****

**POINT OF SALE: CAFÉ**

**Prepared For :**

**REZWAN AHMED**

**Project submitted by-**

**Name ID**

|  |  |
| --- | --- |
| **Ahmad, Mutasim Billah Bin** | **16-31533-1** |
| **Ahmed, Md.Fahim** | **16-30987-1** |
| **Noshin, Fariha Tabassum** | **16-31832-1** |

Table Of Content :

**1 . System Summery ………………………………………………………………....………...….. . 3**

**2.** Basic Methodology ………………………………………………………………..…..… 3

3. **An ERD diagram …………………………………………………………..……..…….…. 4**

**4. Class Diagram ………………………………..…………………………………………..... 5**

**5. Activity Diagram ………………...…………………………………………….………..… 6**

**6.** Initial Activities ………………………………………………………………………….7

7. Use case diagram ………………………………………………………...........……….. 8

8. Database Tables …………………………………………………………………..……..9

**9. Table Snapshots ……………………………………………………………………………11**

**10. Tasks ……………………………………………………………………………………....14**

**11.Views ……………………………………………………………………………………….15**

**Point of Sale**: Café

# System Summery

About the problem:

In a large business, a good [point of sale system](https://www.lightspeedhq.com/pos-system/) is one of the biggest assets. If it works as advertised, it’ll be a lot like having my own team of experts working behind the scenes, making sure everything’s moving along quickly and efficiently. To remain on top in today’s highly competitive landscape, we need a POS system to help us [run our business the right way](https://www.lightspeedhq.com/pos/retail/manage-with-retail-pos-system/). Here’s why,

1. It will save our valuable time.
2. We will have a full view report of our business.
3. Costs, profits and loses will be clearly indicated.
4. We can manage the data of hired employees.
5. Obviously, it just looks cool!

Basic Methodology:

A POS system for a café is supposed to be very simple comparing to the one used in a retail business. But we have added some sophisticated features as well to provide a good experience to our users.

At the core of the system, there will be an inventory system. It will have the records of purchases from the vendors and the orders placed by the customers. Once an order is placed, the items required to make those order items will be reduced from the inventory system. Again, the items bought from the vendors will appear in the inventory system.

The most active part of this system is “Order” section. A staff usually spends his/her most of the time in this section. Customers make orders and staffs take them putting their details on the system. After the placement of an order, the staff can print the receipt of that order. The inventory gets updated accordingly when the order reaches into “Served” state.

Now you can guess, there is also a “Purchase” section in this system which is similar to the “Order”. In a purchase, a vendor provides some items and receives a receipt from the system. Staffs put the details of purchases into the system. Whenever a purchase is done, the inventory gets updated with new items.

The details of the orders and purchases are stored in the database. The owner or any other staff with the admin access can view the transactions from the system. There will be different useful views from the database in the admin section as well such as, which vendors are best suited for which items, which staff was taking the order while a customer made a report, if an order took so long to be served or not, which items are now popular among the customers and so on!

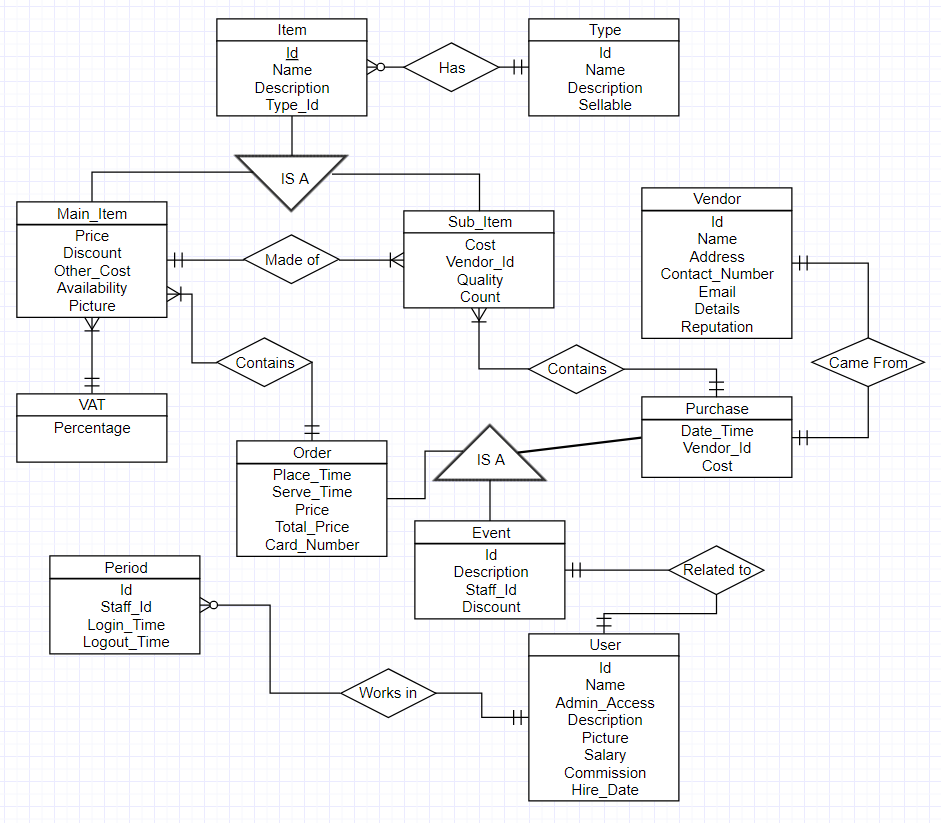
****

Fig: ER - Diagram

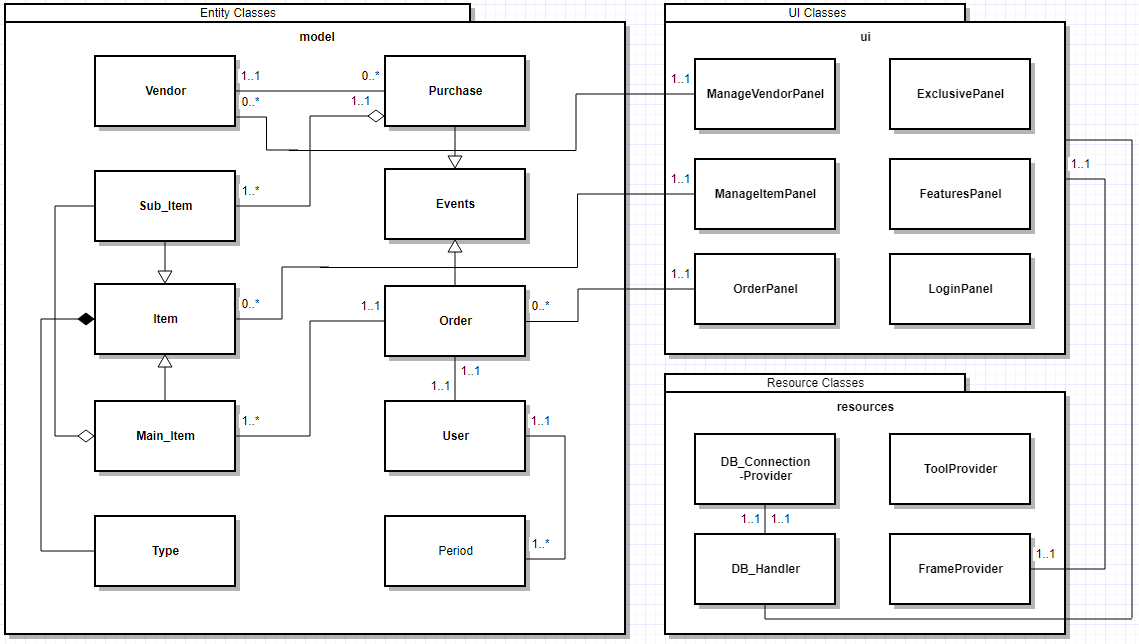
****

Fig: Class Diagram

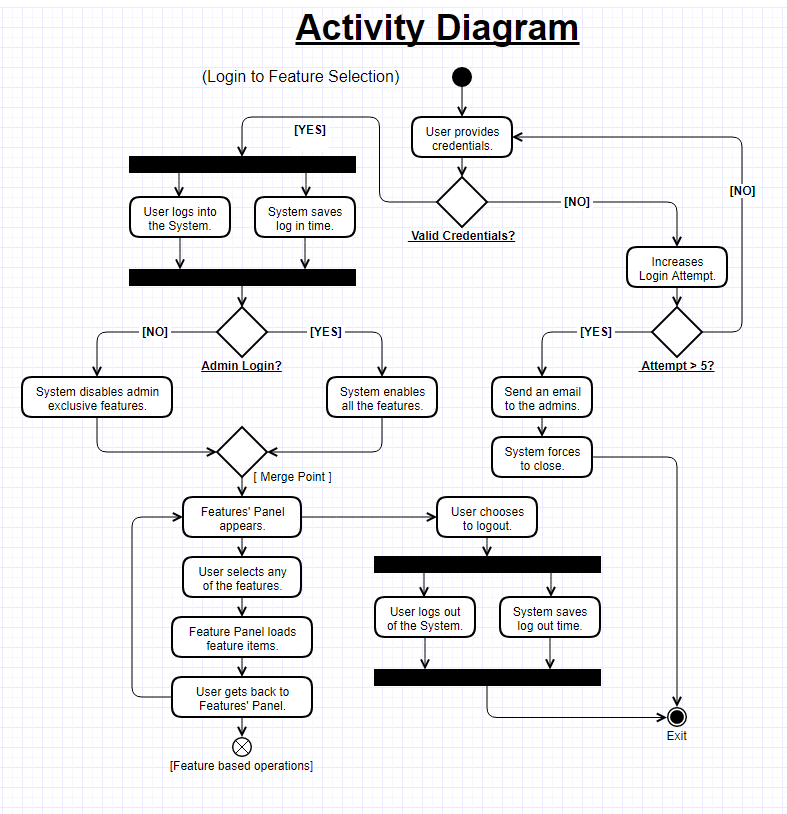
****

Fig: Initial Activities

This activity diagram describes the flow from the beginning to the feature selection panel. To reach there, a staff should log into the system with valid credentials. Then according the staff category, this system will load its features. If a staff fails to provide his / her valid credentials, the application will be closed by force and then, an email will be sent to the users having admin access. The time duration of the system being used by staff will also be saved in the database.

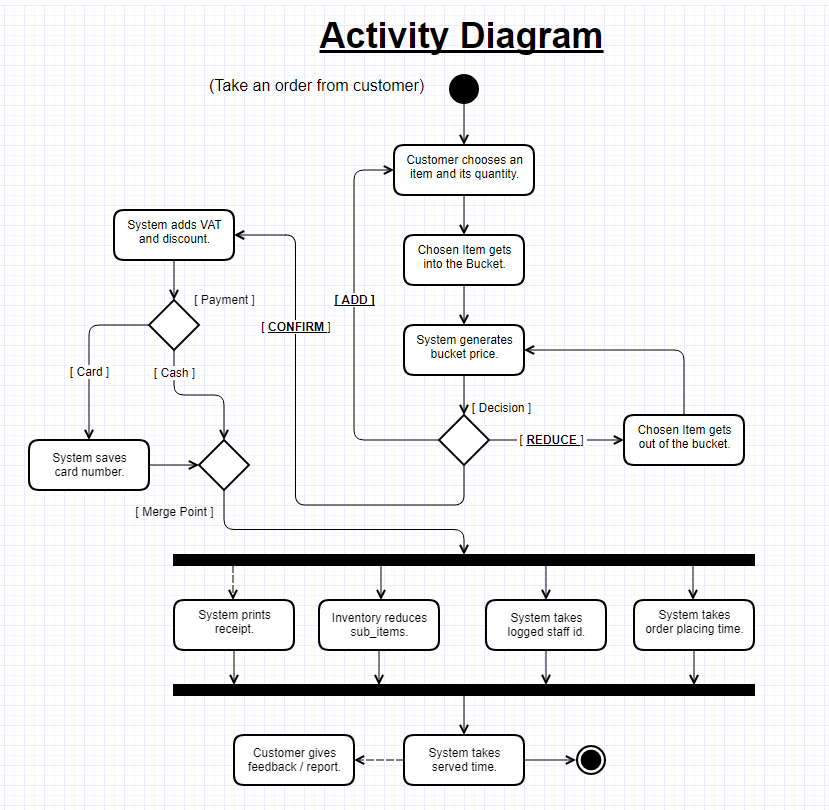
****

Fig: Initial Activities

This diagram represents the flow of an ordering process. Customers can order a fully customizable bucket of items. After the his / her confirmation, he / she will need to pay for the bucket. Then he / she will be provided the receipt and the database will update the inventory accordingly. With these, system will also update the order placing time and the staff id into the database. A feedback or report from the customer will be an optional feature in the ordering process.

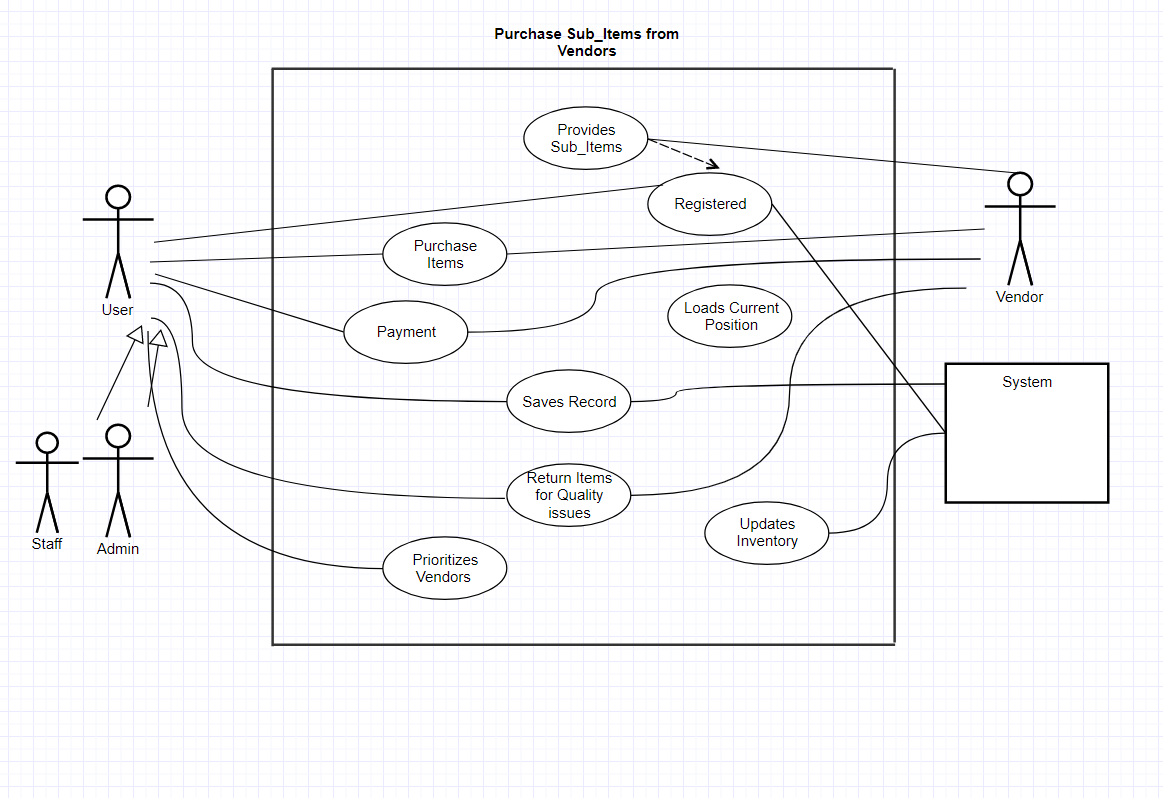
****

Fig: Use case diagram of purchase from vendor

Database Tables:

***Main\_Item***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Name | Price | Description | T\_Id | Discount\_Rate | Other\_Charges | Availability | Count | Picture |

***VAT***

|  |
| --- |
| Vat\_Percentage |

***Sub\_Item***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Name | Description | Max\_Cost | Count |

***Main\_Sub***

|  |  |  |
| --- | --- | --- |
| M\_Id | S \_Id | S\_Count |

***Type***

|  |  |  |
| --- | --- | --- |
| Id | Name | Description |

***Vendor***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Name | Address | Contact\_Number | Email | Details | Reputation |

***Purchase***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | DESCRIPTION | TIME | U\_ID | V\_ID | TOTAL\_COST | DISCOUNT\_RATE |

***Purchase\_Item***

|  |  |  |  |
| --- | --- | --- | --- |
| P\_Id | S\_Id | Price | Quantity |

***Order***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Description | Place\_Time | Serve\_Time | U\_Id | Price | Discount | Card\_Number |

***Order\_Item***

|  |  |  |
| --- | --- | --- |
| O\_Id | M \_Id | Quantity |

***User***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Name | Password | Admin\_Access | Description | Picture | Salary | Commission | Hire\_Date |

***Period***

|  |  |  |  |
| --- | --- | --- | --- |
| Id | U\_Id | Login\_Time | Logout\_Time |

***Change\_Log***

|  |  |  |
| --- | --- | --- |
| Id | U\_Id | DESCRIPTION |

# Table Snapshots:

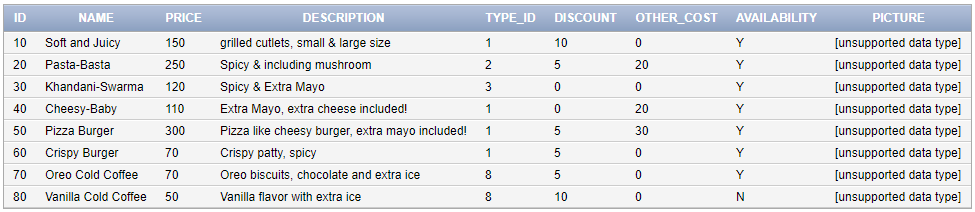


Fig: Main\_Item

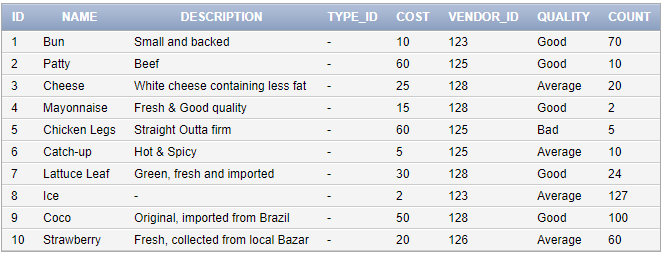


Fig: Sub\_Item



Fig: Vendor



Fig: VAT

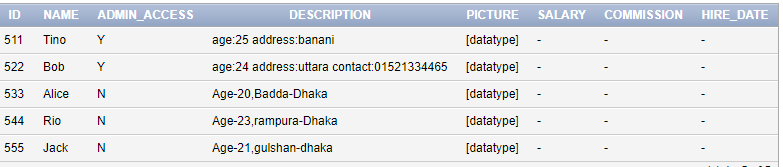


Fig: User

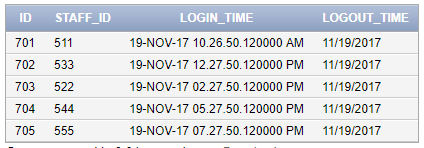


Fig: Period

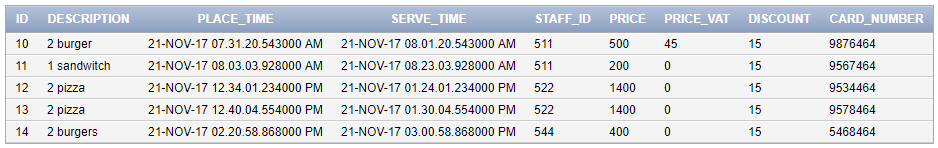


Fig: Order

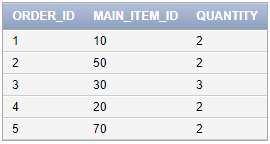


Fig: order\_Item

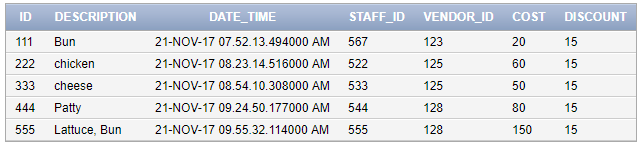


Fig: purchase

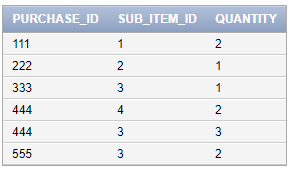


Fig: purchase\_Item

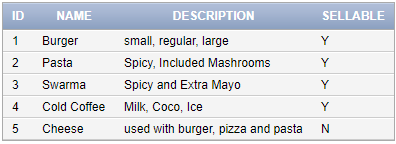


Fig: type

# **Queries:**

1. Show the types of product that are orderable.

* SELECT \* FROM type WHERE sellable = ‘Y’;

1. Display the staffs having admin access to the features.

* SELECT \* FROM users WHERE admin\_access = ‘Y’

1. Find out the staff who have taken maximum number of orders.

* select u.id from orders o, users u where o.staff\_id = u.id group by u.id having count(\*) = (select max(count(\*)) from orders o, users u where o.staff\_id = u.id group by u.id);

1. Find out the items which have been sold for maximum amount of time.

* select main\_item\_id from order\_item group by main\_item\_id having sum(quantity) = (select max(sum(quantity)) from order\_item group by main\_item\_id);

1. Find out the staff who was working during a report had been submitted.

* select staff\_id from period where (timestamp '2017-11-19 13:05:30') BETWEEN login\_time and logout\_time;

1. Find out the items which have been purchased mostly.

* select sub\_item\_id from purchase\_item group by sub\_item\_id having sum(quantity) = (select max(sum(quantity)) from purchase\_item group by sub\_item\_id);

1. Find out the staff who have been with most of the purchases.

* select u.id from purchase p, users u where p.staff\_id = u.id group by u.id having count(\*) = (select max(count(\*)) from purchase p, users u where p.staff\_id = u.id group by u.id);

1. Which of the items do use maximum ingridients?

* select main\_item\_id from main\_sub group by main\_item\_id having sum(quantity) = (select max(sum(quantity)) from from main\_sub group by main\_item\_id);

1. Find out how much burgers were sold for last 30 days.

* select sum(quantity) from order\_item, orders where orders.id = order\_item.order\_id and place\_time between current\_timestamp and (current\_timestamp - interval '30' day) and main\_item\_id = (select id from main\_item where name = 'burger');

1. Find out the most junior staff.

* select \* from users where hire\_date = (select max(hire\_date) from users);

# **Views:**

1. View sub\_vu will display name,description and cost from Sub\_Item table

* create view sub\_vu as select name,description,cost from Sub\_Item

1. View p\_vu will display id,date\_time,cost from purchase table levelling them p\_id,p\_date\_time,p\_cost

* create view P\_vu (P\_id,P\_date\_time,p\_cost) as select id,date\_time,cost from purchase

1. View price150 will display id,name,price from Main\_Item table for all the items containing price 150

* create view price150 (Item\_Name,Item\_Id,Item\_Price) as select id,name,price from Main\_Item where price=150 with check option constraint ck

1. View SubI\_VU WILL DISPLAY the mentioned column from both table Sub\_Item and purchase\_item

* view SubI\_VU WILL DISPLAY the mentioned column from both table Sub\_Item and purchase\_item

1. View SI\_VU will display the mentioned column from both table Sub\_Item,Main\_Sub order by sub\_count

* create view SI\_VU(SubNmae,SubId,SubDescription,SubCost) as select s.name,s.id,s.description,s.cost from Sub\_Item s,Main\_Sub m where s.id=m.id order by m.sub\_count

1. View I\_vu will display the minimum and maximum price of Item sub\_count wise.

* create view I\_VU(Sub\_Count,MinPrice,MaxPrice) as select s.sub\_count,MIN(m.price),MAX(m.price) from Main\_Item,Main\_Sub s m where m.id=s.id group by s.sub\_count