Supermarket Automation Software (SAS)
Structured Analysis (SA)
Structured Design (SD)
Document

1.Abhishek Panigrahi - 14CS10001
2.Siddhant Singh- 14CS10045

March 2016

Contents

1 CLASS DIAGRAM

The class diagram is the main building block of object-oriented modelling. It is used both for general conceptual modelling of the systematics of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

1.1 Brief Description

There are 11 main classes . Those include

1. Item

2. Employee

3. Sales Clerk

4. Manager

5. Employee Database

6. Database(basically it's inventory database)

7. Transaction

8. Item transaction history

9. AWM/C

10. BCR

11. Printer

The relationships among various classes have been shown below. The required attributes and methods of each class derived from use case diagram have been clearly mentioned in the use case diagram .

Item

- name : string
- bar_code : string
- manufacturer: string
- MRP: float
- cost : float
- discount : float
- quantity : float
- type : bool
- no. of transactions
- history : Item_transaction_history[1..n]

+ Item()
+ add_history(bill : Transaction) : void
+ see_history(time:int):void
+ get_type() : bool
+ update_quantity (x:float) :void
+ calculate_profit (): float
+ update_price(x:float,y:float,z:float):void
+ get_name():string

Employee

- name : string
- empl_id : string
- salary : long int
- email : string
- mobile : string
- date_of_join : date
- is_sales_clerk : bool

+ Employee()
+ def_is_sales_clerk()
+ request_update ():void
+ get_details() : void
+ match_empl_id(x:string) : bool

Employee_Database

- no_of_employee
- emparr : Employee[1..n]
- manager : Manager

+ addEmployee(Employee e):void
+ fireemp(empl_id:string) : void
+ addManager(Manager m):void
+ check_empl_id(x:string):bool
+ check_manager_id(x:string):bool

Manager

- name : string
- empl_id : string
- salary : long int
- email : string
- mobile : string
- date_of_join : date

+ query_statistics():void
+ query_inventory() : void
+ update_price (): void
+ get_details() : void
+ match_empl_id(x:string) : bool

BCR

+ read_barcode(Item) : string

AWM/C

+ measure_weight(Item):float

Sales_clerk

+ create_transaction(x : Item[1..n]):void

Database

- no_of_item : long int
- Prodarr : Item[1..n]

+ search_product (bar_code:string) : Item
+ get_details (x:Item) : void
+ get_entire_inventory( ) : void
+ add_product(x:Item) : void
+ provide_statistics(bar_code:string,time_peiod : int) : void
+ del_product(x:Item) : void
+ change_prod_price(bar_code:string) : void
+ update_prod(barcode:string,quantity:float) : void
+ ispresent(barcode:string):bool

Item_transaction_history

- Date_of_sale : date
- Total_profit : float
- Price_realized : float
- Quantity: float

+ Item_transaction_history(x:date,y:float,z:float,a:float)
+ getDetails()
+ Operation 3 (arg list) : return
+ Operation 4 (arg list) : return

Printer

+ print_bill():void
+ print_statistics( ) : void

Transaction

- no_of_item : int
- array : Item[1..n]
- trans_id: string
- dd: date

+ Transaction(x:item[1..n])
+ addItem(Item x) : void
+ getprod(bar_code:string):Item
+ Provide_details():void
+ getTotal() : int
+ update_each_item_history()

1..* 0..1 1 1 1..* <<uses>> 1 0..* 1 1 1 <<uses>> 1 <<uses>> <<uses>>

Figure 1: CLASS DIAGRAM

2 SEQUENCE DIAGRAMS

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

2.1 SALES TRANSACTION:Sequence Diagram I

The following sequence diagram shows sequential message flow for transaction process . Briefly it is

for each item (Sales clerk —->BCR —->Database —>Sales clerk) —->Transaction (for each item)(——>Particular item object) ——>Printer.

The messages between the objects have been clearly shown in the diagram.
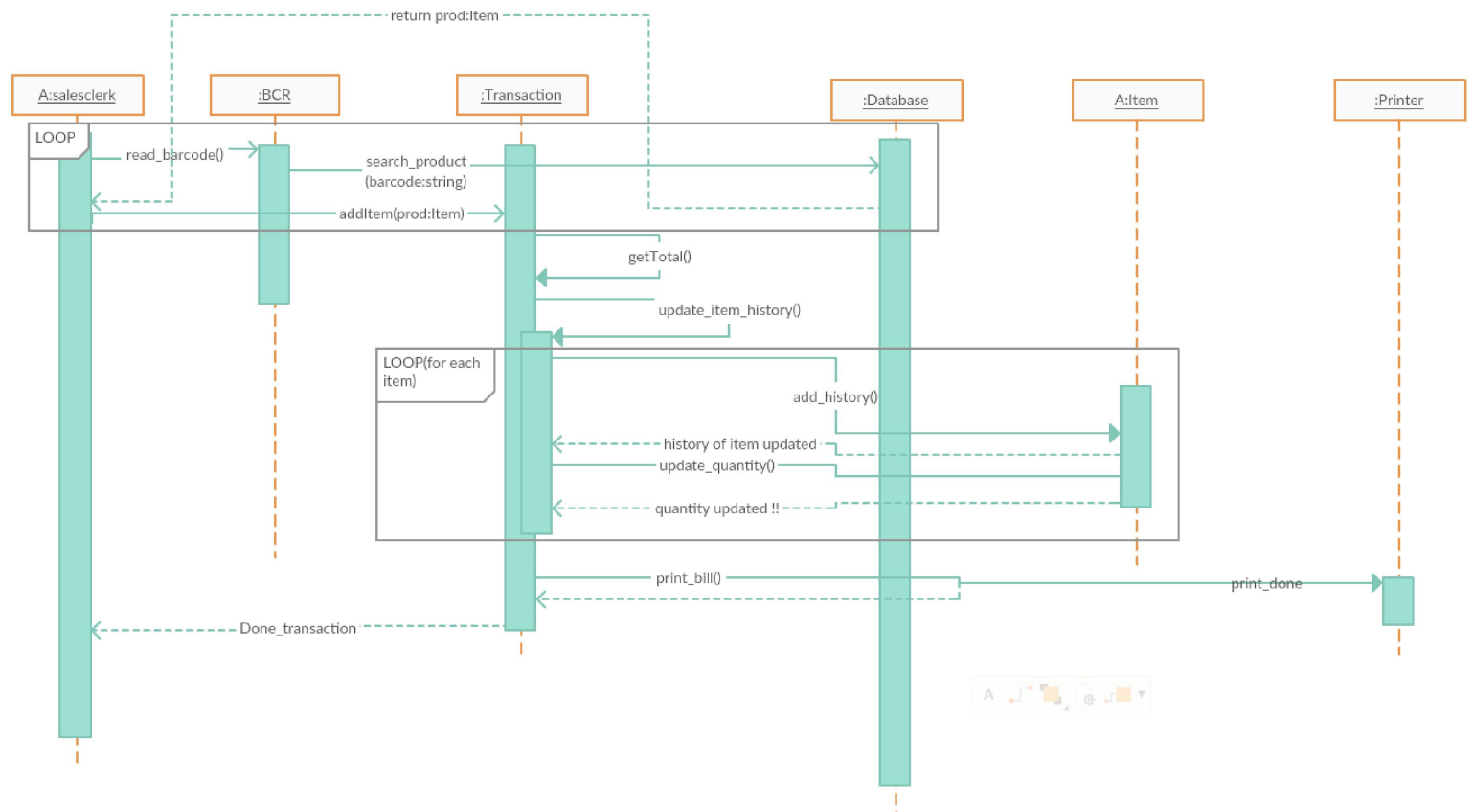
Figure 2: SALES TRANSACTION:Sequence Diagram I

2.2 VIEW SALES STATISTICS:Sequence Diagram II

The following sequence diagram shows sequential message flow for viewing sales statistics . Briefly it is

Manager—->Database—->particular Item —->Printer

The messages between the objects have been clearly shown in the diagram.

2.3 UPDATE INVENTORY : SEQUENCE DIAGRAM III

The following sequence diagram shows sequential message flow for updating inventory . Briefly it is

(for each item) (Employee —->AWM/C and BCR —->—>Database Employee—->Database)

If item is present in the database , the system simply adjusts the quantity of item else if the item isn't present , it adds a new item to the database .

The messages between the objects have been clearly shown in the diagram.
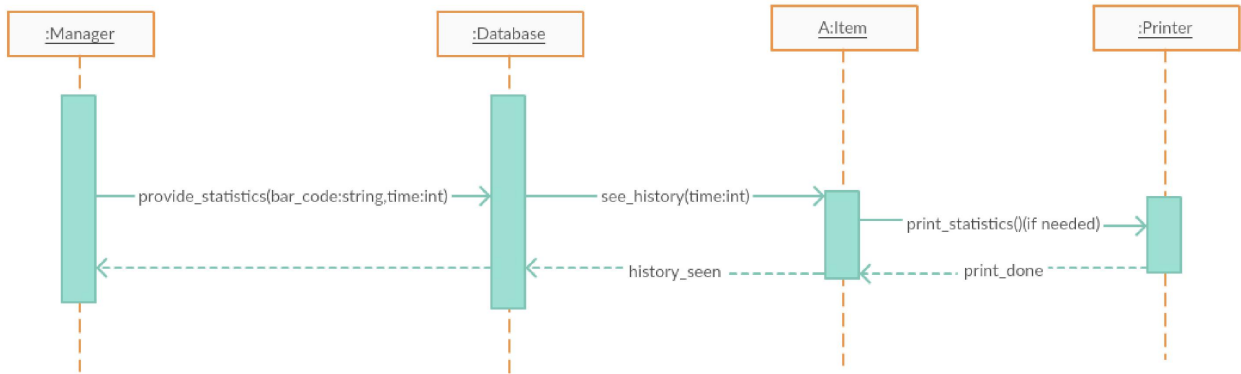
(a) **VIEW SALES STATISTICS : Sequence Diagram II**
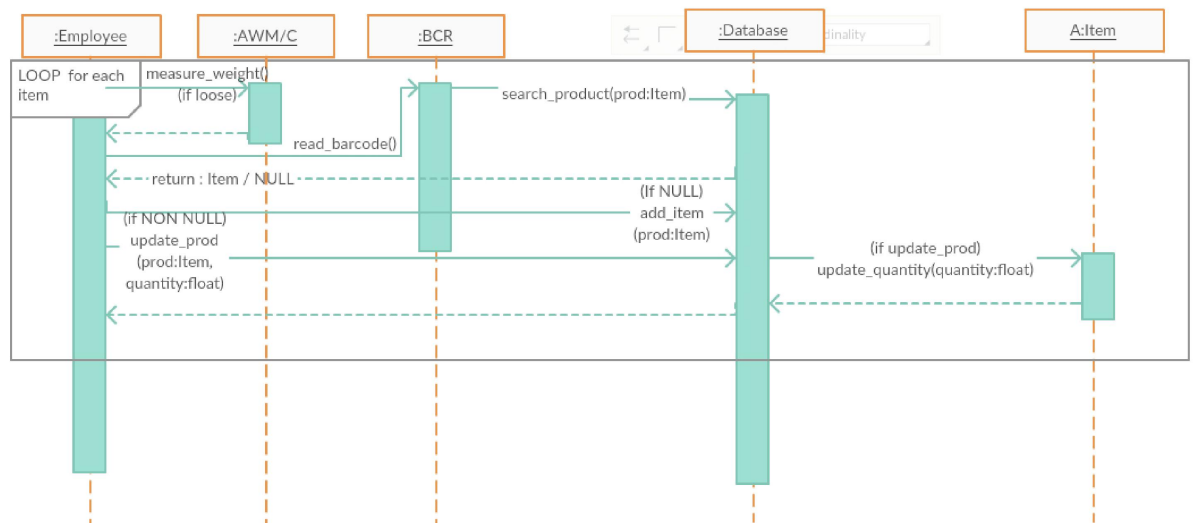
(b) **UPDATE INVENTORY : SEQUENCE DIAGRAM III**

Figure 3: **Sequence Diagrams II and III**

2.4 UPDATE PRICE : SEQUENCE DIAGRAM IV

The following sequence diagram shows sequential message flow for updating inventory . Briefly it is

(for each item) (Manager —->Database —->Printer)

If item is present in the database , the system simply adjusts the quantity of item else if the item isn't present , it adds a new item to the database .

The messages between the objects have been clearly shown in the diagram.
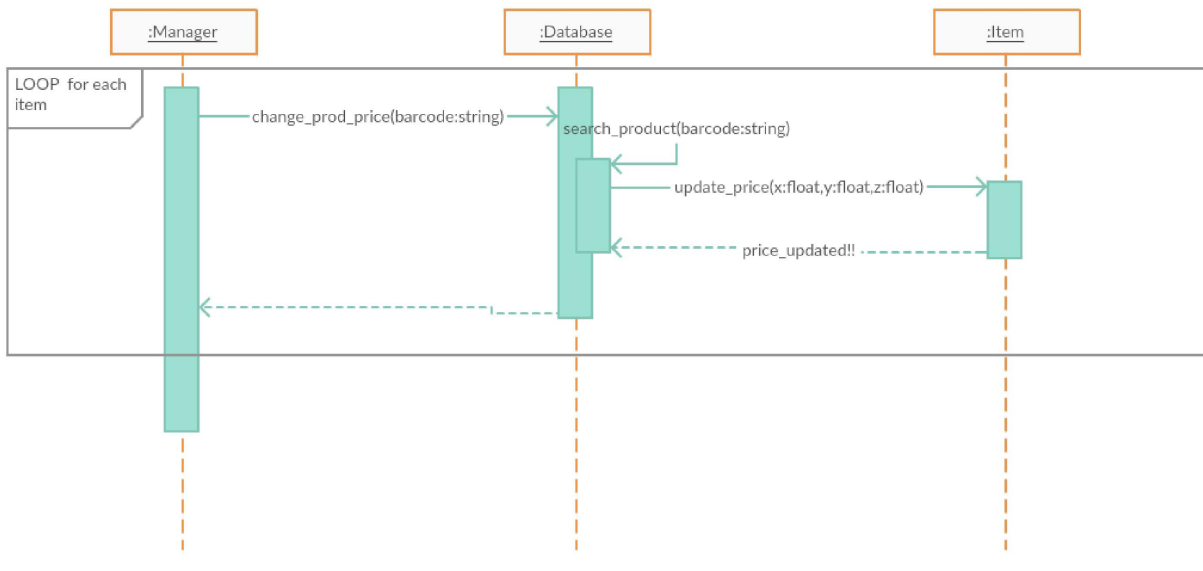
Figure 4: UPDATE PRICE : SEQUENCE DIAGRAM IV

3 ACTIVITY DIAGRAMS

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.

Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

1. rounded rectangles represent actions;

2. diamonds represent decisions;

3. bars represent the start (split) or end (join) of concurrent activities;

4. a black circle represents the start (initial state) of the workflow;

5. an encircled black circle represents the end (final state).

3.1 SALES TRANSACTION : ACTIVITY DIAGRAM I

Basic data flow includes

1. When the customer brings in order , the sales clerk places items one by one in front BCR and AWM/C(if loose) .

2. The BCR gives the bar code of the item to the system . The system searches for item details in the database and adds those details in the transaction .

3. After doing this for all items , final billing is done and inventory of each good is decreased by required amount .
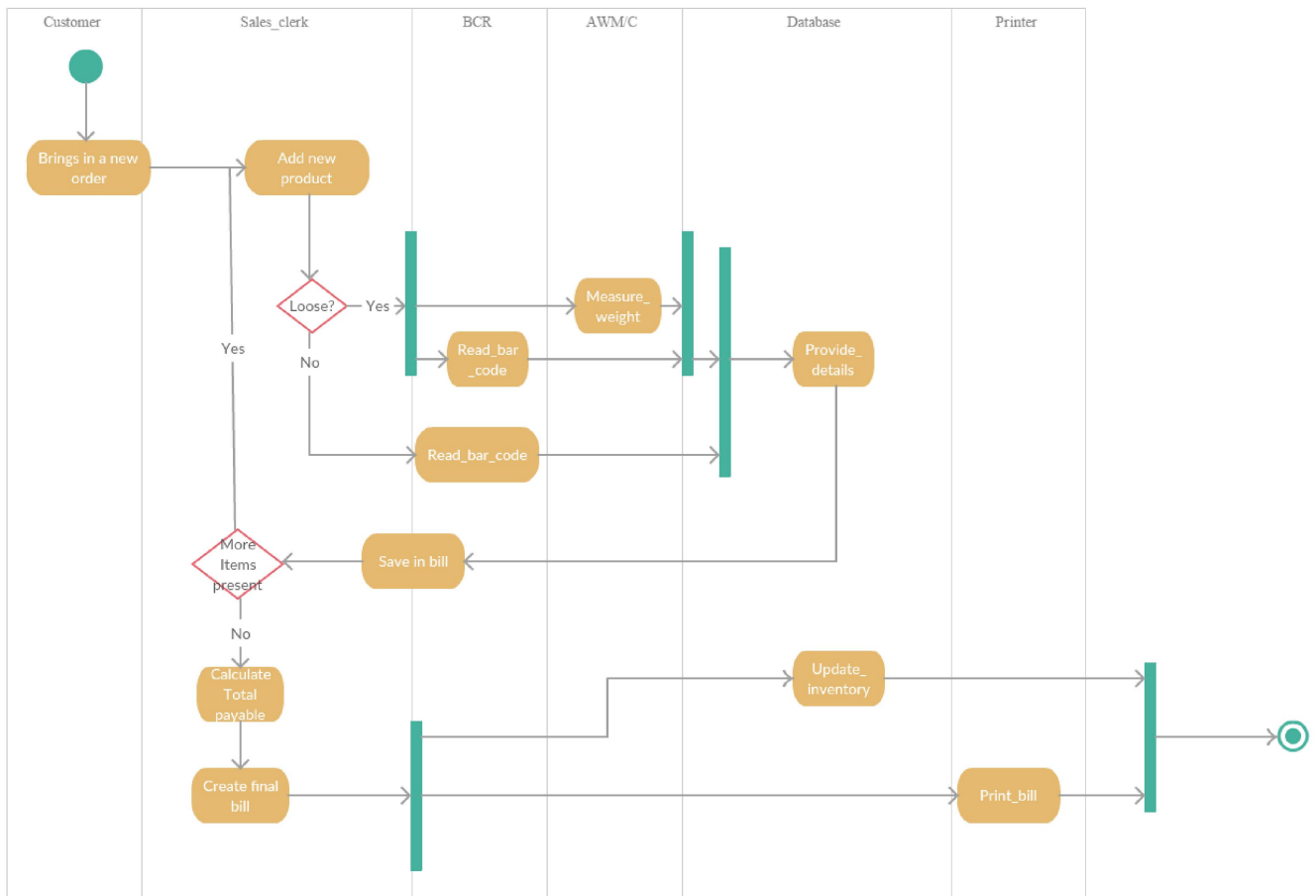
Figure 5: SALES TRANSACTION : ACITIVITY DIAGRAM I

3.2 VIEW SALES STATISTICS : ACTIVITY DIAGRAM II

Basic data flow includes

1. When the manager gives item bar code and time period to the system , the system searches the item in it's database .

2. After the item is found , the item's history is passed on to the screen .

3. Printing is optional.
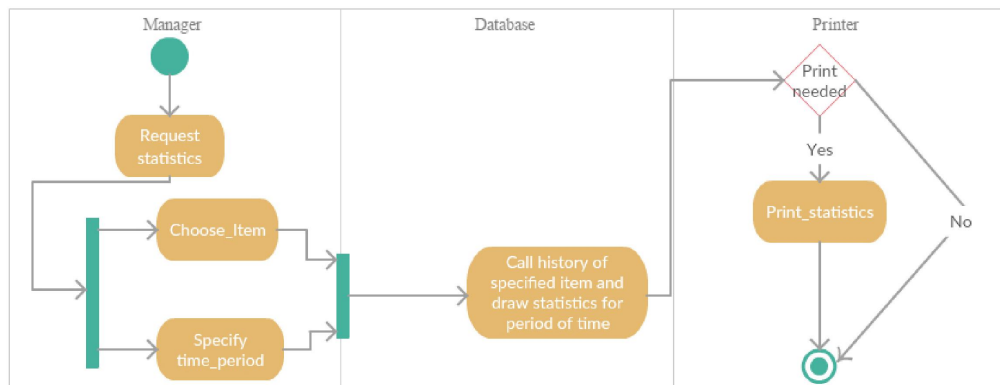
Figure 6: VIEW SALES STATISTICS : ACTIVITY DIAGRAM II

3.3 UPDATE INVENTORY : ACTIVITY DIAGRAM III

Basic data flow includes

1. On new arrival , the employee places the item in front of BCR and AWM/C(if loose) . The database searches for the item in the database.

2. If item is present in the database , the system simply adjusts the quantity of item else if the item isn't present , it adds a new item to the database .

3.4 UPDATE PRICE : ACTIVITY DIAGRAM IV

Basic data flow includes

1. The manager first specifies item barcode as input.

2. Then the system searches for the item in it's database .

3. Next , the system changes the item's price details as specified by the manager .
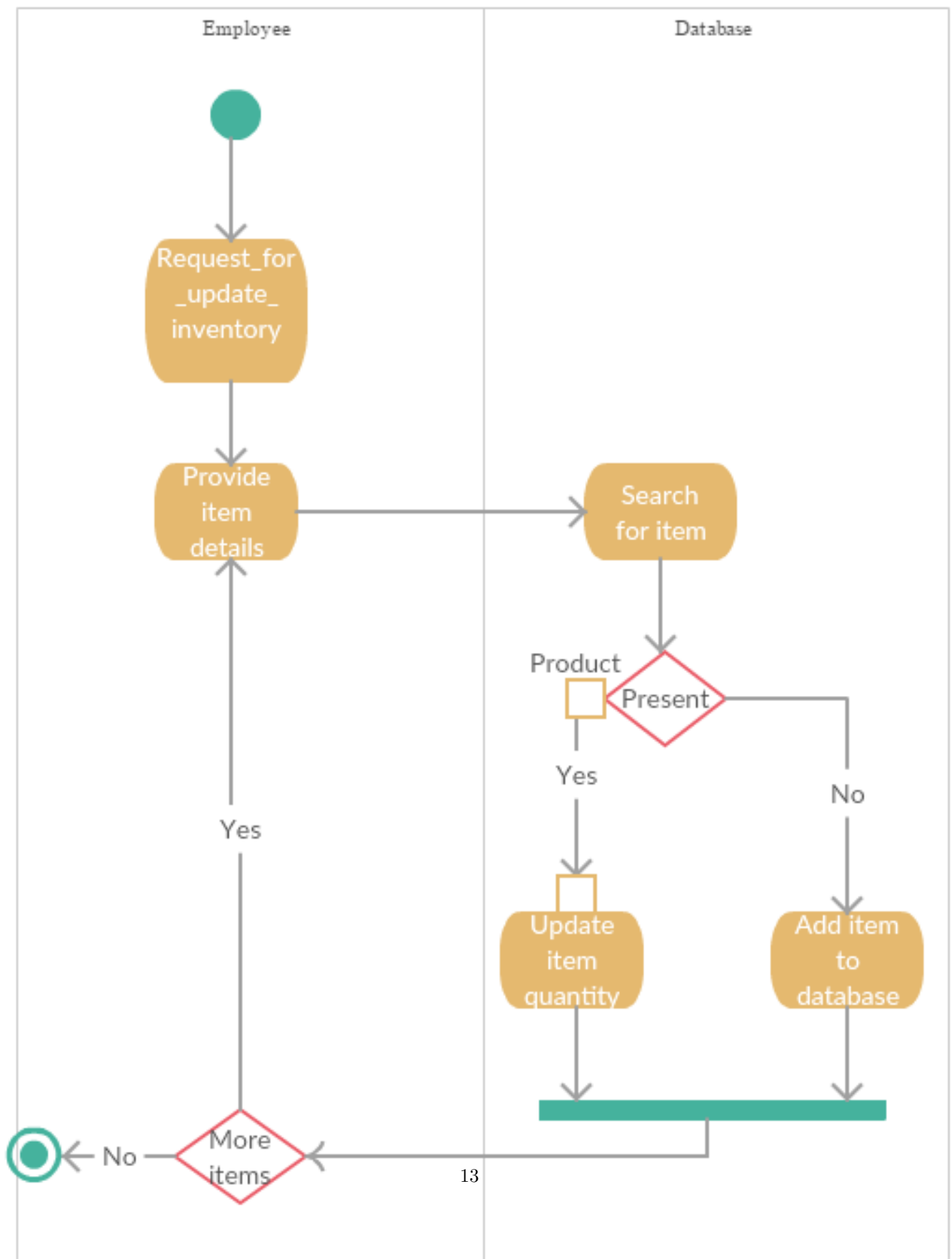
Figure 7: UPDATE INVENTORY : ACTIVITY DIAGRAM III
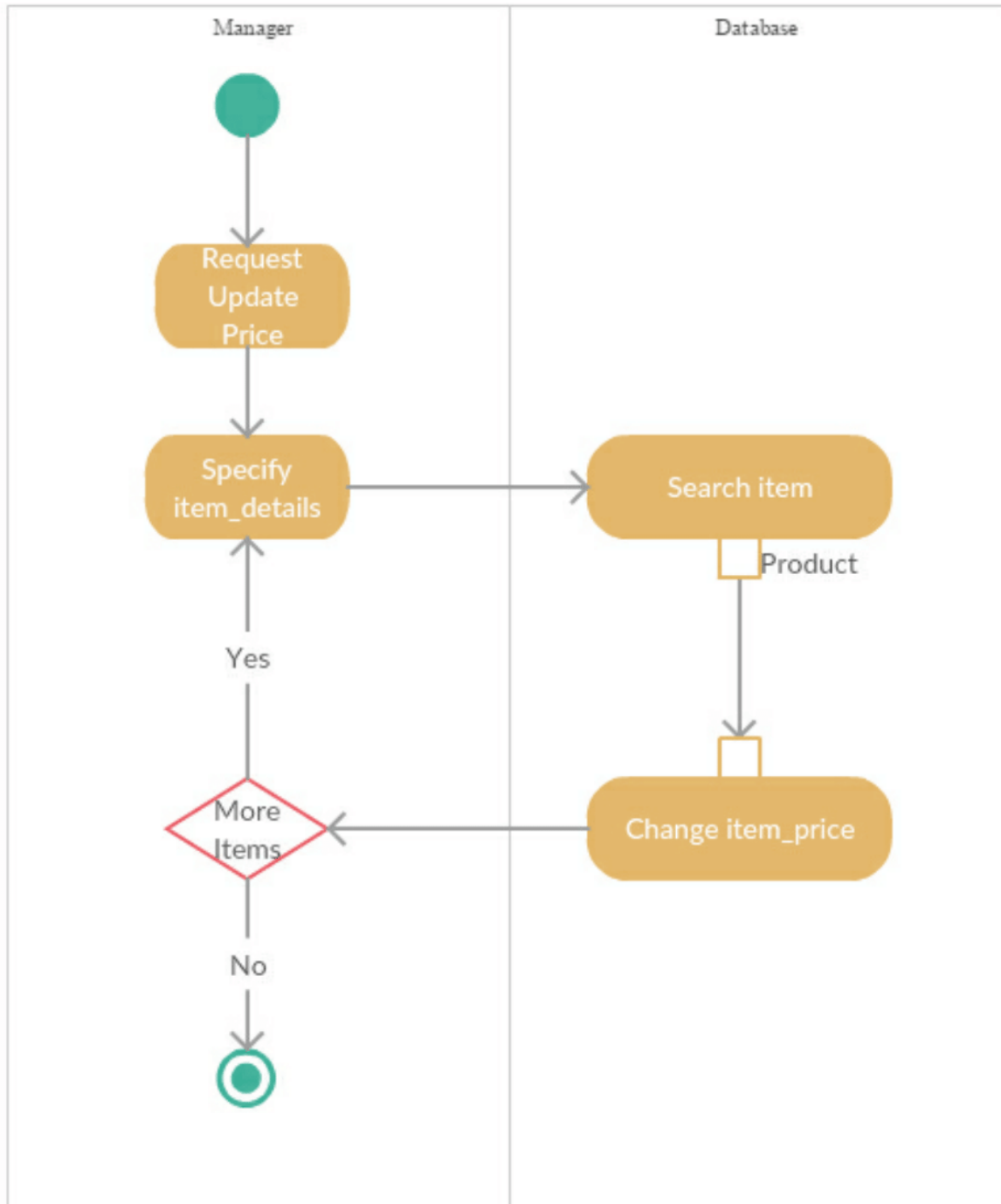
Figure 8: UPDATE PRICE : ACTIVITY DIAGRAM IV

4 COMMUNICATION DIAGRAMS

A Communication diagram models the interactions between objects or parts in terms of sequenced messages. Communication diagrams represent a combination of information taken from Class, Sequence, and Use Case Diagrams describing both the static structure and dynamic behavior of a system.

4.1 SALES TRANSACTION:Communication Diagram I

The following sequence diagram shows sequential message flow for transaction process . Briefly it is

for each item (Sales clerk —->BCR —->Database —>Sales clerk) —->Transaction (for each item)(——>Particular item object) ——>Printer.

The messages between the objects have been clearly shown in the diagram.
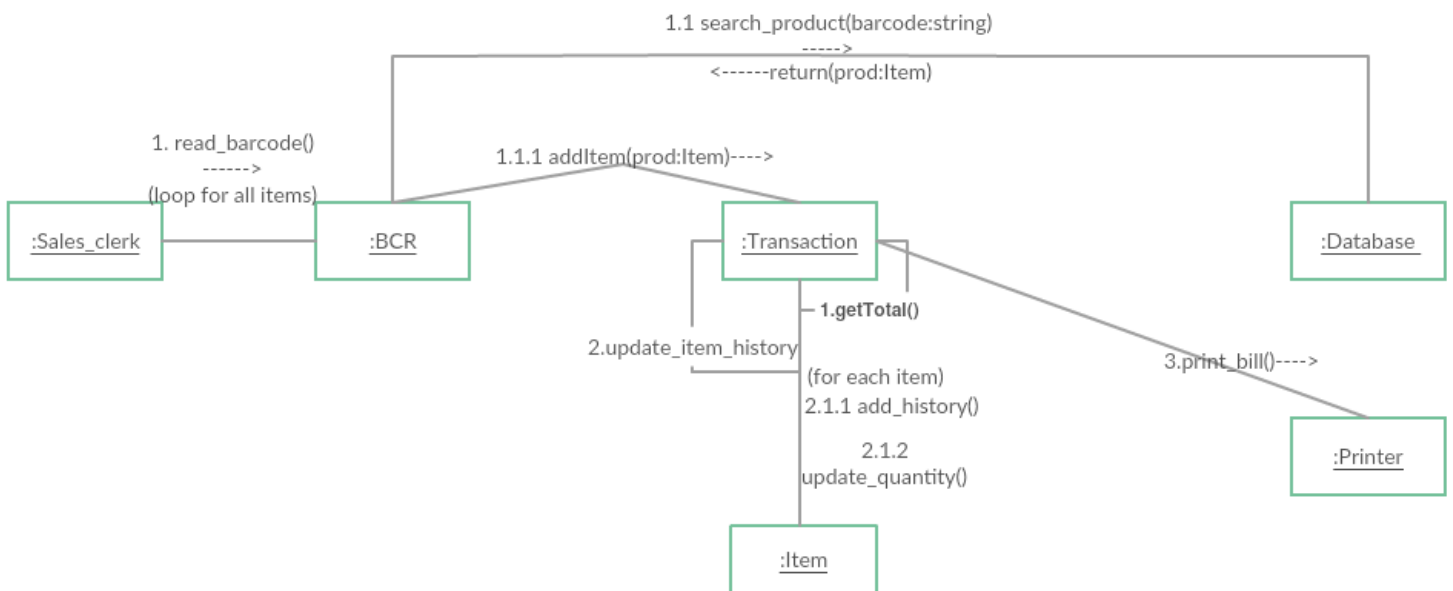
Figure 9: SALES TRANSACTION : COMMUNICATION DIAGRAM I

4.2 VIEW SALES STATISTICS:Communication Diagram II

The following sequence diagram shows sequential message flow for viewing sales statistics . Briefly it is

Manager—->Database—->particular Item —->Printer

The messages between the objects have been clearly shown in the diagram.

Figure 10: VIEW SALES STATISTICS : COMMUNICATION DIAGRAM II

4.3 UPDATE INVENTORY : Communication Diagram III

The following sequence diagram shows sequential message flow for updating inventory . Briefly it is

(for each item) (Employee —->AWM/C and BCR —->—->Database Employee—->Database)

If item is present in the database , the system simply adjusts the quantity of item else if the item isn't present , it adds a new item to the database .

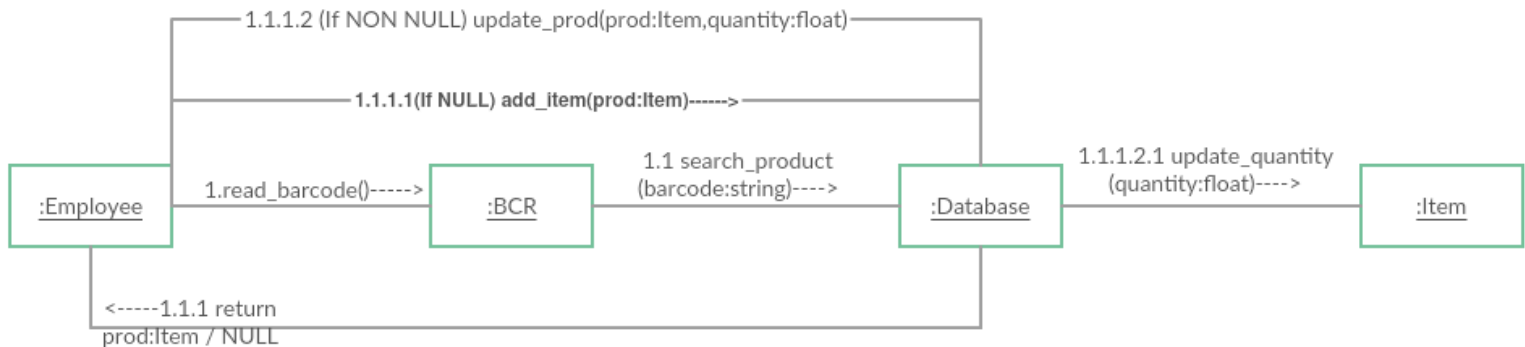The messages between the objects have been clearly shown in the diagram.

Figure 11: UPDATE INVENTORY : COMMUNICATION DIAGRAM III

4.4 UPDATE PRICE : Communication Diagram IV

The following sequence diagram shows sequential message flow for updating inventory . Briefly it is

(for each item) (Manager —->Database —->Printer)

If item is present in the database , the system simply adjusts the quantity of item else if the item isn't present , it adds a new item to the database .

The messages between the objects have been clearly shown in the diagram.
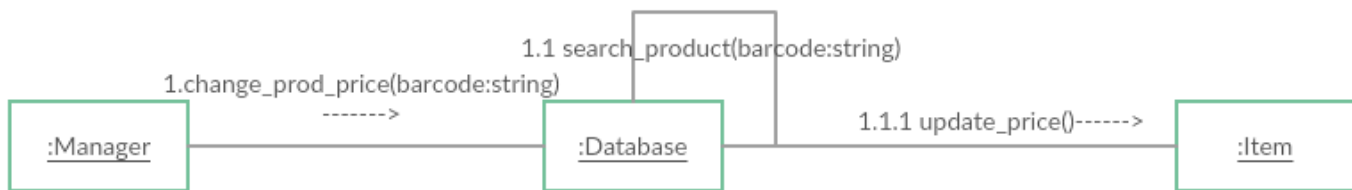
Figure 12: UPDATE PRICE : COMMUNICATION DIAGRAM IV

5 STATECHART DIAGRAMS

State diagrams are used to give an abstract description of the behavior of a system. This behavior is analyzed and represented as a series of events that can occur in one or more possible states. Hereby each diagram usually represents objects of a single class and track the different states of its objects through the system .

5.1 ACCOUNT : STATECHART DIAGRAM I

The various states are

1. Idle

2. Activated

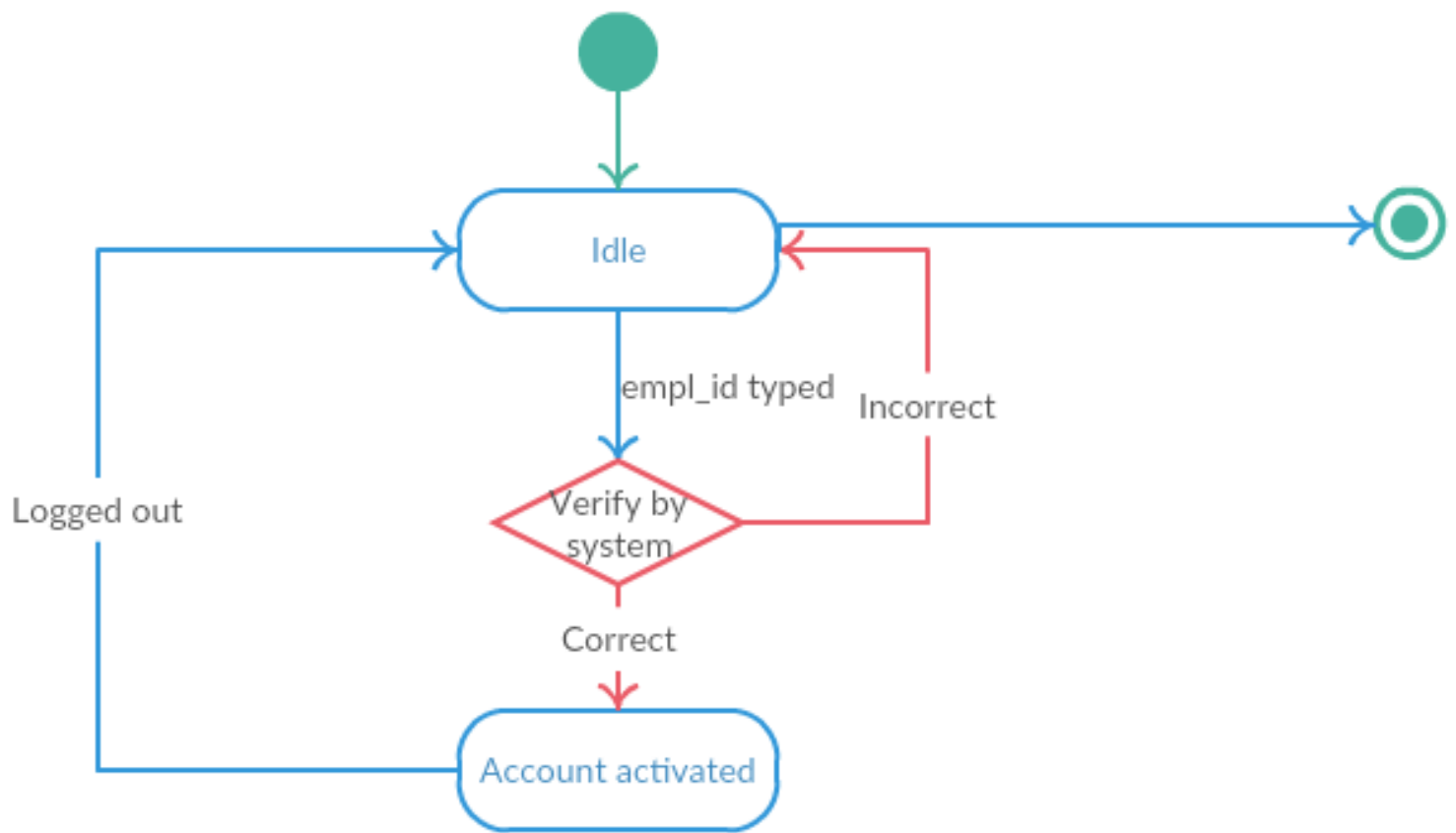The various state transitions are shown in the diagram .

Figure 13: ACCOUNT OF AN EMPLOYEE : STATE CHART DIAGRAM I

5.2   BARCODE READER : STATECHART DIAGRAM II

The various states are

1. Idle

2. Bar code identified

3. barcode processed

4. barcode returning to database

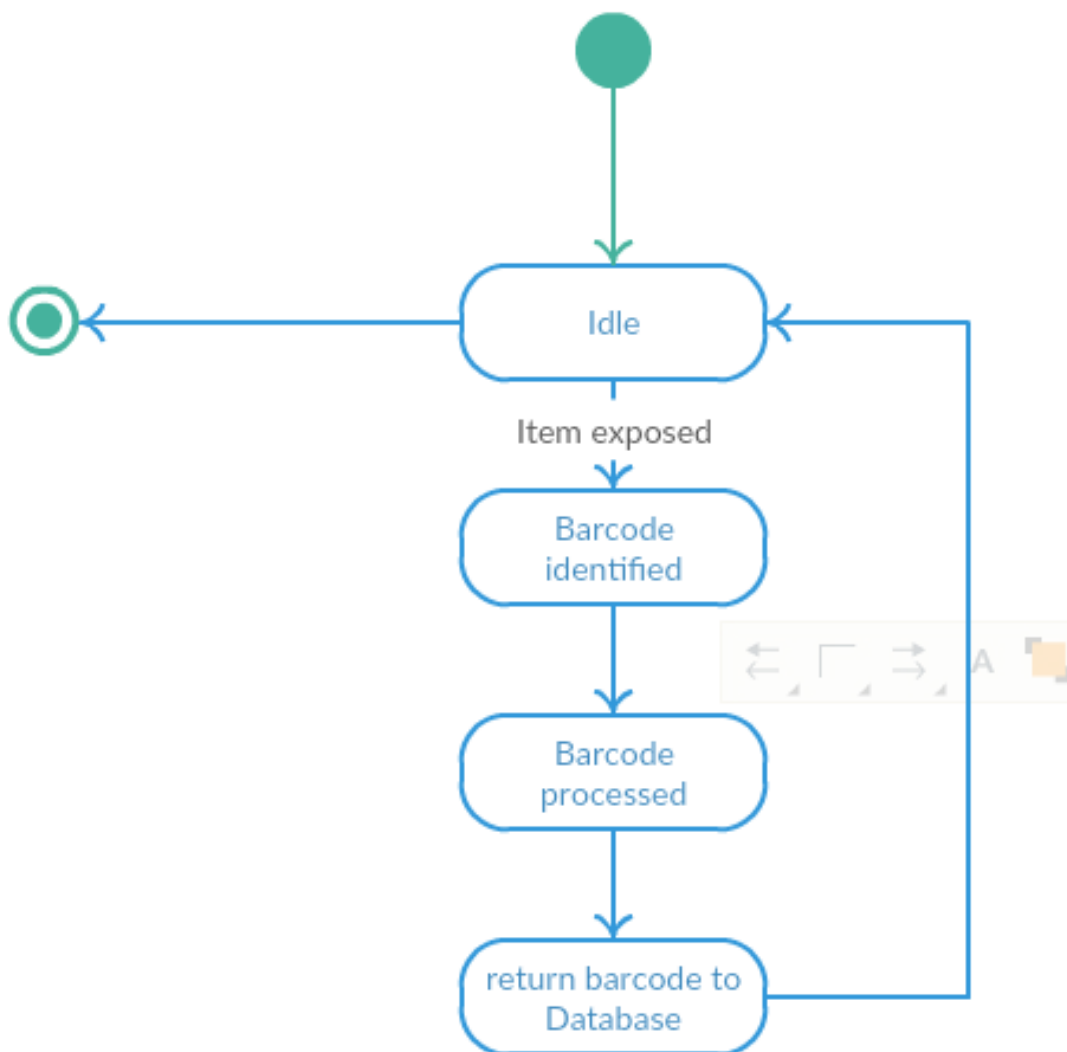The various state transitions are shown in the diagram .

Figure 14: BARCODE READER(BCR) : STATE CHART DIAGRAM II

5.3 AUTOMATIC WEIGHING MACHINE : STATECHART DIAGRAM III

The various states are

1. Idle

2. Weight measured

3. return weight

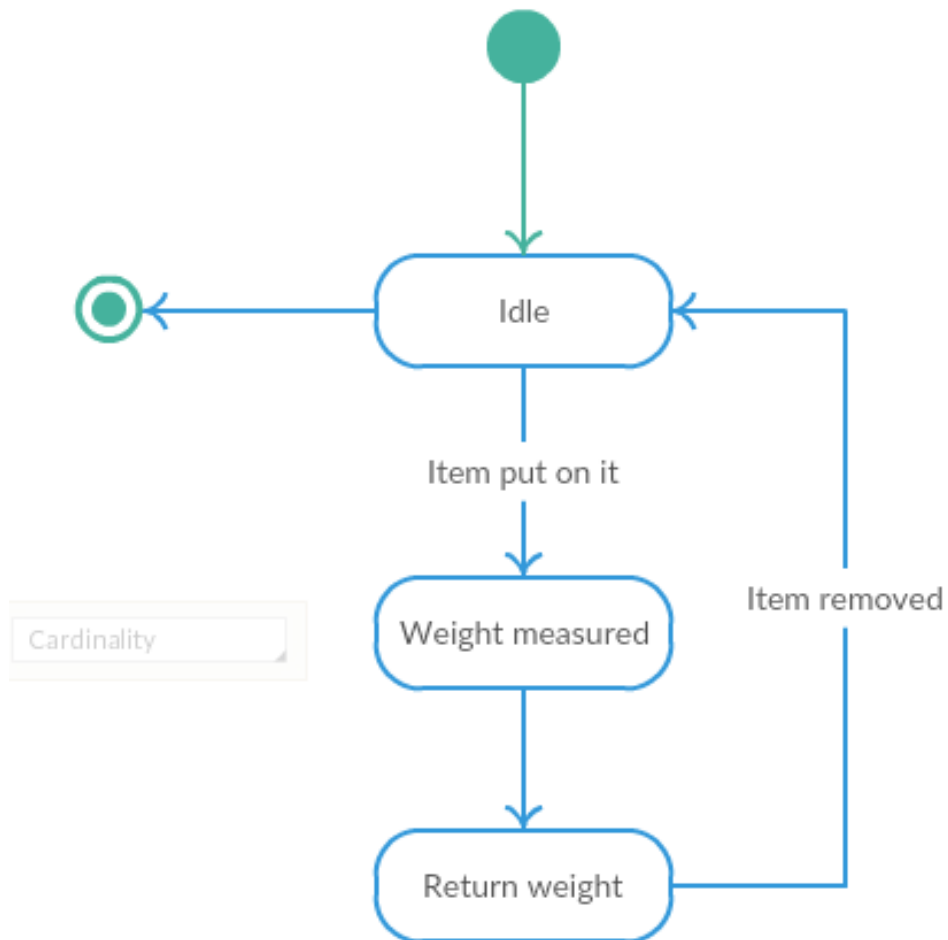The various state transitions are shown in the diagram .

Figure 15: AUTOMATIC WEIGHING MACHINE(AWM/C) : STATE CHART DIAGRAM III

5.4   PRINTER : STATECHART DIAGRAM IV

The various states are

1. Idle

2. Process document for print

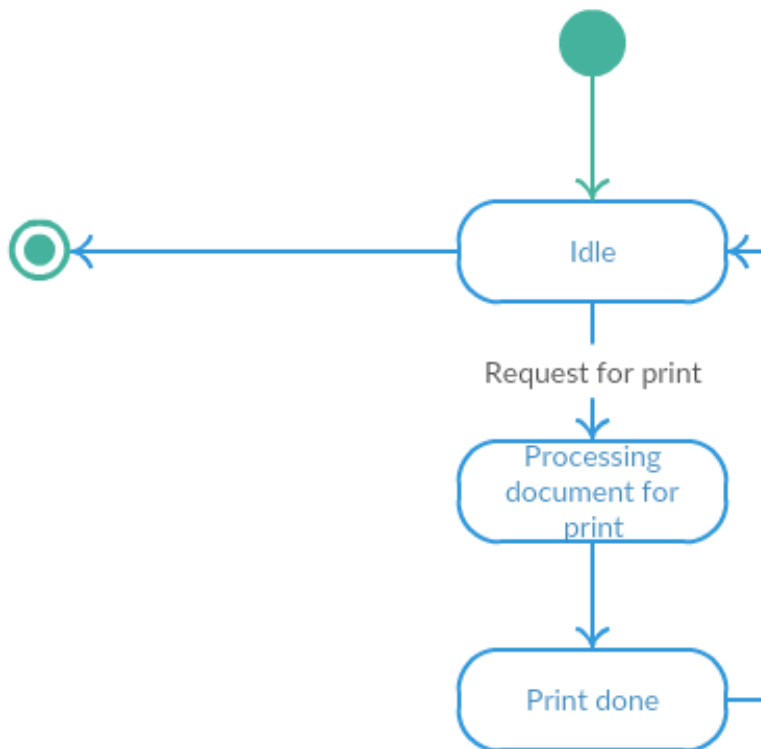3. print done

The various state transitions are shown in the diagram .

Figure 16: PRINTER : STATE CHART DIAGRAM IV