**Modeling UML Use Case diagrams and capturing Use Case scenarios.**

## B.1 Actors:

## 1. User (Employee/Manager/Owner)

## 2. Database

## B.2 Use cases:

## 1. Register

## 2. Login

## 3. Search

## 4. Check availability

## 5. Generate bill

## 6. Payment

## 7. Print/Display Bill

## 8. Manipulate Database

## B.3 Use Case diagrams:

## 

## B.4 Use Case Specifications:

## Precondition:

## Must have the software

## Must have a pre-existing account

## Post-Condition:

## Changes to database will be saved

1.Register

* Enter the details
* Create account

2. Login

* Enter correct username
* Enter correct password
* Authentication using the database
* Go to the specific user’s page

3. Search

* Enter the name of the item to be searched

4. Check availability

* Check database of inventory to see of item is available

5. Generate Bill

* Add items selected to the bill
* Add the costs of each item
* Give details of items selected
* Give total cost of all items selected

6. Payment

* Perform the payment transactions

7. Print/Display Bill

* Print the bill for the customer

8. Manipulate database

* Update the database when item is sold
* Allow employee to view inventory and edit customer database
* Allow manager to view inventory and edit customer and employee database
* Allow owner to view and edit inventory, customer, employee, supplier and manager database

**Aim: -** To draw the behavioral view diagram: Sequence diagram, Collaboration diagram

## B.1 Objects:

## Entity objects:

## Database (DB), User

## Boundary objects:

## UI

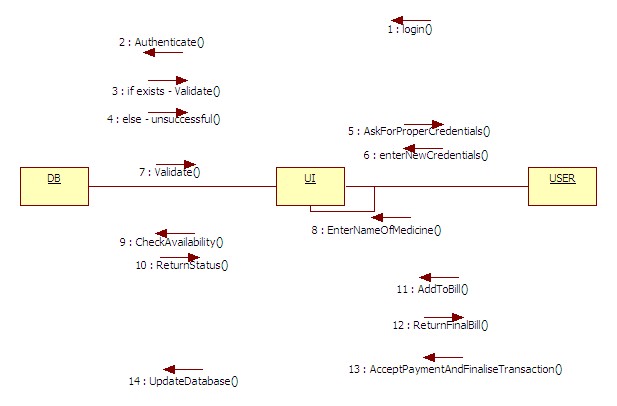
## Control objects:

## None (the logic behind UI)

## B.2 Sequence diagram:

## 

## B.3 Collaboration diagram:

****

**Modeling UML Class diagrams.**

## B.1 Actors:

## Database

## Person

## -Owner

## -Customer

## -Manager

## -Employee

## -Supplier

## B.2 Class Diagram:

## 

## B.3 Entity Classes:

## Supplier, Owner, Manager, Employee, Database(Inventory)

## B.4 Boundary Classes:

## Supplier software class

## B.5 Control Classes:

Shop

**Aim: -** To draw the behavioral view diagram: Activity diagram

## 

## SWIMLANE:

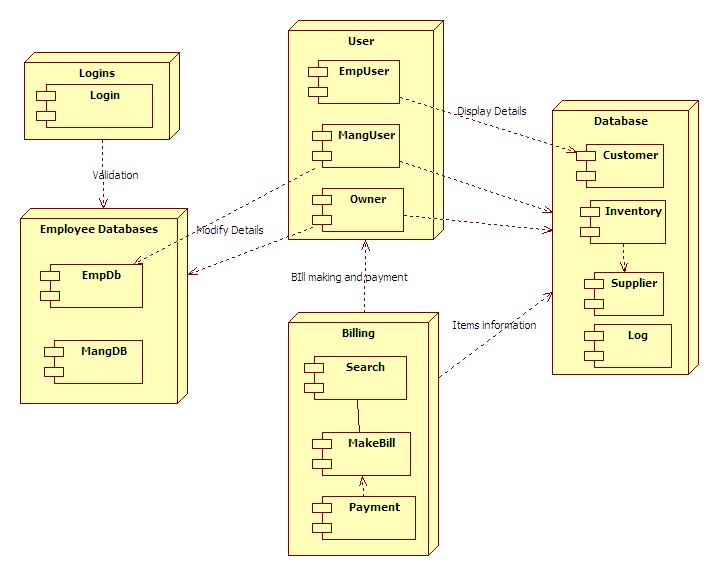
## 

**Aim: -** To draw the State Diagram

## 

## COMPONENT:

## Deployment Diagram

****