SUPERMARKET AUTOMATION SOFTWARE v1.0

Software analysis and design

Sachin Kumar 11CS30043

Siddharth Rakesh 11CS30036

Table of contents

CLASS DIAGRAM
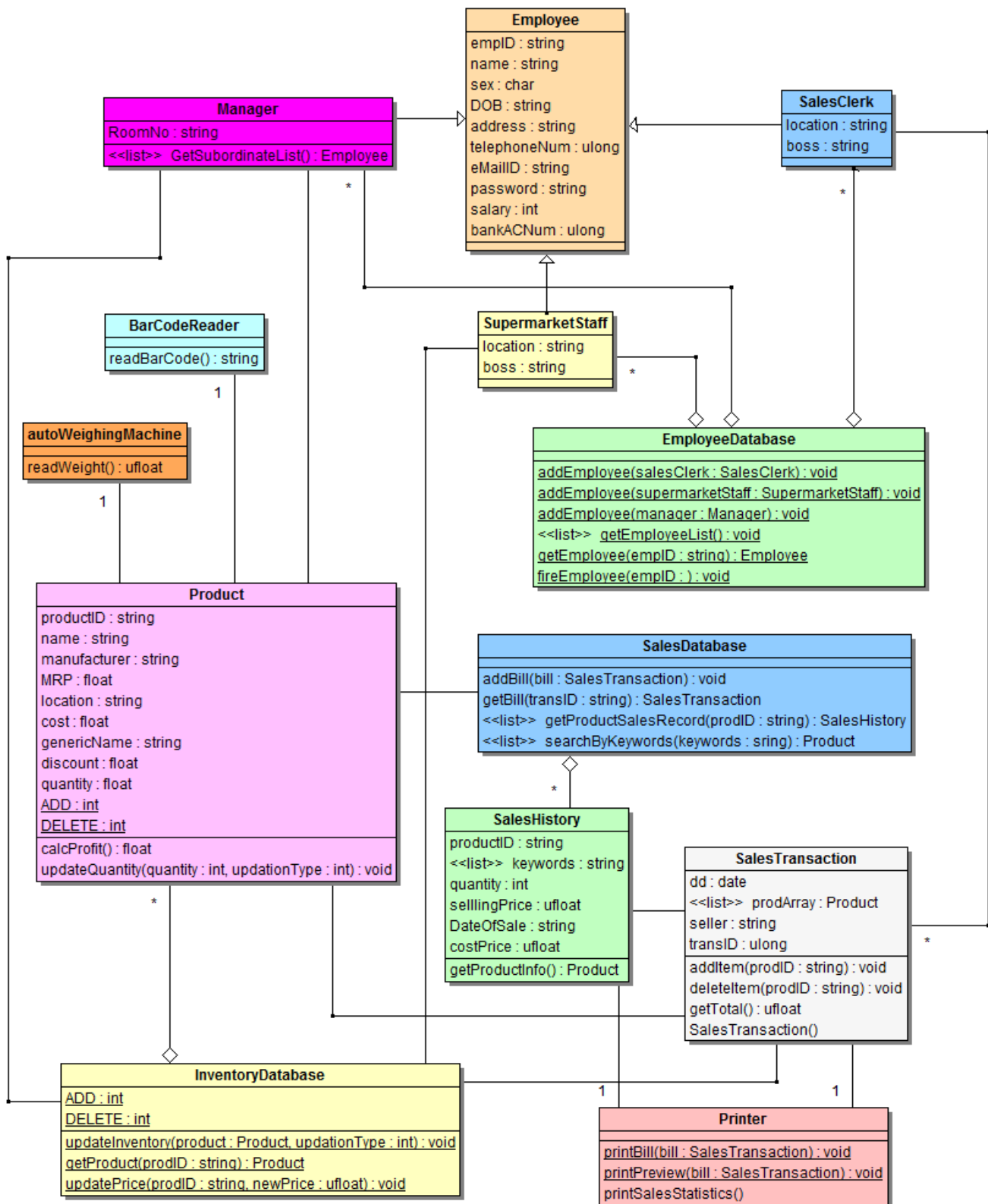
Employee

empID : string
name : string
sex : char
DOB : string
address : string
telephoneNum : ulong
eMailID : string
password : string
salary : int
bankACNum : ulong

Manager

RoomNo : string

<<list>> GetSubordinateList() : Employee

SalesClerk

location : string
boss : string

BarCodeReader

readBarCode() : string

1

SupermarketStaff

location : string
boss : string

*

autoWeighingMachine

readWeight() : ufloat

1

EmployeeDatabase

addEmployee(salesClerk : SalesClerk) : void
addEmployee(supermarketStaff : SupermarketStaff) : void
addEmployee(manager : Manager) : void
<<list>> getEmployeeList() : void
getEmployee(empID : string) : Employee
fireEmployee(empID :) : void

Product

productID : string
name : string
manufacturer : string
MRP : float
location : string
cost : float
genericName : string
discount : float
quantity : float
ADD : int
DELETE : int

calcProfit() : float
updateQuantity(quantity : int, updationType : int) : void

*

SalesDatabase

addBill(bill : SalesTransaction) : void
getBill(transID : string) : SalesTransaction
<<list>> getProductSalesRecord(prodID : string) : SalesHistory
<<list>> searchByKeywords(keywords : sring) : Product

*

SalesHistory

productID : string
<<list>> keywords : string
quantity : int
selllingPrice : ufloat
DateOfSale : string
costPrice : ufloat

getProductInfo() : Product

SalesTransaction

dd : date
<<list>> prodArray : Product
seller : string
transID : ulong

addItem(prodID : string) : void
deleteItem(prodID : string) : void
getTotal() : ufloat
SalesTransaction()

*

InventoryDatabase

ADD : int
DELETE : int

updateInventory(product : Product, updationType : int) : void
getProduct(prodID : string) : Product
updatePrice(prodID : string, newPrice : ufloat) : void

1

1

Printer

printBill(bill : SalesTransaction) : void
printPreview(bill : SalesTransaction) : void
printSalesStatistics()
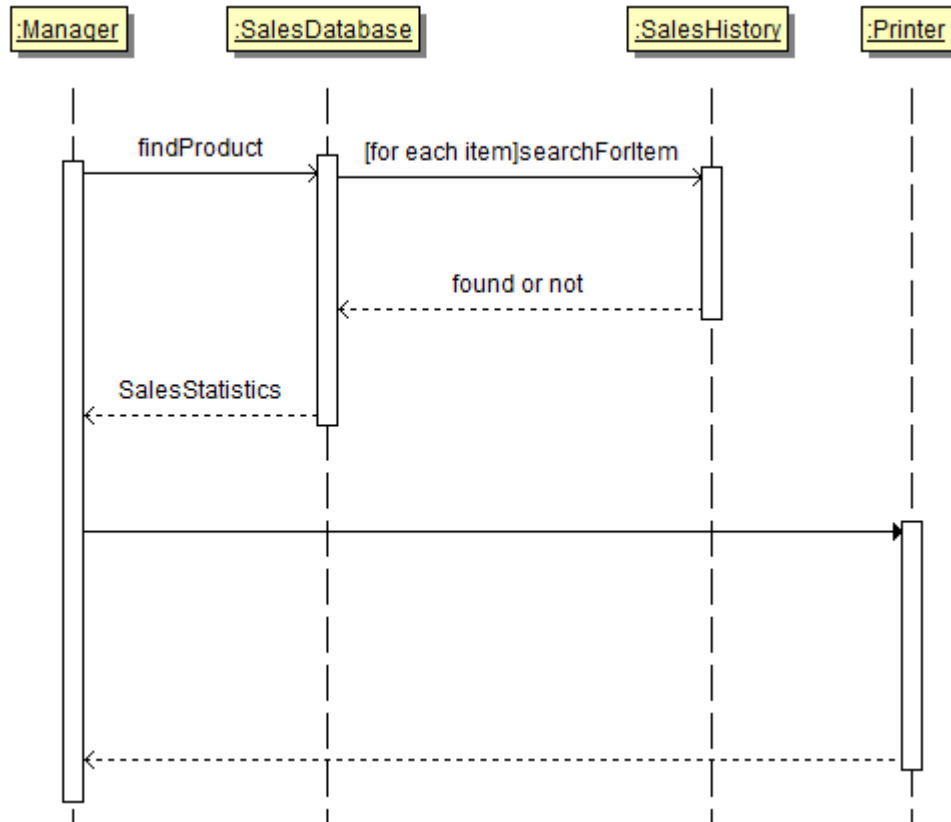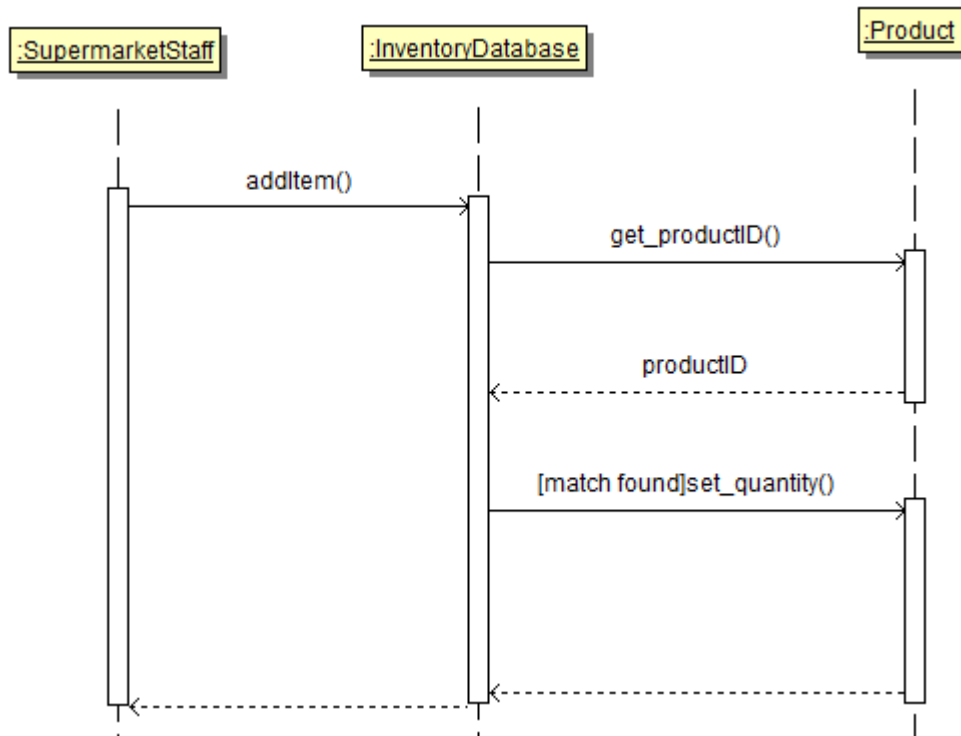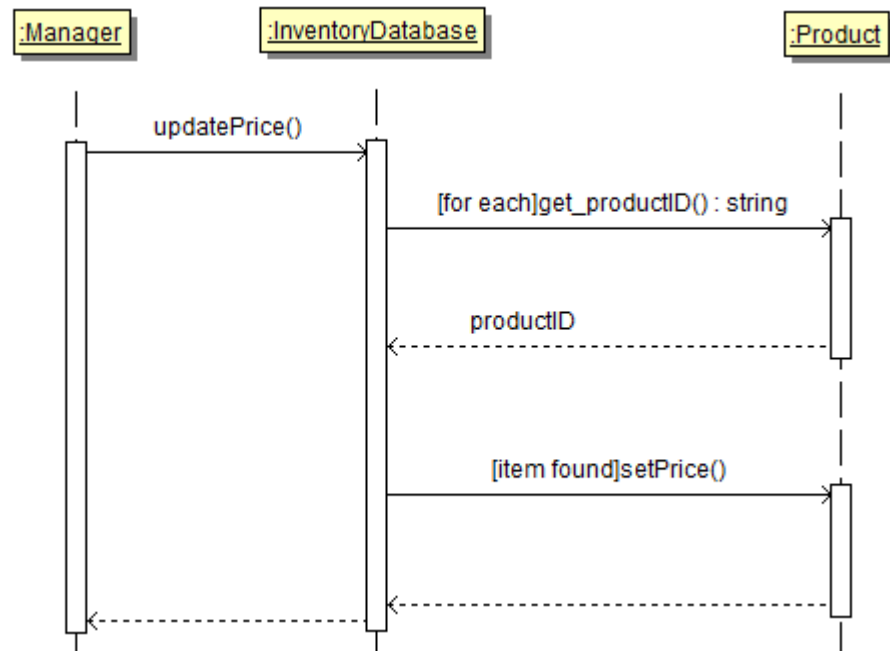
SEQUENCE DIAGRAMS
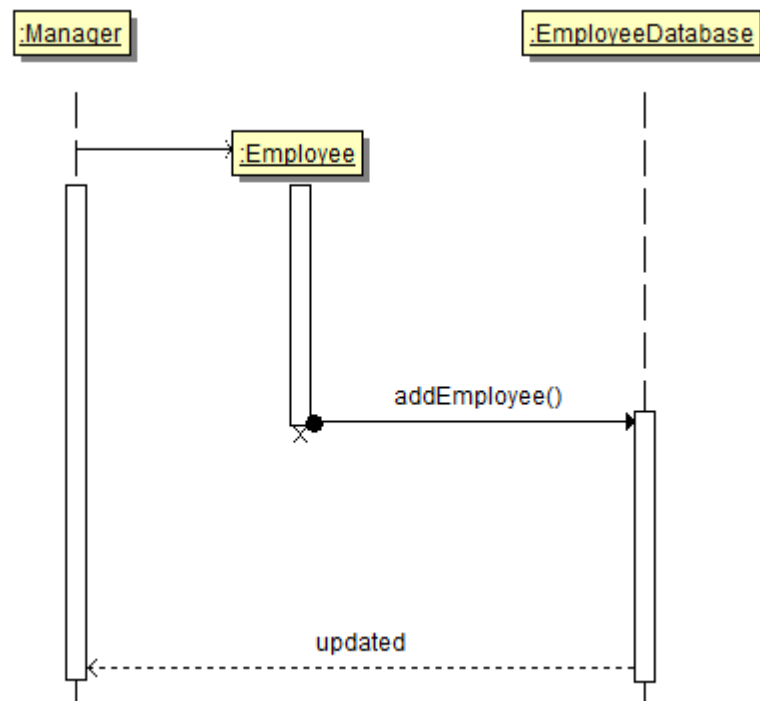
1. Sales Transaction
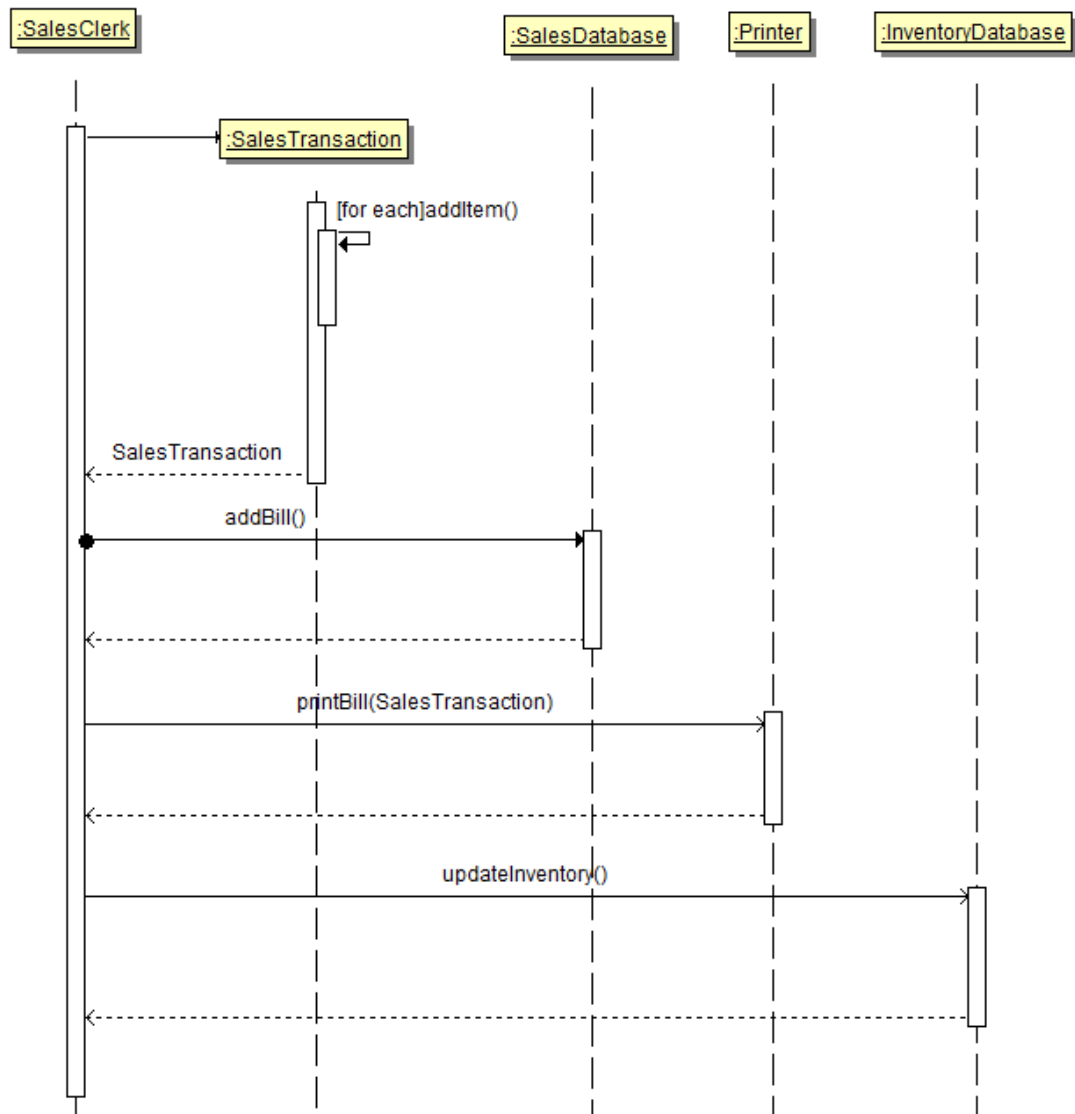
2. View Sales Statistics

3. Update Inventory

4. Update Price

:Manager :InventoryDatabase :Product

updatePrice()

[for each]get_productID() : string

productID

[item found]setPrice()

5. Add Employee

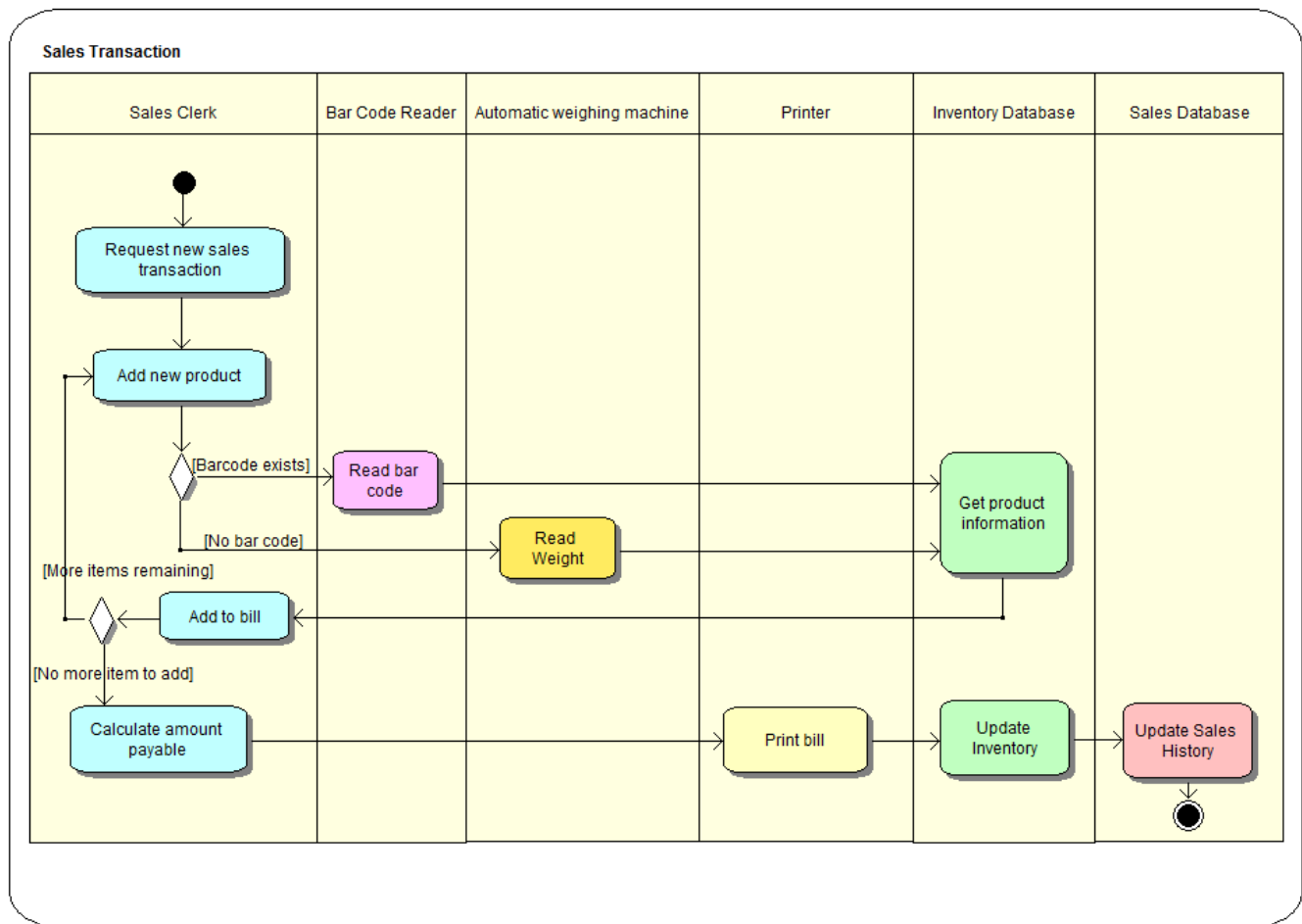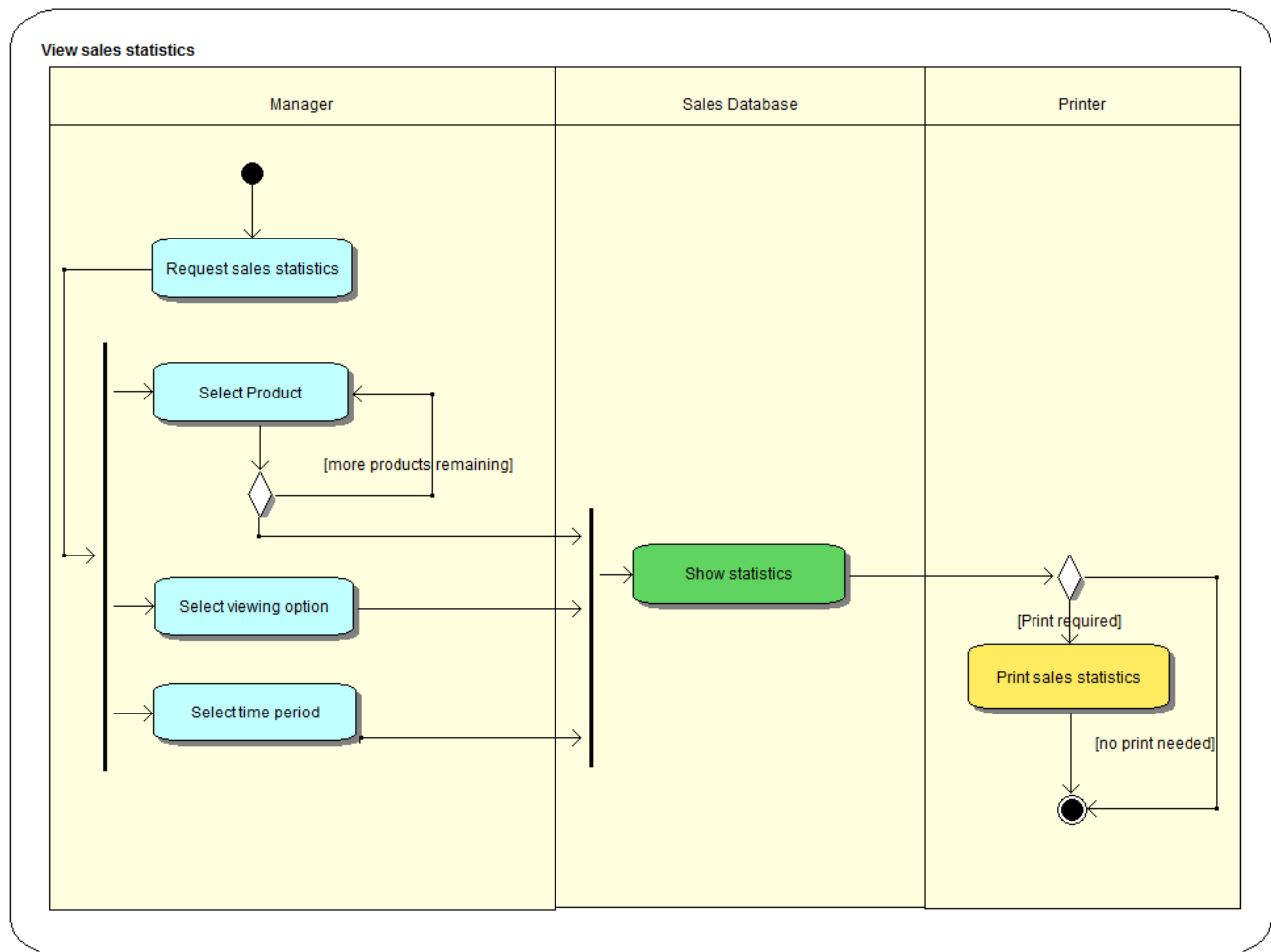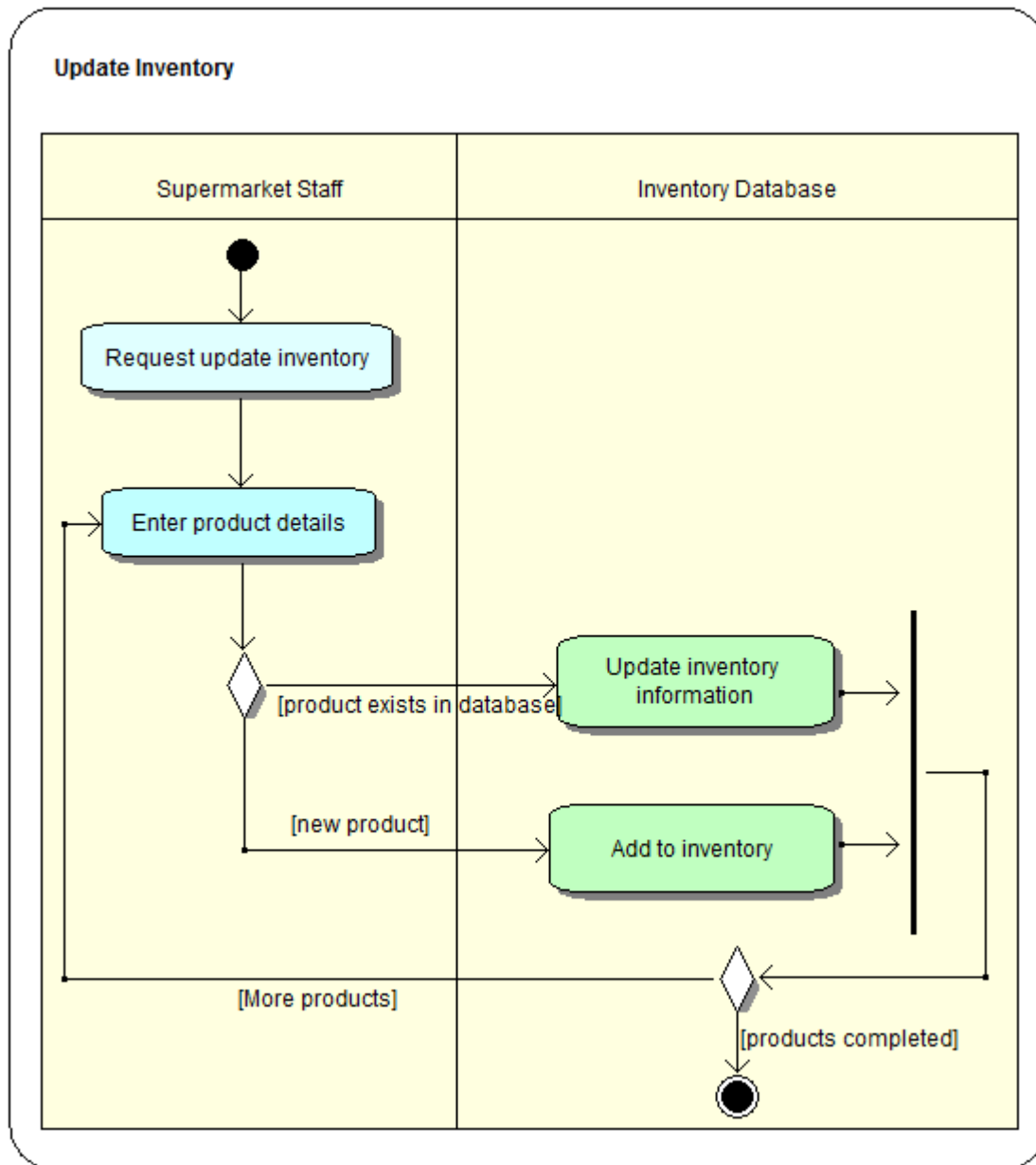:Manager :EmployeeDatabase

:Employee

addEmployee()

updated

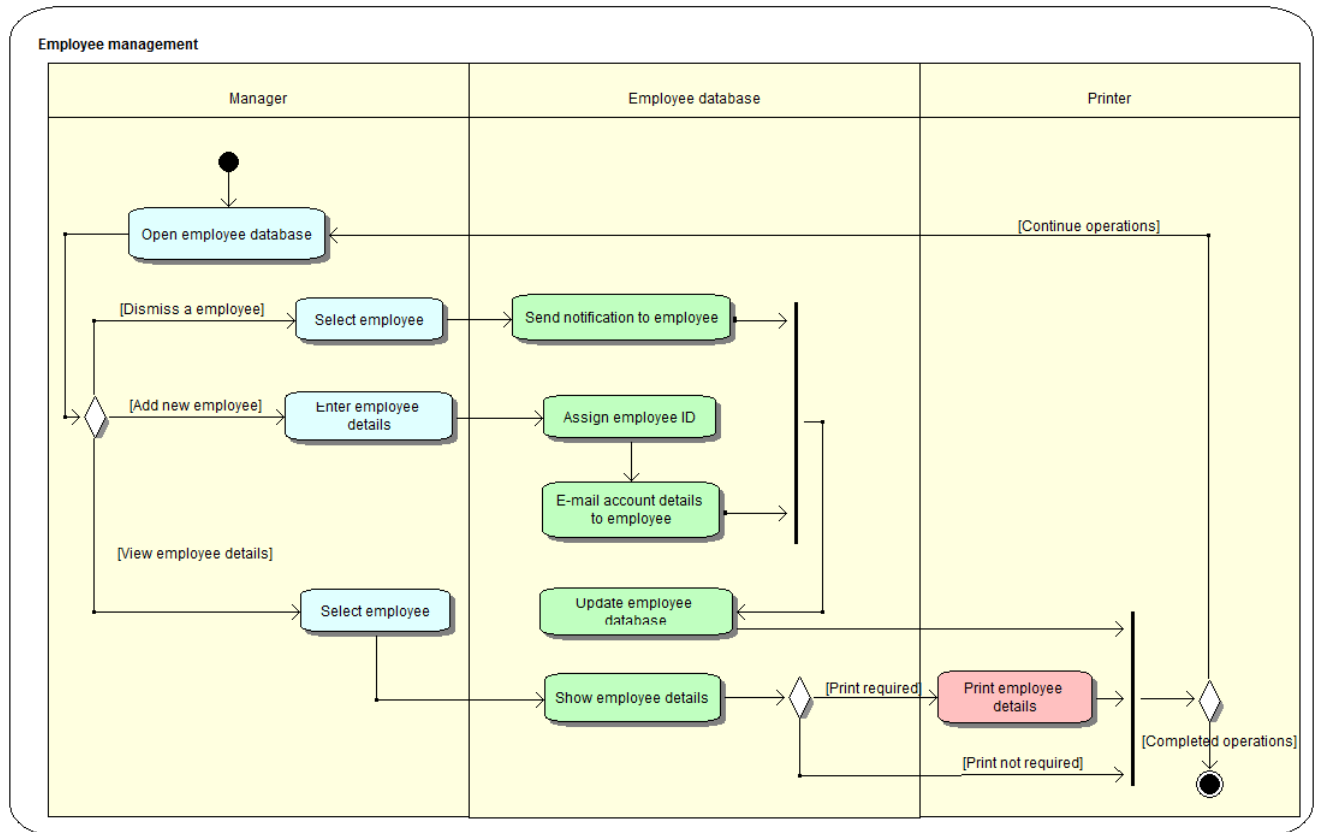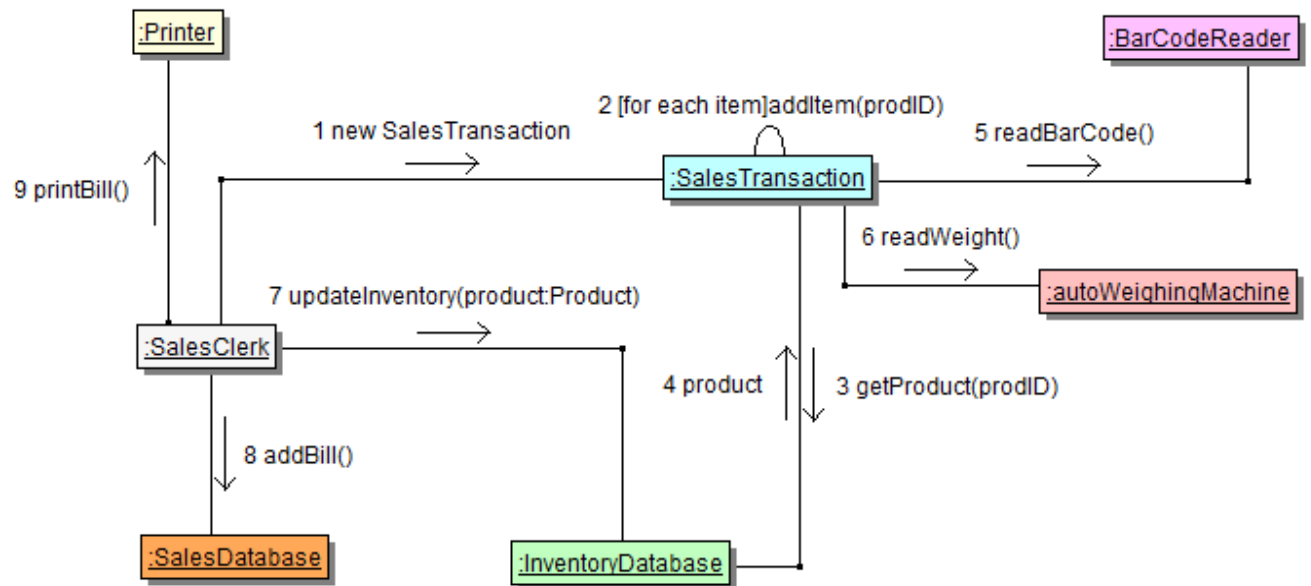# ACTIVITY DIAGRAMS

1. Sales Transaction

2. View Sales Statistics

View sales statistics

Manager	Sales Database	Printer
Request sales statistics		
Select Product		
[more products remaining]		
Select viewing option	Show statistics	[Print required]
Select time period		Print sales statistics
		[no print needed]

3. Update Inventory

Update Inventory

Supermarket Staff	Inventory Database
Request update inventory	
Enter product details	
[product exists in database]	Update inventory information
[new product]	Add to inventory
[More products]	[products completed]

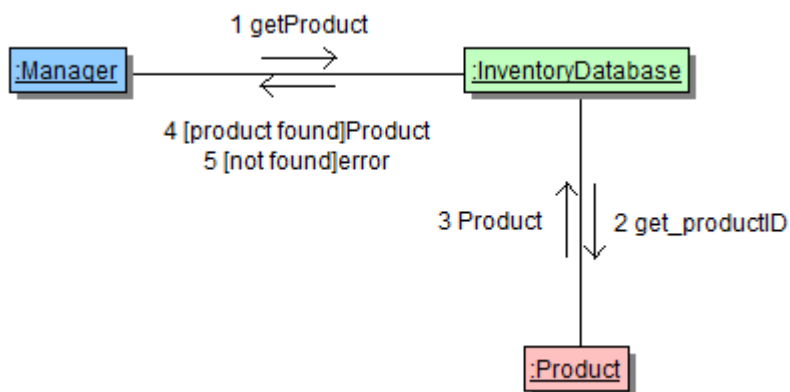# 4. Update Price

5. Manage Employees

Employee management

COMMUNICATION DIAGRAMS

1. Sales Transaction

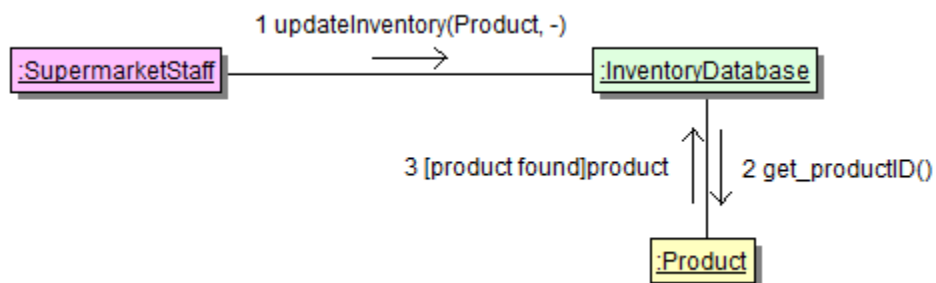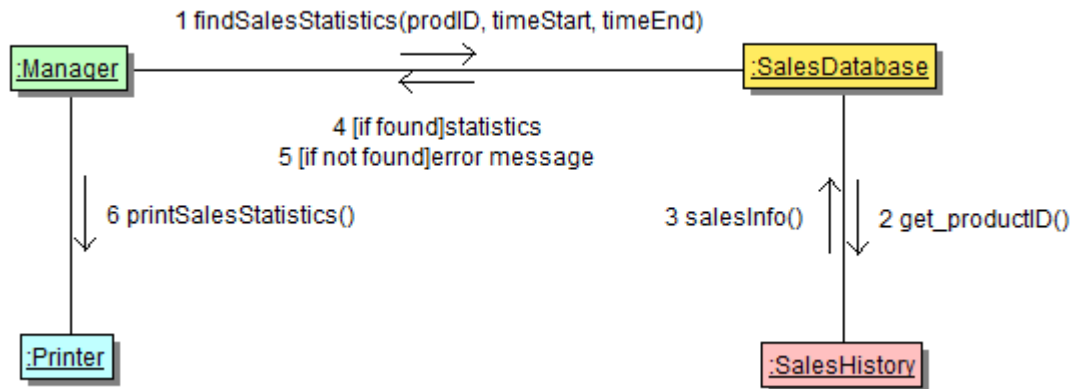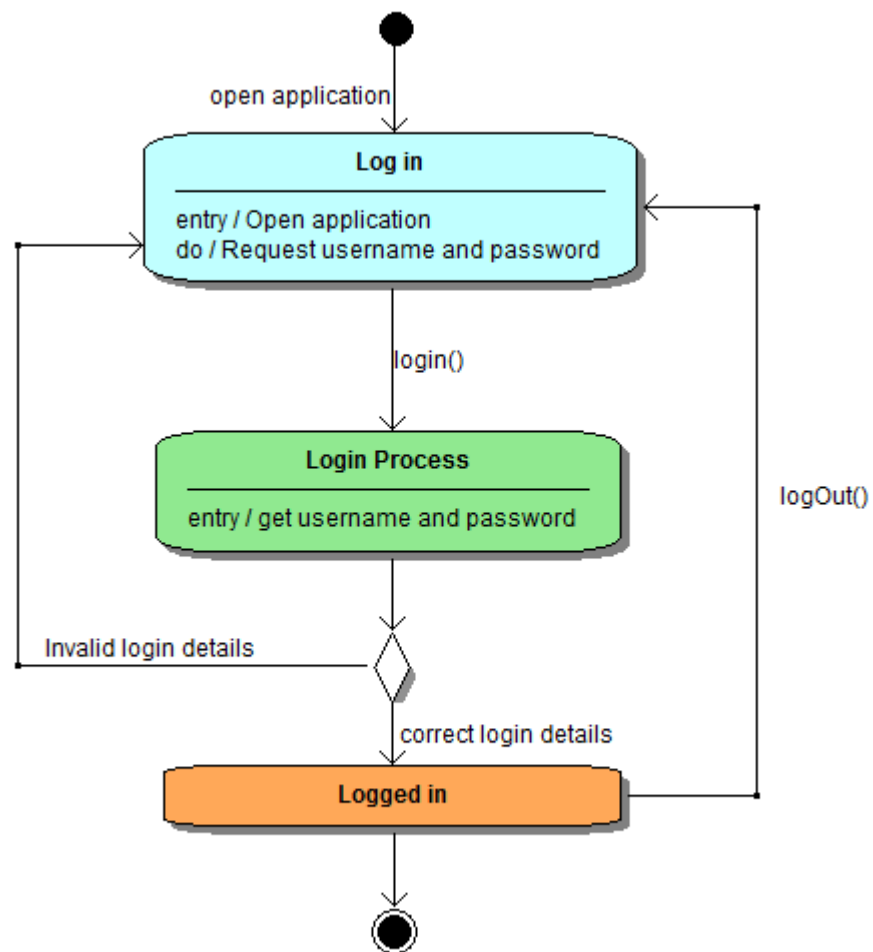## 2. Search Inventory

3. Update Price

1 updatePrice(prodID, newPrice)

:Manager → :InventoryDatabase

3 product

2 get_productID
4 set_MRP(new value)

:Product

4. Update Stock

1 updateInventory(Product, -)

:SupermarketStaff → :InventoryDatabase

3 [product found]product

2 get_productID()

:Product

5. Add Employee

1 findSalesStatistics(prodID, timeStart, timeEnd)

:Manager → :SalesDatabase

4 [if found]statistics
5 [if not found]error message

6 printSalesStatistics()

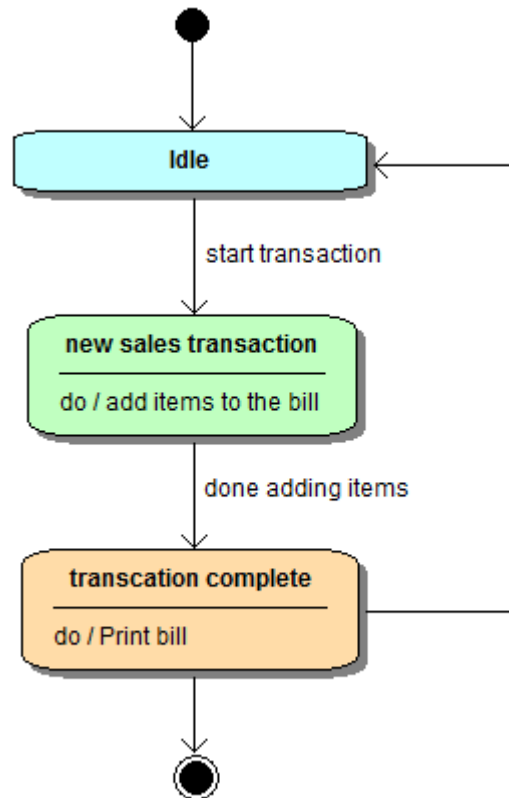3 salesInfo()   2 get_productID()

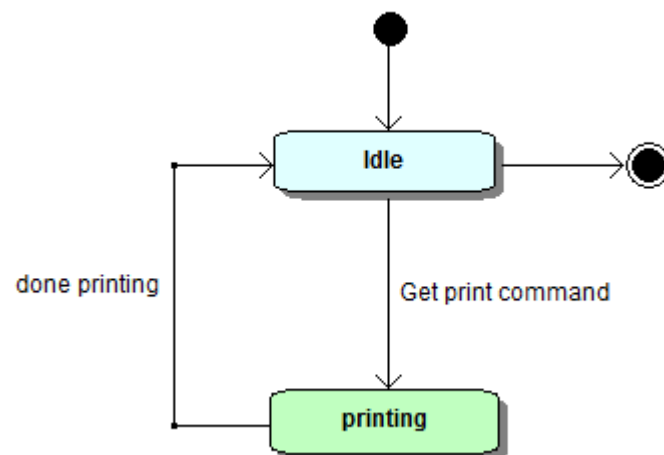:Printer

:SalesHistory

STATE CHART DIAGRAMS
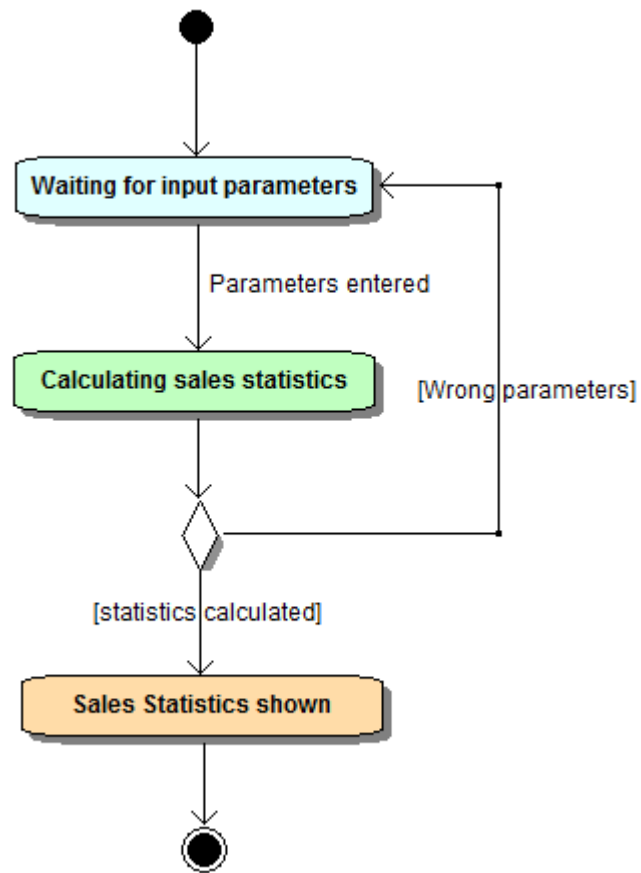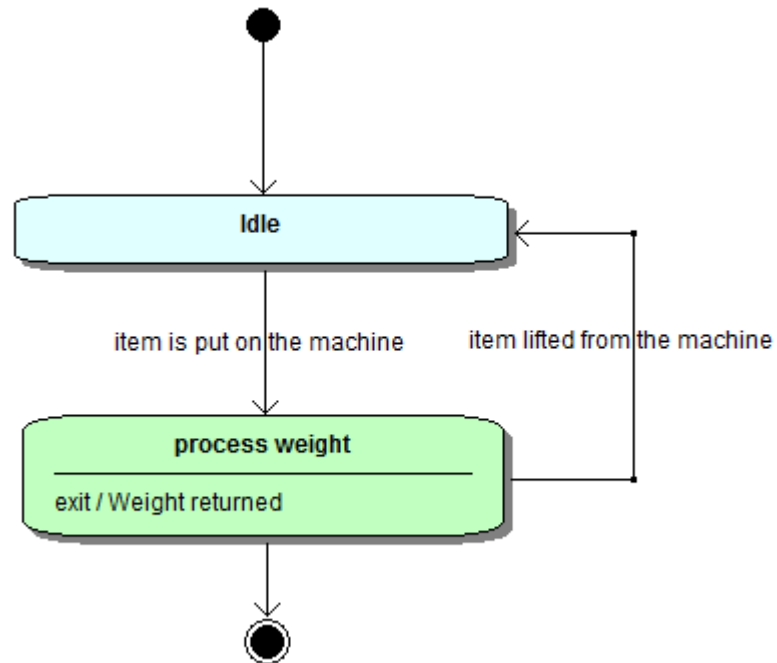
1. Log In

2. Sales Transaction

3. Printer

4. Sales Statistics

5. Automatic Weighing Machine

6. Bar code Reader