnight, interacts with InventoryManager and TransactionManager to print inventory message and daily report

«application logic» TransactionManager
-transLog: TransactionLog
#GenerateAndPrintTransactionReport(s: string): int
Generates and prints transaction reports.

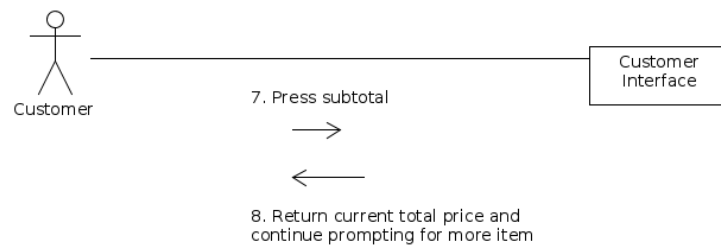


Dynamic Models

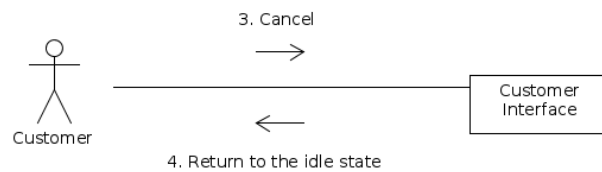
Common Checkout Order



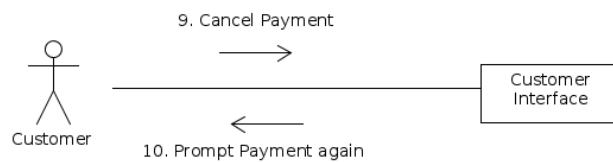
Subtotal



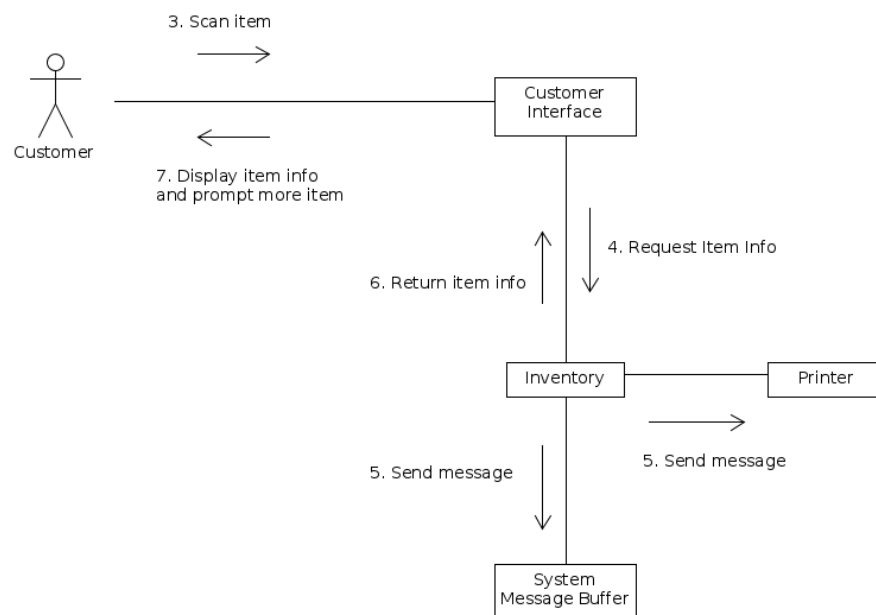
Cancel Checkout



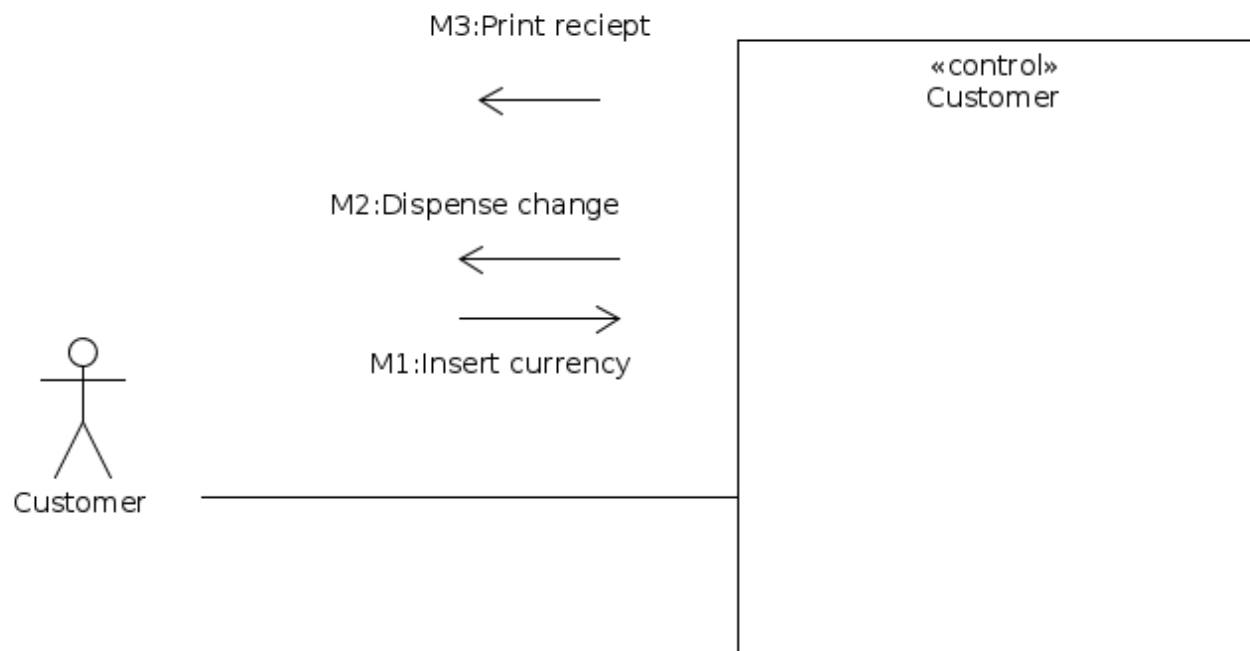
Cancel Payment



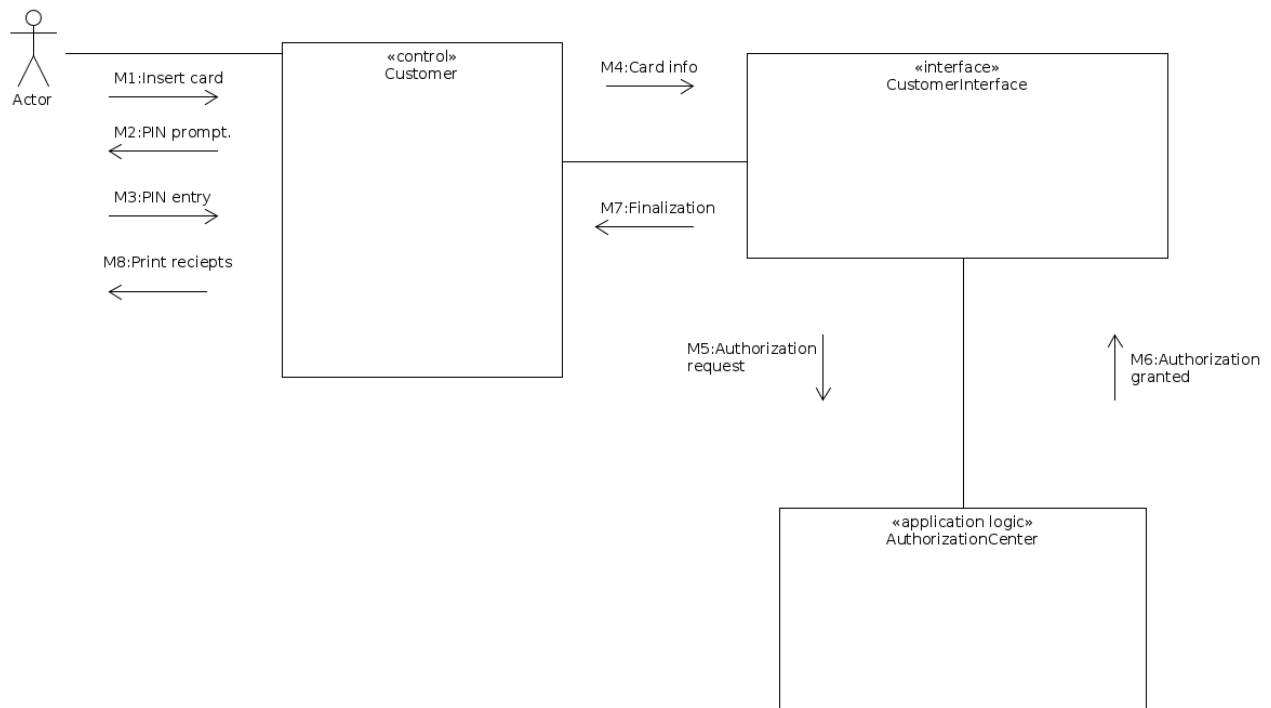
Send Inventory Message



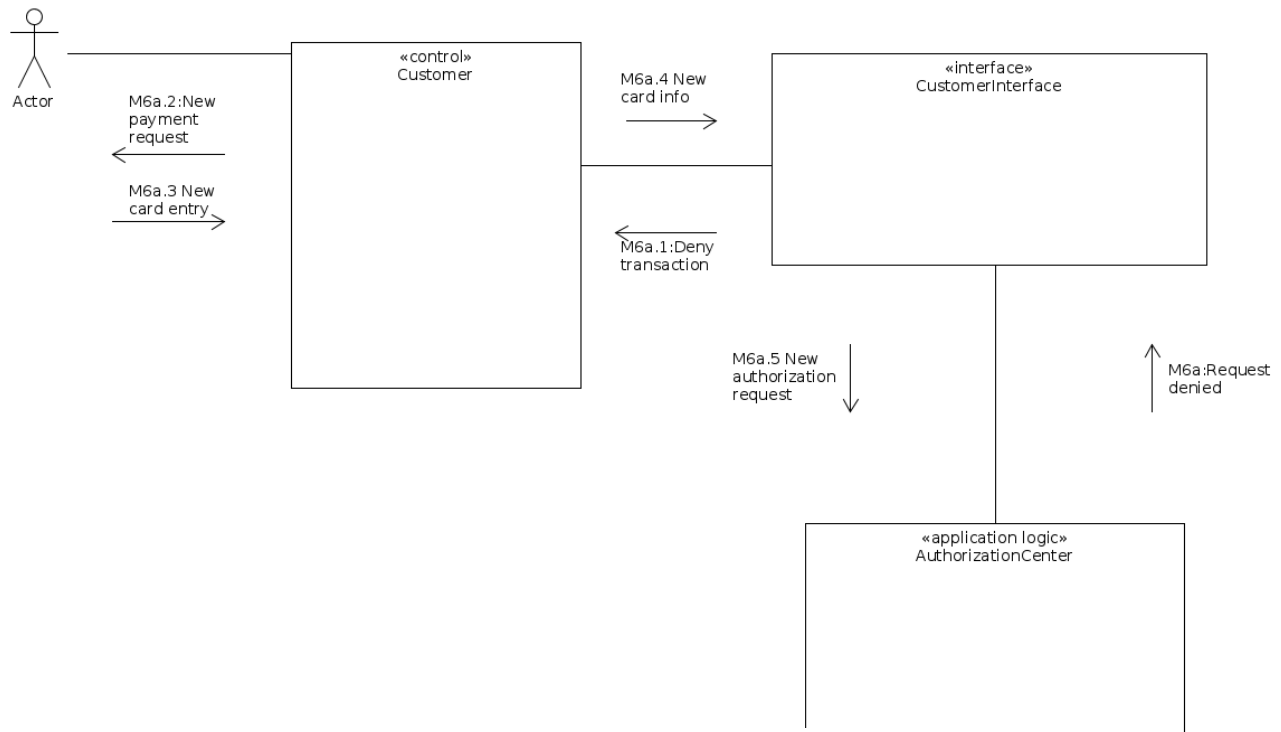
Pay By Cash use case



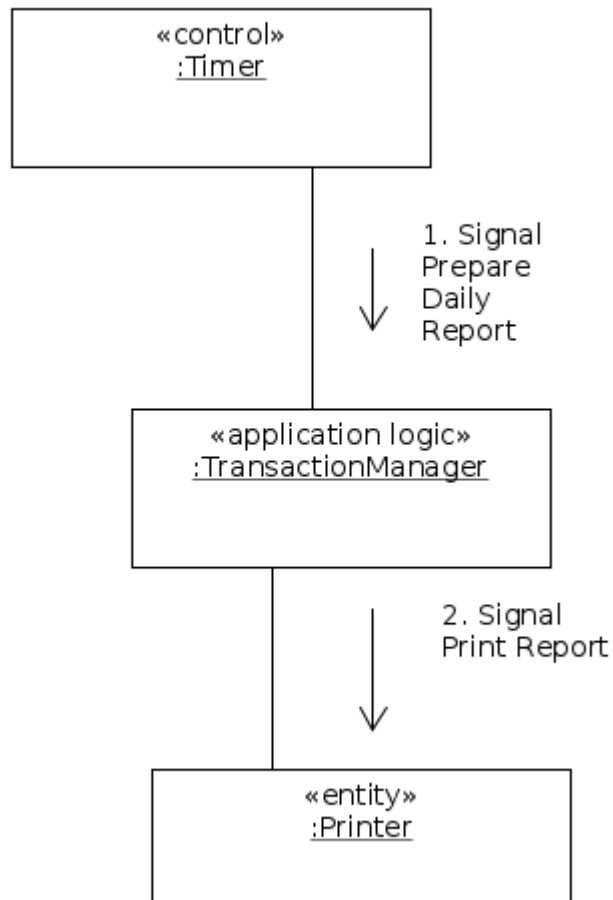
Pay by credit card(Approved) use case.



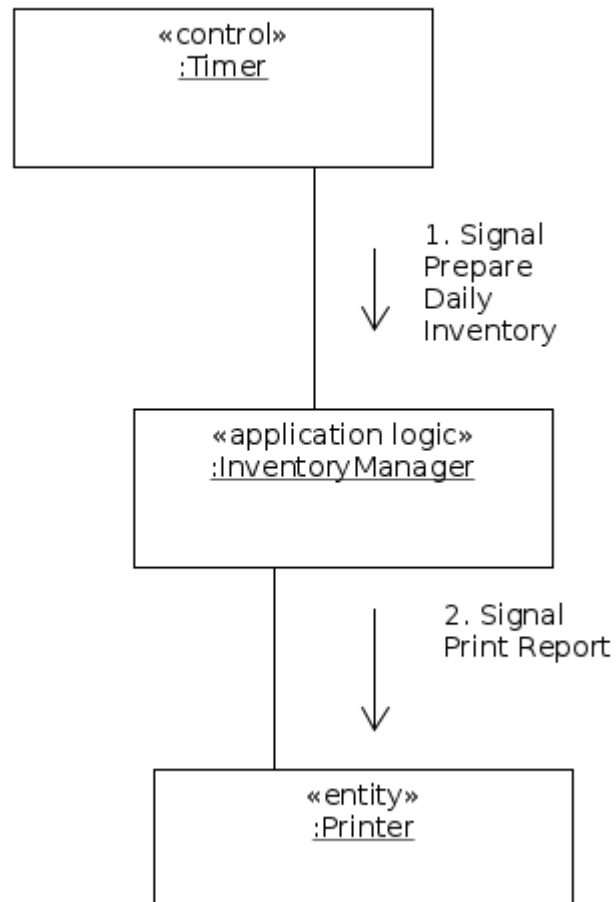
Pay by credit card(Declined) use case.



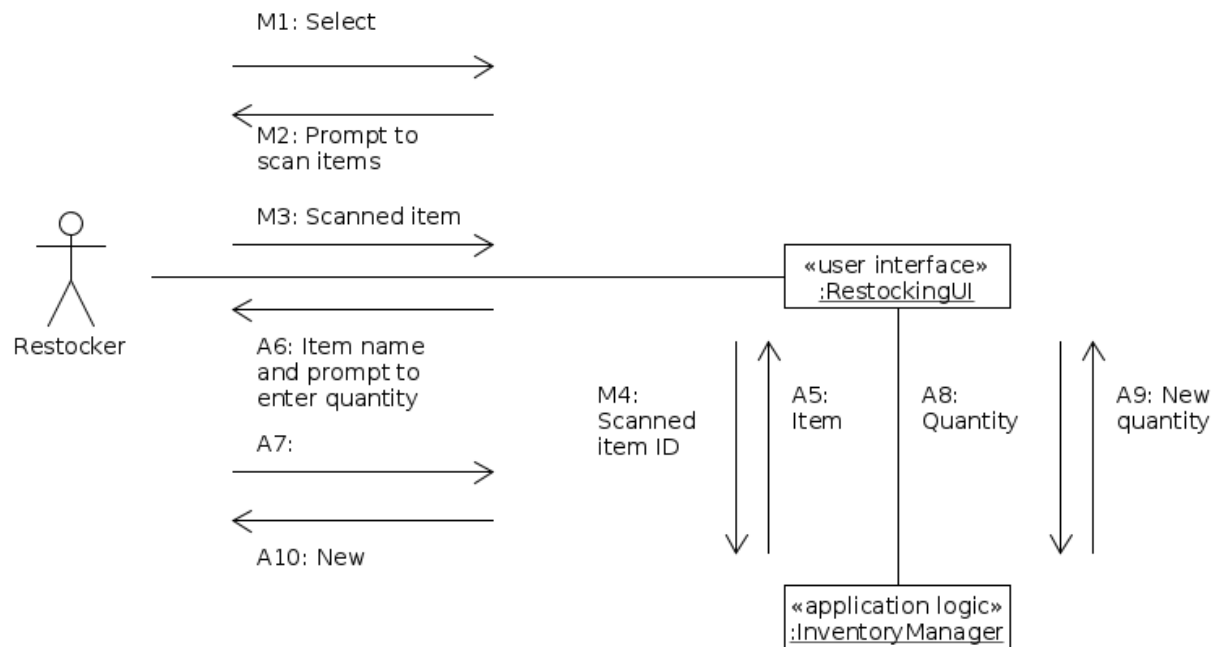
Print Daily Report Use Case



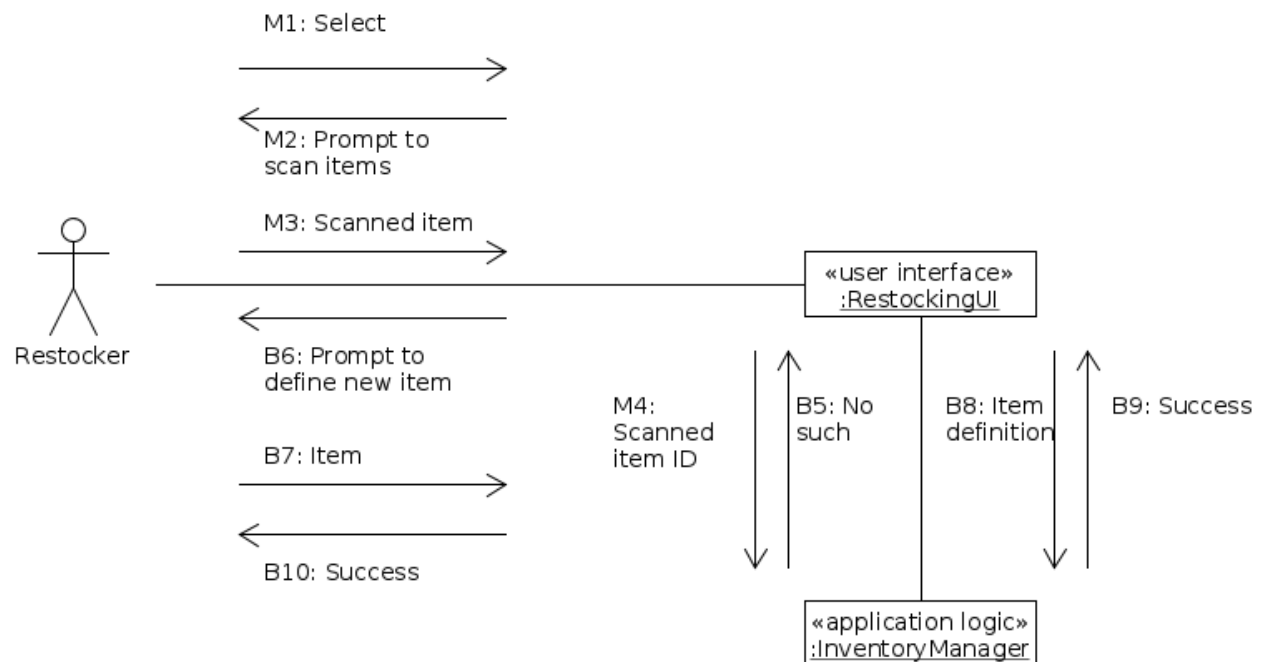
Print Inventory Message Use Case



Restock Inventory use case



Stock New Item use case



View or Update Inventory use case

