

Project Report



Project Title: Fast Food Restaurant Management System

Client Name: Dr. Taimoor Khan

Submitted by: The Conquerors

Group Members:

P156046 Ammar Khan

P15-6079 Muhammad Mudassir

P166040 Noormah Omar Khan

P166055 Sabahat Ijaz

P166058 Hamza Bukhari

P166059 Yousuf Khan

P166068 Muhammad Hamza

P166069 Muhammad Toqeer

P166117 Laiba Gul



14TH DECEMBER, 2018
FAST NUCES PESHAWAR
Phase 1, Industrial estate, Hayatabad

1. SUMMARY

The project is basically a fast food management system. We are making a system for the restaurant which will record sales, store information of customers and bought and used items. We can also store information related to the deals we want to provide to the customers. At the end, the system will predict the top customers, used and unused items and will display all the information in the form of a graph which will provide a more convenient way of analysis by the owner.

A fast food management system is essential for the restaurant. With this software, our sales and expenses will be managed automatically. Daily profit, loss and average sales will be shown in the form of graphs. Top customer information, food deals and customers' orders will be managed by the software. It will replace the old management system of hard form where registers are used to store information and the time consuming business analysis are done manually. Hence the software will be a revolutionary experience since the information stored by the user in the software will automatically do the analysis in a much shorter period of time.

We are well aware of whatever is used in the project like UML diagrams. From implementation point of view, we have much idea related to classes. We are able to manage database and acquire enough skills to implement it.

Project Details:

Location: FAST NUCES Peshawar

Proposed Starting Date: 30th October, 2018

Project Duration: 2 months

Total Budget: Rs. 10,000

2. BACKGROUND AND JUSTIFICATION

This project is about Fast Food Restaurant. It includes all details about their sales, purchases, customers and profit plus the salaries of all employees.

3.Requirements

a. Functional Requirements:

- Admin login
- Operator login
- Remove items, promotions, deals, operators, customer (by admin)
- Add items ,promotions, deals, operators (by admin)
- Add customer and sales (by operator)
- Update items, promotions, deals, operators (by admin)
- Search for item, sale, and customer.
- Make bill in PDF format.
- Make graph based on information in Databases.
- Database having Admin, customer, sales and operator information
- Regular customer details(some sort of ID number)
- Multiple search type either ID, Date, Name
- Create new employee(waiter, operator)
- Discount log maintenance
- Create user (admin)
- Printer linking and usage
- Graph features i.e. top products, top customers, daily weekly and monthly sales record
- Display of statistics every time.

b. Non- Functional Requirements:

- Password Type
- Online Order
- Email based or phone number based logins
- Deals type(daily, weekly, monthly)
- Determine user type(admin/operator)
- Graph type i.e. pie, bar etc.
- Tax bar

4.Team

Following are the team members.

Sabahat Ijaz and Muhammad Touqeer:

Their coding is excellent. They made different projects in C++ and assembly language respectively. At the same time they have leadership skills as well.

Noormah Omar Khan and Laiba Gul:

They have good documentation skills. They write many proposals for events, reports etc.

Hamza Bukhari and Yousaf Khan:

They helped Sabahat and Touqeer with their work. They have good marketing skills.

Mudassir Athar Hamza Jamil and Ammar Khan:

They have good communication skills.

5. OBJECTIVES

As we know today world is of technology. Computer is basic need of every field. This project will facilitate every restaurant organization. As it will help store information and calculation about their profits, sales and employees etc.

6. Budget

Salary per Employee: Rs. 1000/-

Stationary: Rs. 500/-

Printing: Rs. 300/-

7. Deliverables after every two weeks (fortnight)

1st Deliverable: Inception Phase

GOALS:

To build a software for FAST FOOD RESTURANT which would help them in analyzing their sales and control sales and customers and to analyze the best sales and the best customer of the month.

Top 10 requirements:

- 1- Admin login
- 2- Operator login
- 3- Remove items, promotions, deals, operators, customer (by admin)
- 4- Add items ,promotions, deals, operators (by admin)
- 5- Add customer and sales (by operator)
- 6- Update items, promotions, deals, operators (by admin)
- 7- Search for item, sale, and customer.
- 8- Make bill in PDF format.
- 9- Make graph based on information in Database
- 10- Database having Admin, customer, sales and operator information

Analysis of Project:

The software will maintain the database for this purpose. Database will have admin ID, Name, Address, Gender, Email, Password. It will record details of the customer like Name, Address, Phone number, Gender, CNIC and points. It will record the details of the operator like Name, ID, Address, Phone number, Gender, CNIC, password. It will record sales based on information like customerName, salesId, ItemId, ItemName, price, quantity, totalBill, Date, Time. The two types of user are operator and admin. We need to assign ID and Password to each operator. So that we can allow them to login to add customer and sales. Operator will enter the information of customer and sales and when it will save the information it will be saved in Database. Database will maintain the tables for customer, sales and operator. And admin will also have the ID and password after verification through database he will be allowed to access and edit the database. He can add/remove and alter the details of operator, deals, items and promotions. He can change the price of an item. When he will hire an operator he will add the details related to him. When he will fire the operator he can remove the record. He can see the sales of the day as well.

Bill will be prepared in PDF format. We are going to need PDF software and the details of the bill will be shared to two printers. One at front door and the other in Kitchen. PC having software will be

connected to both the printers and the bill will have details like customersName, salesId, ItemId, ItemName, price, quantity, totalBill, Date, Time and GST (tax). Based on the information being stored, the software will generate graphs as daily sale in month to know which day has greatest sale by using totalSalesPerday and Day. Generate graph for daily sale analysis as the mostly sold item in a day by using items and quantity. Graphs will be generated easily if we maintain tables having information being used in graphs. By using that tables we can analyze which item is sold most and we can have top 3 items. Using sales table we can check top 3 customers. Moreover we will keep track of feedback and adaptation after every iteration

HR budget:

Workers:

Two Database controller (to create tables for each and every requirement with primary keys)

Two Interface developer (to build the interface in order to interact with database).

Working hours:

8am – 8pm

Days required:

60 days

Per iteration time:

2 weeks

Person hours:

2 persons per hour.

We are doing this project because it is feasible and in business domain. We will build this project In sha Allah.

2nd Deliverable: Elaboration Phase Iteration 1(Risks were Highlighted and Some use cases are fully implemented)

Actors of the system:

- 1- Operator
- 2- Admin
- 3- Customer (Secondary user to some extent if we extent it to online ordering)

More Requirements:

- 1- Regular customer details(some sort of ID number)
- 2- Multiple search type either ID, Date, Name
- 3- Create new employee(waiter, operator)
- 4- Discount log maintenance
- 5- Create user (admin)
- 6- Printer linking and usage
- 7- Graph features i.e. top products, top customers, daily weekly and monthly sales record
- 8- Display of statistics every time.

Non- Functional requirements

- 1- Password Type
- 2- Online Order
- 3- Email based or phone number based logins
- 4- Deals type(daily, weekly, monthly)
- 5- Determine user type(admin/operator)
- 6- Graph type i.e. pie, bar etc.
- 7- Tax bar

High Risk Elements

1. Cancellation of order
2. Updating order
3. Late Order

Cancellation/updating of order will cause an error in system where you have no chance other than to cancel it will cause financial as well as system error. The system show the sale but actually it is not.

Solution:

We set a timeout after order in between that time user can change its decision and the order will be started to be prepared.

Late order will be dealt in the agreements and rules.

Addressing Most Critical Use Cases

Fully dressed Use Cases

Login

Actor: Operator and Admin

Stack Holder: Operator or Admin

Precondition: There must be an account of the user (admin/operator)

Post Condition: User successfully logged into the system.

Main Success Scenario: User start the system and enter its credentials. The system checked in its data base to verify its credentials. And message printed user successfully logged in.

Extensions: Credentials are incorrect and not logged inn.

Special Requirements: GUI

--

Add Customer

Actor: Operator

Stack holder: Operator want to enter the customer features to its list of customer in system.

Interests: Owner of the Fast food get new customer more business more profit.

Pre-Condition: Operator has an account and must have login before add customer.

Post-Condition: Customer successfully added to the system and updated in customer list.

Main Success Scenario: A customer arrived for buy/order the food, operator validate him, but he found he is new customer. Operator enter his/her credentials into the system and submit to system that he update it. A new customer account been created and issued a card/ID to customer.

Extensions: Customer is already registered or has credentials insufficient.

Special Requirements: Separate UI.

--

Take Order

Actor: Operator

Stack Holders: Operator wants to enter the items of order from customer.

Interests: Owner

Pre-Conditions: Operator have a login ID and successfully logged to the system and customer must be in the customer list.

Post-Conditions: Order successfully placed and an auto generated receipt of the order created.

Main Success Scenario: Customer came to place his/her order. Operator asked about regular or first time customer. Customer if regular he will give his/her ID, and if the ID valid. The operator start to place his/her order and add his items of interest. At the end of submission the total amount to pay will be shown to customer. Customer pay the amount and wait for the order.

Extensions: Customer is not valid or new customer than the order may not be placed. If he did not have the total amount to pay also order must be cancelled.

Special Requirements: GUI which shows Items added of order.

--

Cancel Order

Actors: Operator

Stack Holders: Operator want to cancel the order of the person with in due time on the concern of Customer.

Interests: Customer

Pre-Conditions: There must be an order placed by customer.

Post-Conditions: The customer order easily cancelled by operator.

Main Success Scenario: Customer ordered the food. For the time being i.e. in time limit he want to cancel his/her order may be time issue or anything else. Operator check his her order and cancel the order of the customer.

Extensions: No order is placed and the time limit is exceeded.

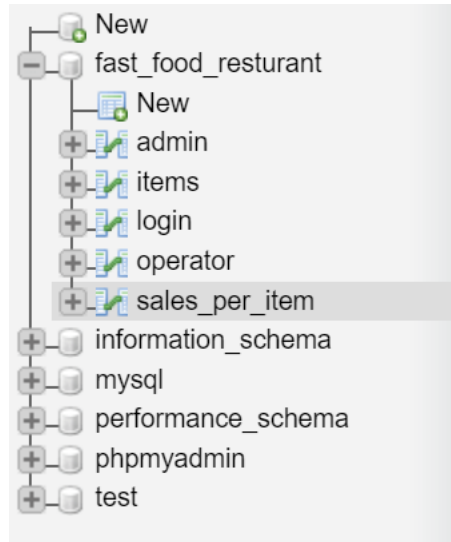
Special Requirements: No

--

3rd Deliverable: Elaboration Phase Iteration 2(prototypes were designed and Data Model was Designed)

Data model

- Till now we have created 5 tables based on our requirements:



➤ Tables Structure

1- Admin:

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---------------------|-------------|-------------------|------------|------|---------|----------|-------|---|
| 1 | Name | varchar(20) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index ▼ More |
| 2 | ID | varchar(10) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index ▼ More |
| 3 | Phone_Number | varchar(11) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index ▼ More |
| 4 | Gender | char(1) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index ▼ More |
| 5 | CNIC | varchar(20) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index ▼ More |
| 6 | Password | varchar(20) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index ▼ More |
| 7 | Address | varchar(50) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index ▼ More |

2- Items:

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|----------------|-------------|-------------------|------------|------|---------|----------|-------|--|
| 1 | Item_Name | varchar(20) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index More |
| 2 | Per_Unit_Price | int(30) | | | No | None | | | Change Drop Primary Unique Index More |

3- Login:

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|-----------|-------------|-------------------|------------|------|---------|----------|-------|--|
| 1 | User_name | varchar(30) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index More |
| 2 | User_Id | varchar(20) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index More |
| 3 | Password | varchar(20) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index More |
| 4 | User_type | varchar(10) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index More |

4- Operator:

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|--------------|-------------|-------------------|------------|------|---------|----------|-------|--|
| 1 | Name | varchar(20) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index More |
| 2 | ID | varchar(10) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index More |
| 3 | Phone_Number | varchar(11) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index More |
| 4 | Gender | char(1) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index More |
| 5 | CNIC | varchar(20) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index More |
| 6 | Password | varchar(20) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index More |
| 7 | Address | varchar(50) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique Index More |

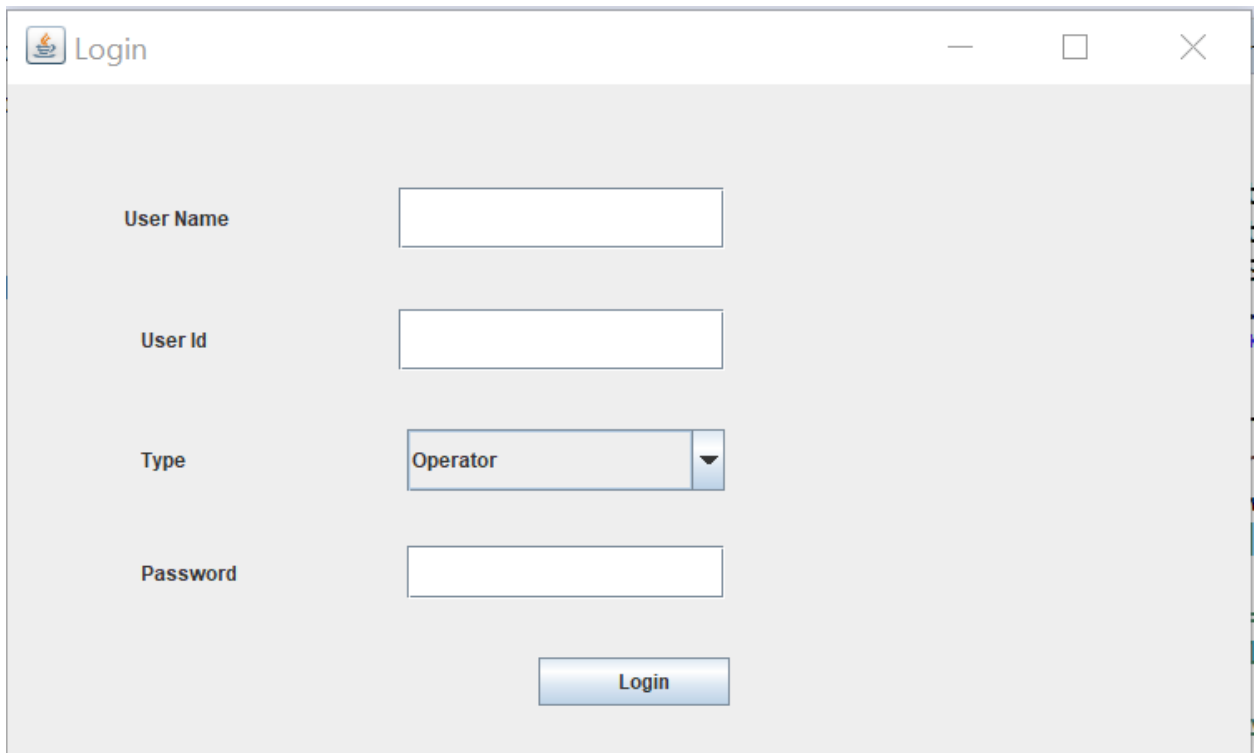
5- Sales per item:

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|----------------|-------------|-------------------|------------|------|---------|----------|-------|-------------------------------------|
| 1 | Date | date | | | No | None | | | Change Drop Primary Unique More |
| 2 | Item_Name | varchar(20) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique More |
| 3 | Quantity | int(10) | | | No | None | | | Change Drop Primary Unique More |
| 4 | Per_Unit_Price | int(30) | | | No | None | | | Change Drop Primary Unique More |
| 5 | Total Price | int(40) | | | No | None | | | Change Drop Primary Unique More |
| 6 | Customer_Name | varchar(20) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique More |
| 7 | Phone_Number | varchar(11) | latin1_swedish_ci | | No | None | | | Change Drop Primary Unique More |

Class Implementation description

➤ Login class

It maintains connection and gets data and check it from database. And then based on type it move to Operator and Admin page.

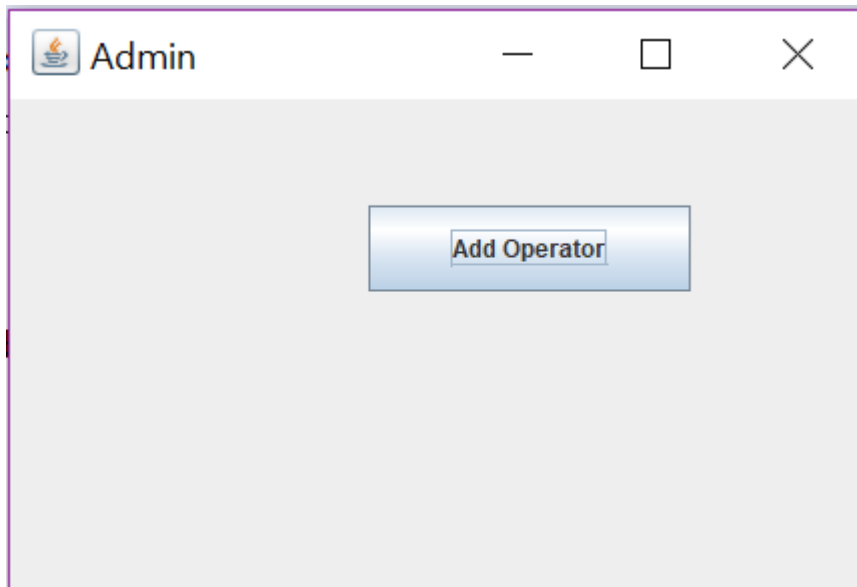


The screenshot shows a 'Login' window with the following fields and controls:

- User Name**: A text input field.
- User Id**: A text input field.
- Type**: A dropdown menu with 'Operator' selected.
- Password**: A text input field.
- Login**: A button at the bottom center.


➤ Admin class

Only add operator facility is given to admin till now.



➤ Add operator class

It add data to the login table and operator table in database

 **AddOperator** — □ ×

| | |
|------------------------------------|----------------------------------|
| Name | <input type="text"/> |
| Id | <input type="text"/> |
| Phone Number | <input type="text"/> |
| Gender | <input type="text" value="M"/> ▼ |
| CNIC | <input type="text"/> |
| Password | <input type="text"/> |
| Address | <input type="text"/> |
| <input type="button" value="Add"/> | |

Prototypes:

- Prototypes of Admin Class



Fast Food Restaurant

Add/Remove operator

Remove Customer

Graph Out

Bonus

◀ Back

Home



Fast Food Restaurant

Add/Remove operator

Operator ID

Remove

Add

◀ Back

Home



Fast Food Restaurant

Bonus

Operator ID

Submit

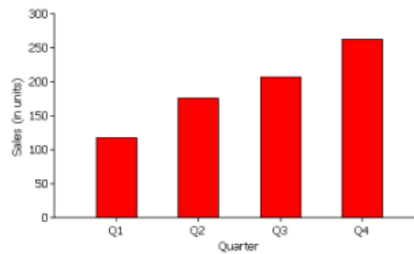
◀ Back

Home



Fast Food Restaurant

Graph Out



[<Back](#)

[Home](#)



Fast Food Restaurant

Remove customer

Customer ID

Remove

[<Back](#)

[Home](#)



Operator class prototypes



Fast Food Restaurant

Add Customer

Add

[< Back](#)

[Home](#)



Fast Food Restaurant

Take Order

Add Customer

[< Back](#)

[Home](#)



Fast Food Restaurant

User Name / Email

Password

LOGIN

[Forget Password](#)



Fast Food Restaurant

Take Order

Add Items

Add

Submit

[Back](#)

[Home](#)

4th Deliverable:

Elaboration Phase Iteration 3(UML diagrams for other requirements were designed)

UML Diagrams are listed below.

5th Deliverable:

Elaboration Phase and Construction Phase (Designed requirements were implemented)

UML Diagrams are listed below.

6th Deliverable:

Construction Phase (All requirements were completely implemented).

8. Assumptions

Our project will be able to maintain the sales. Calculate Gain and loss. Give graphical representation of data. Extract top 3 customers of month. Manage operators. Handle orders.

Our project is going to be the Application that manages all above mentioned things with just click.

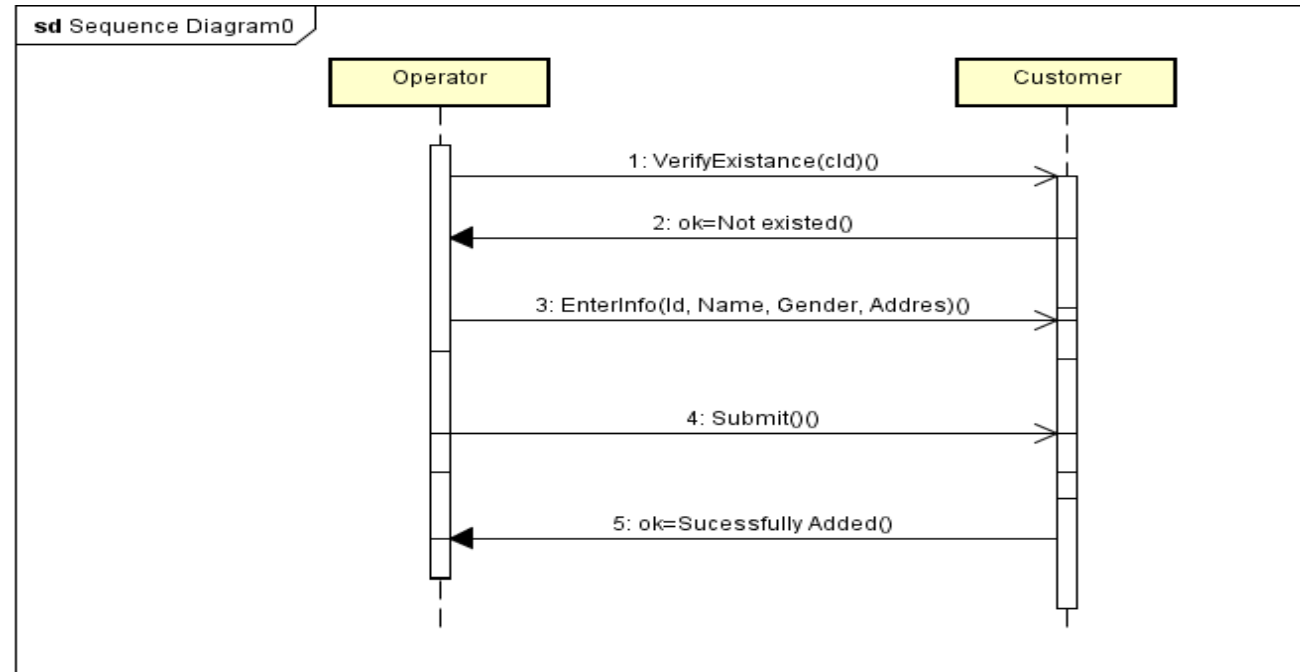
9. Proposed Solution

We will attempt the solution with the help of concept of classes, database, MVC and PDF converter. We have prepared GUI that will be used for Human Computer Interaction. GUI will take data from User and forward it to the related classes. The requirement we will attempt to fulfil, we will in UML Diagrams design it first.

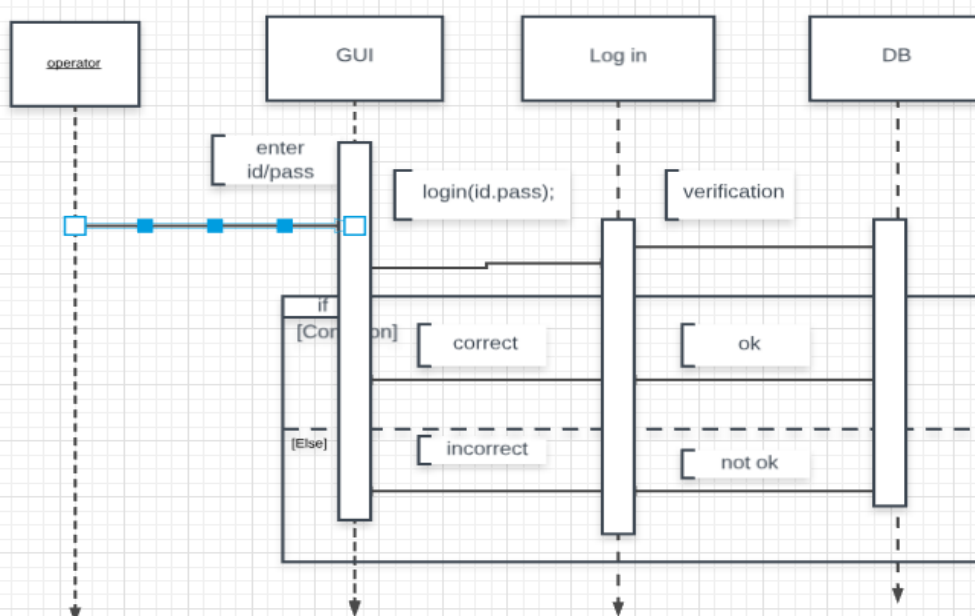
11. UML Diagrams:

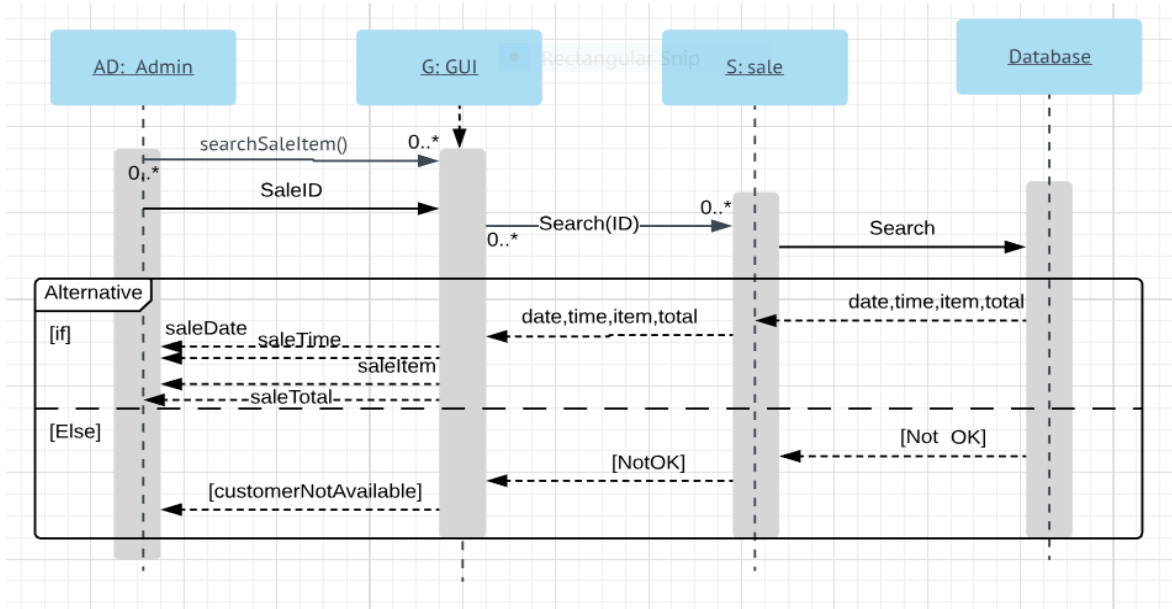
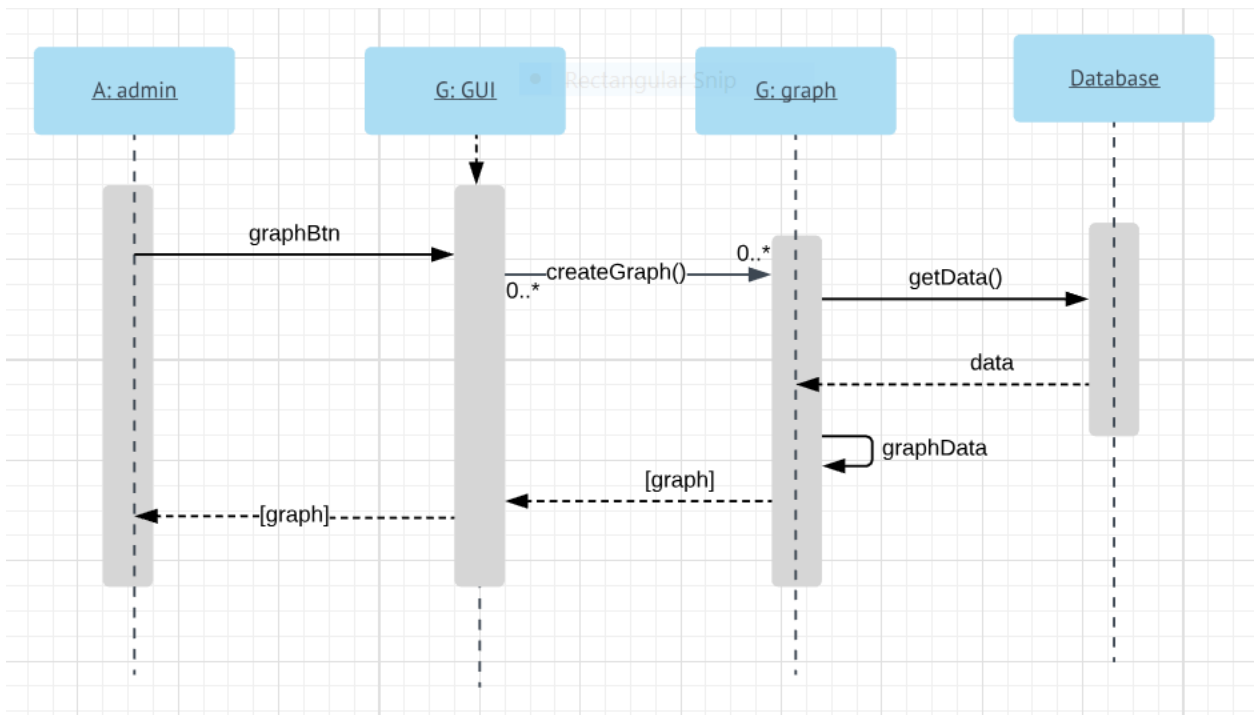
Sequence Diagrams:

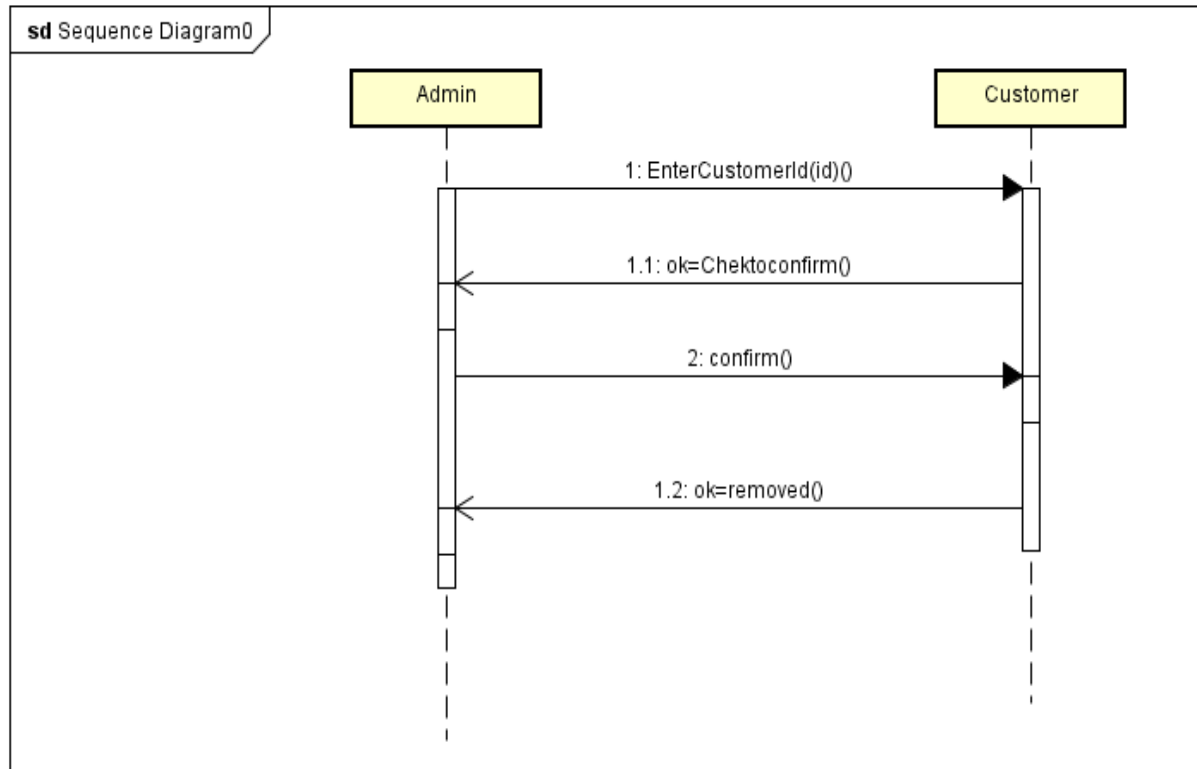
Add Customer:



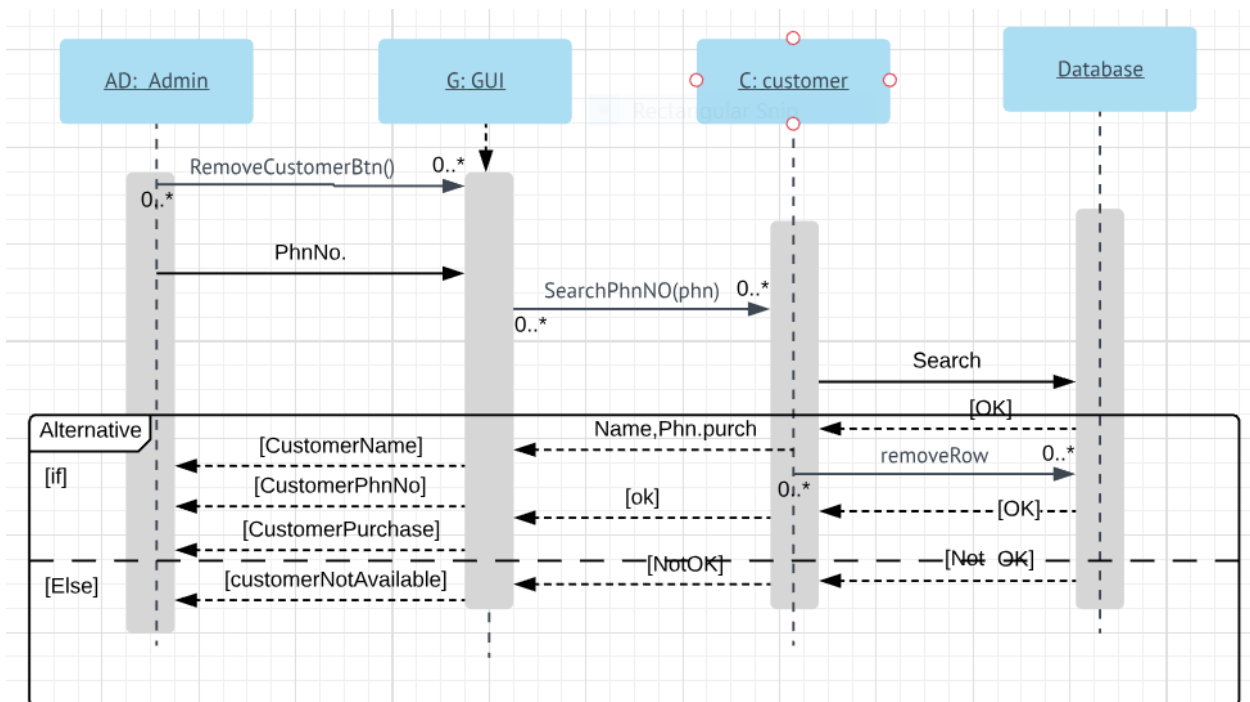
Login:

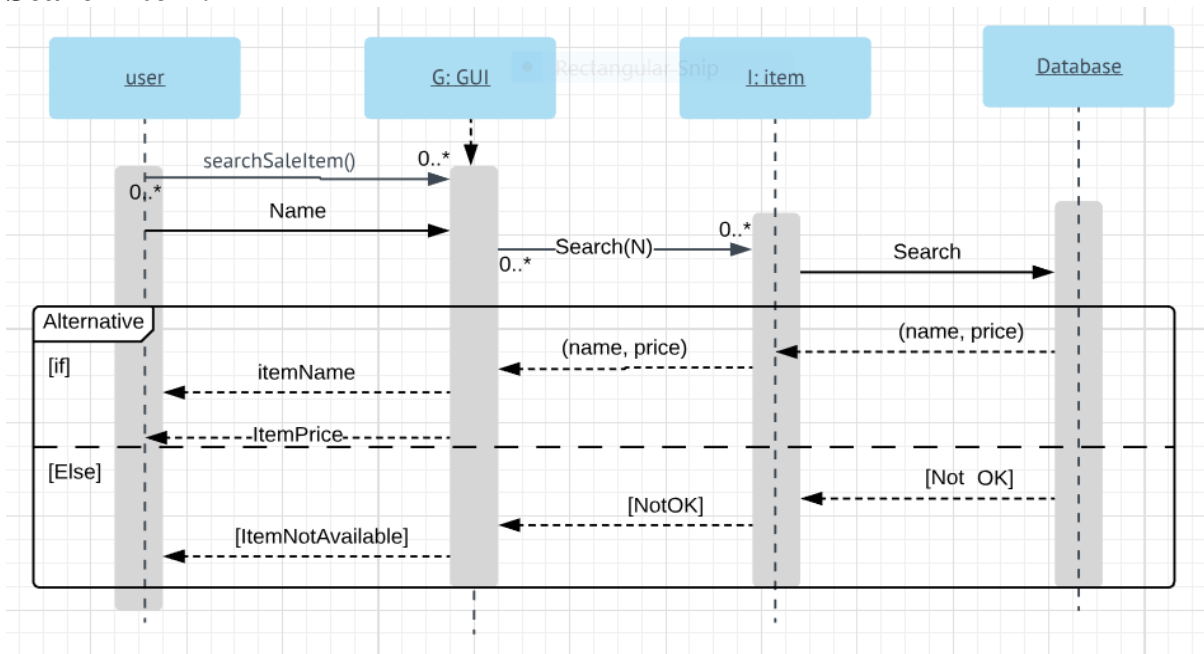
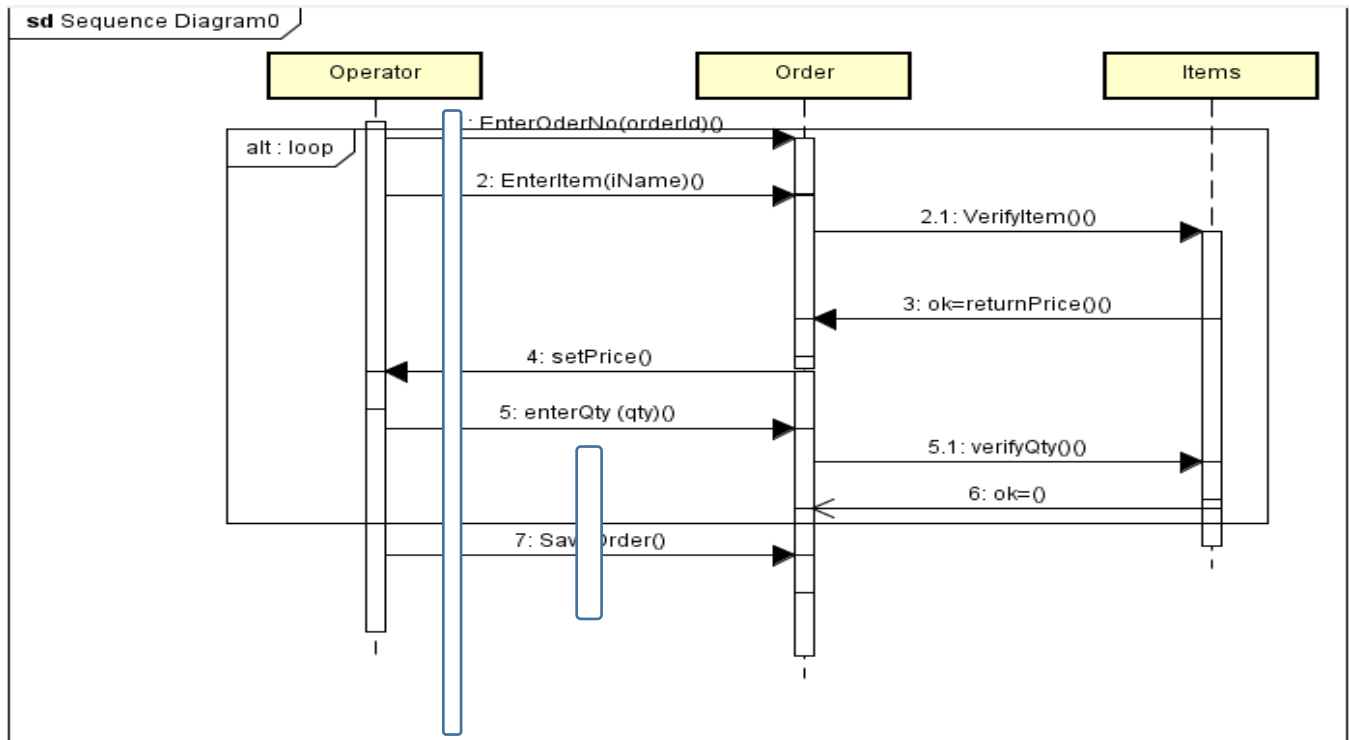


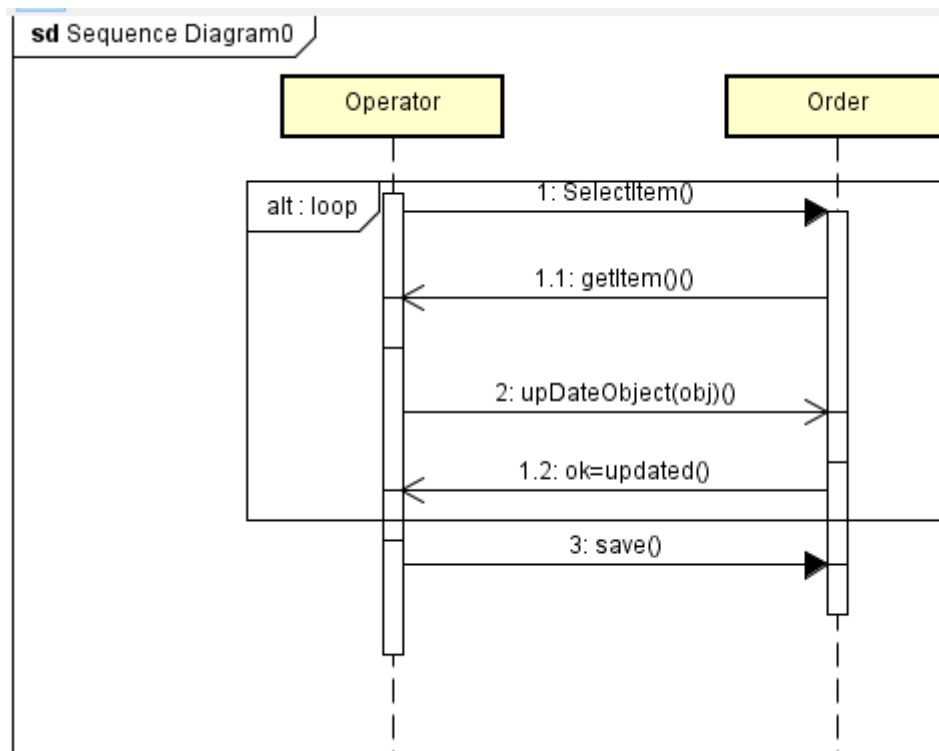
Search Sale:**Graph:****Remove Customer:**



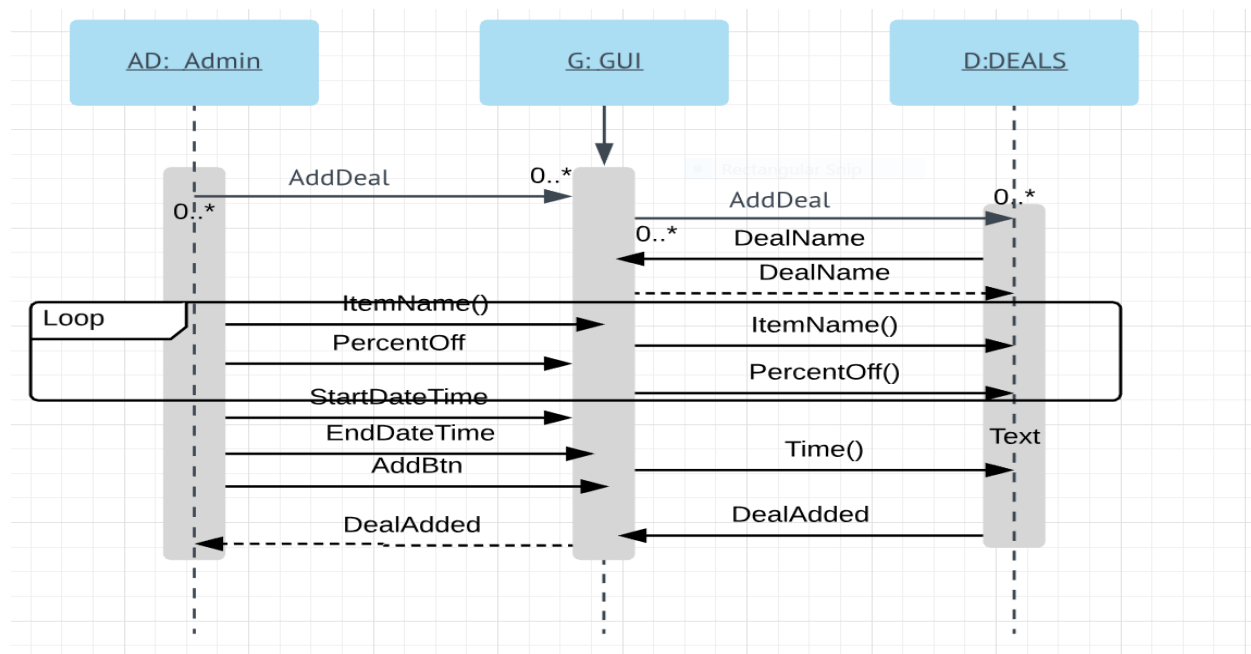
Search Customer:



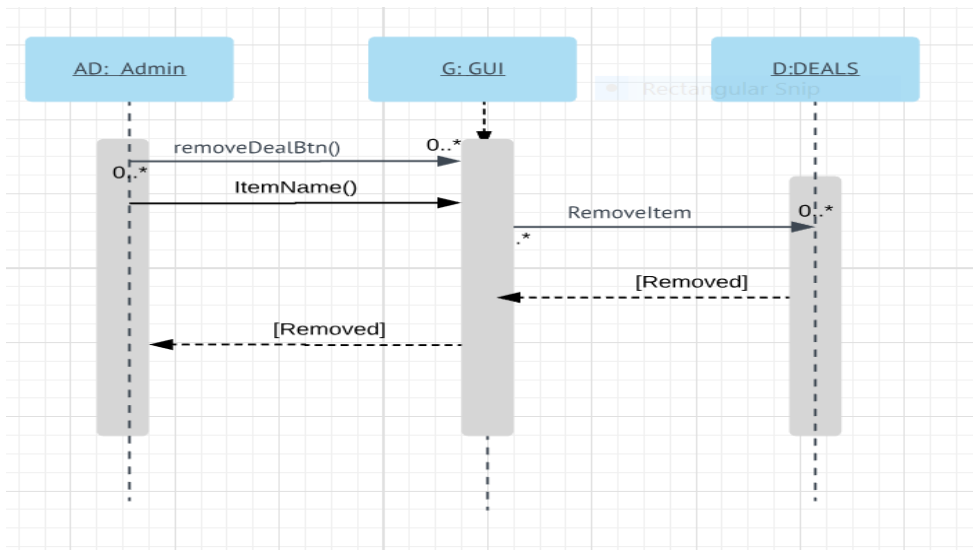
Search Item:**Take Order:****Update Order:**



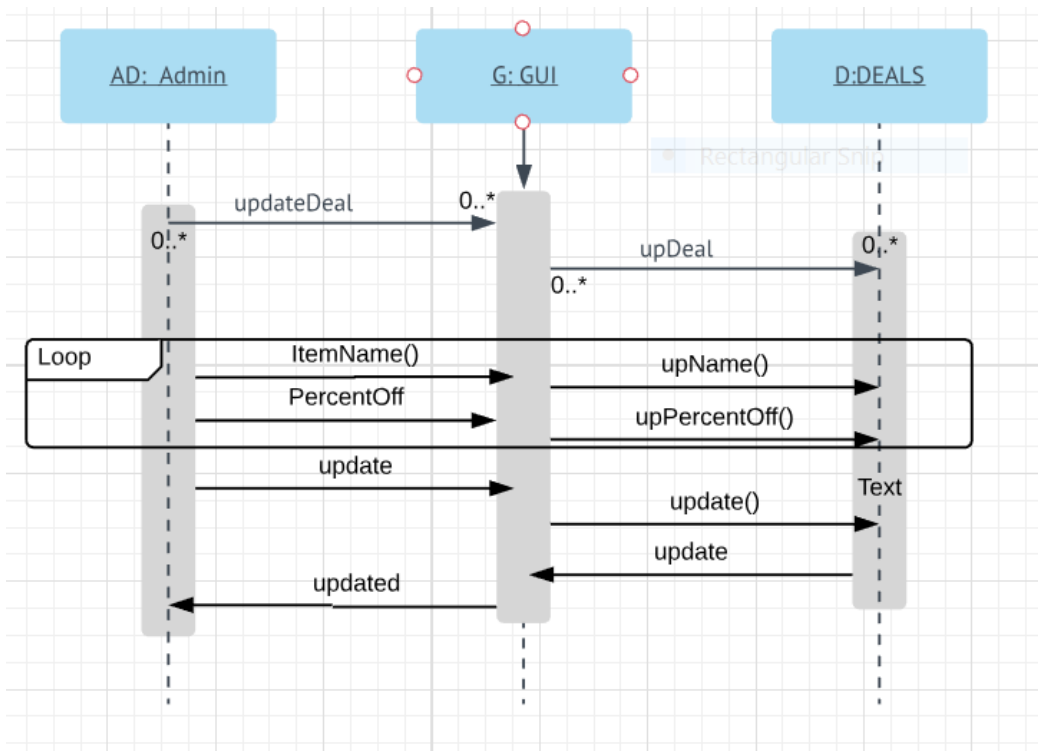
Add Deal:



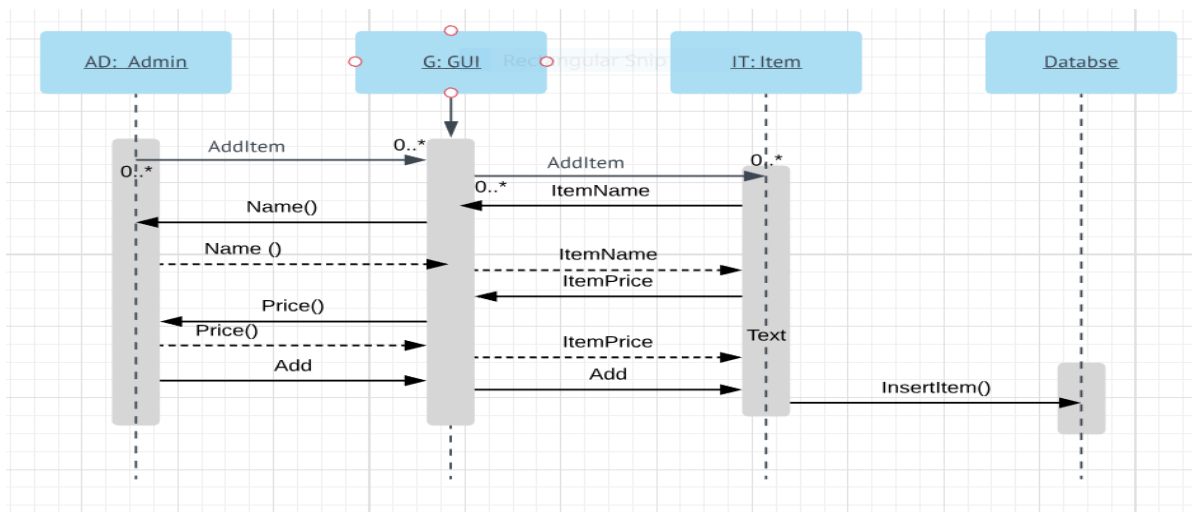
Remove Deal:



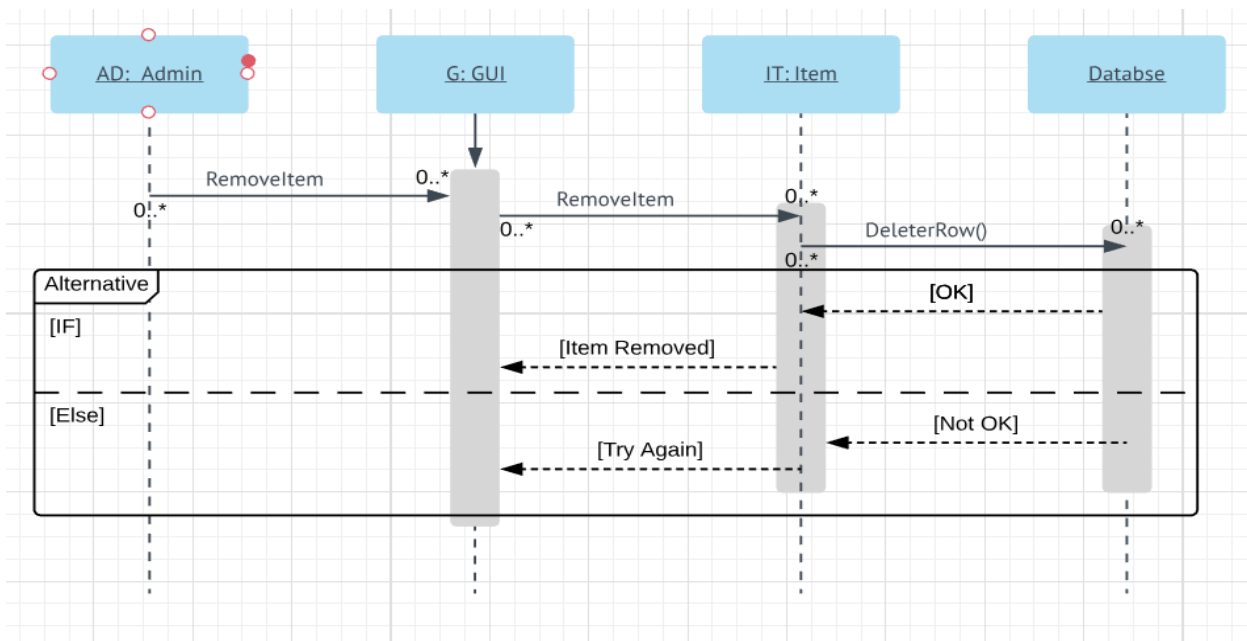
Update Deal:



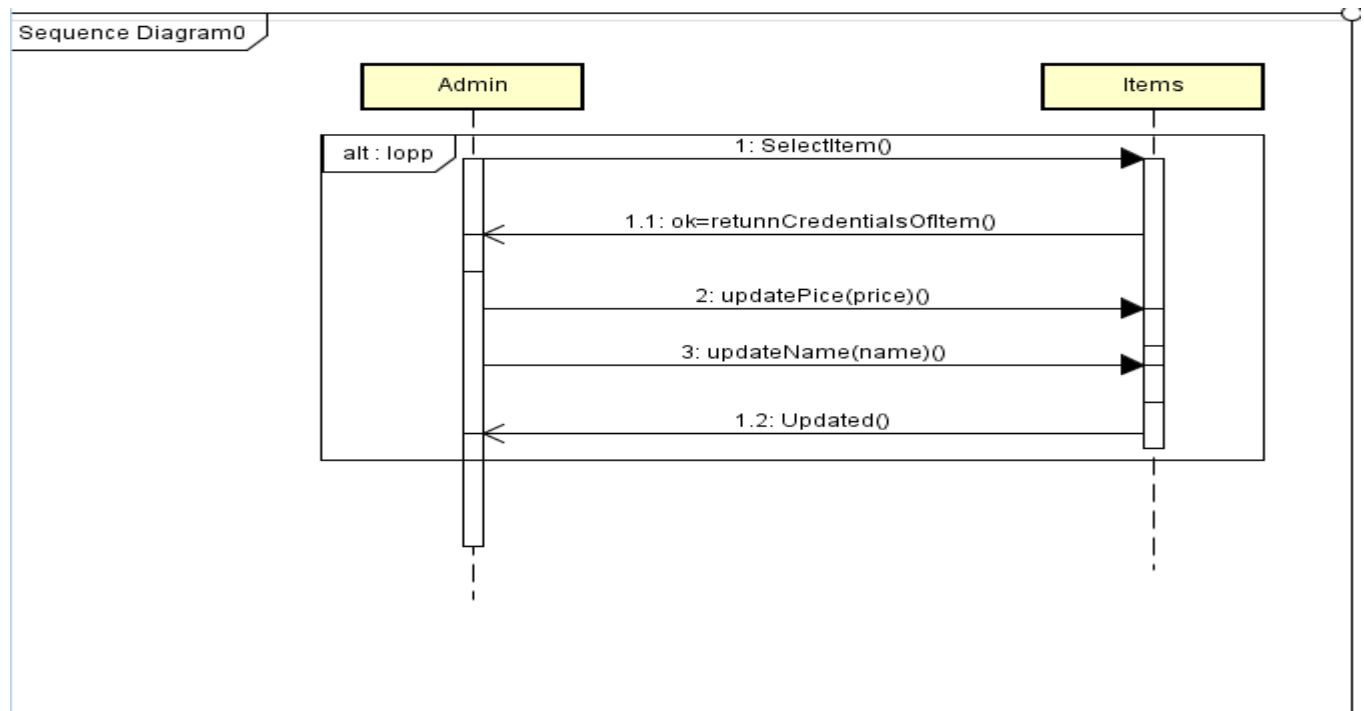
Add Item:



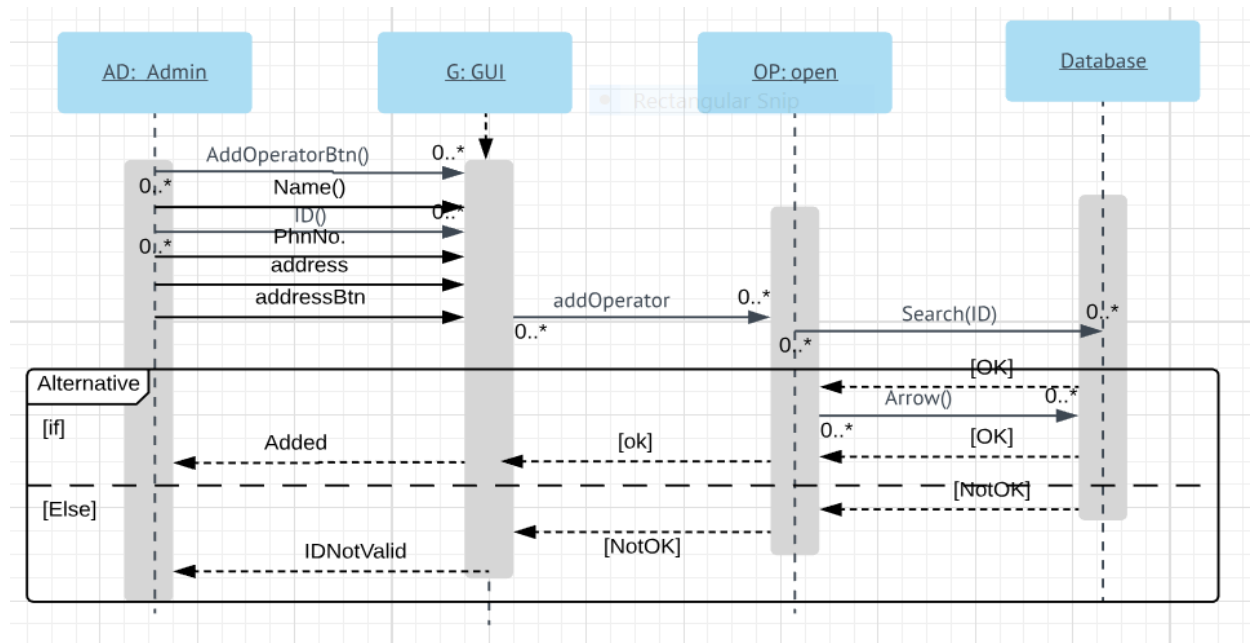
Remove Item:



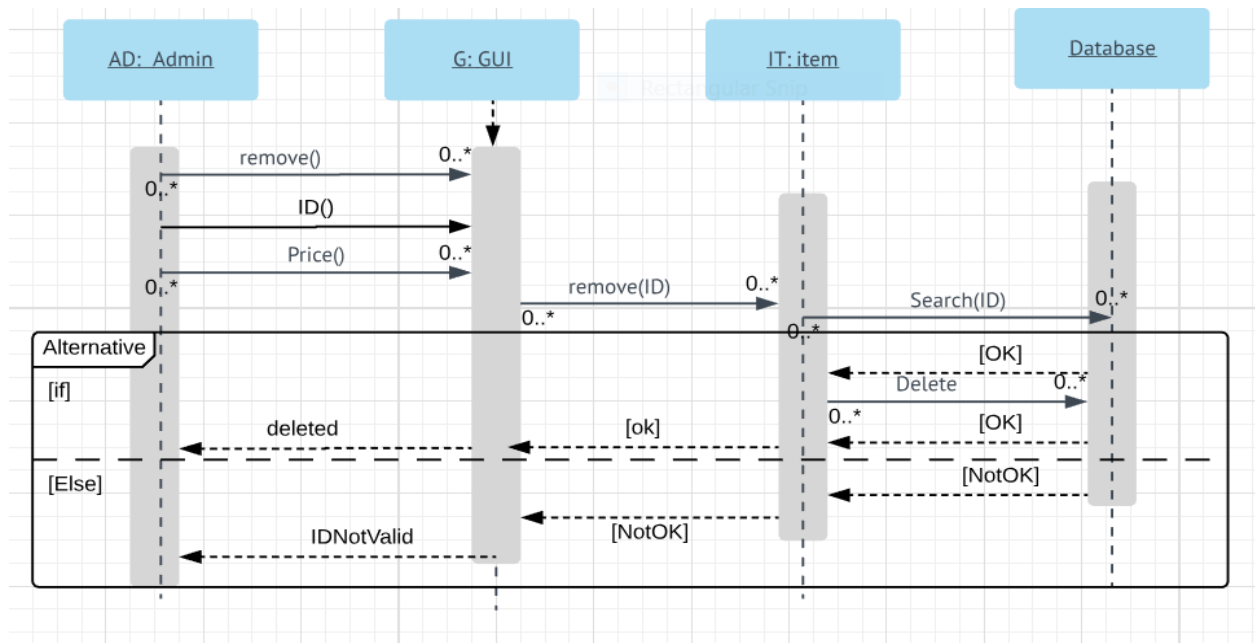
Update Item:



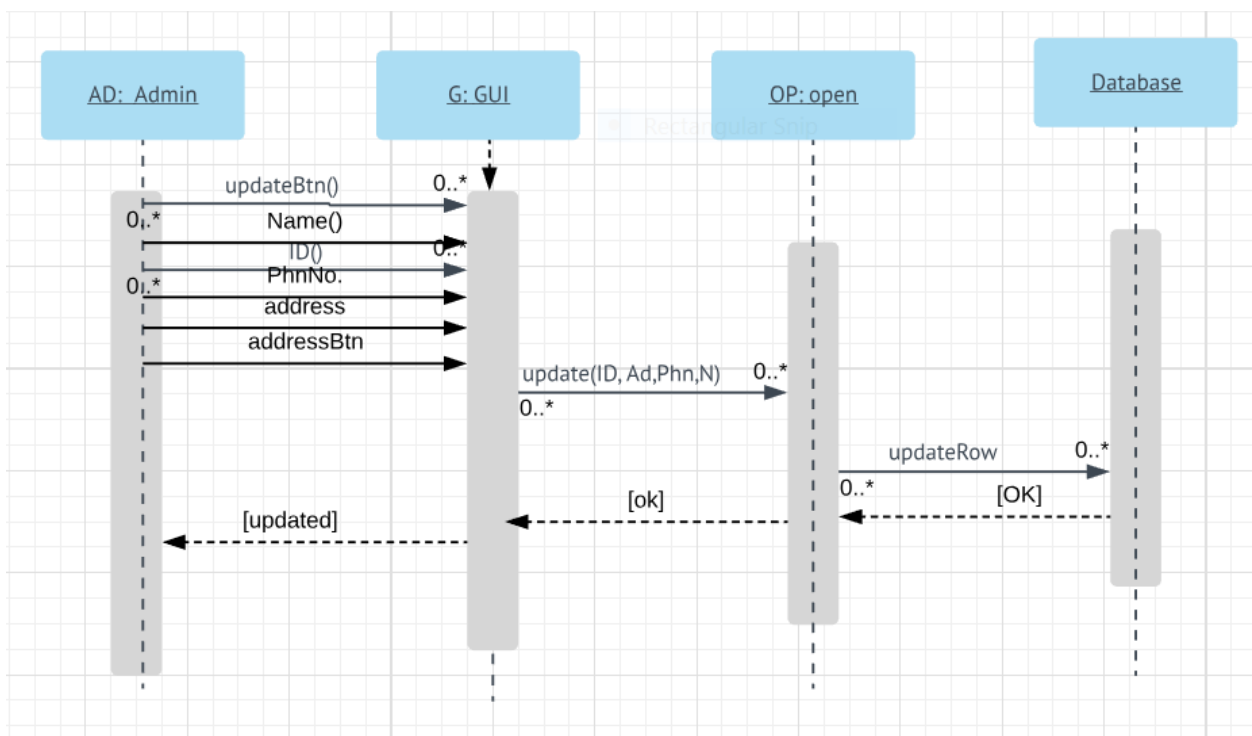
Add Operator:



Remove Operator:

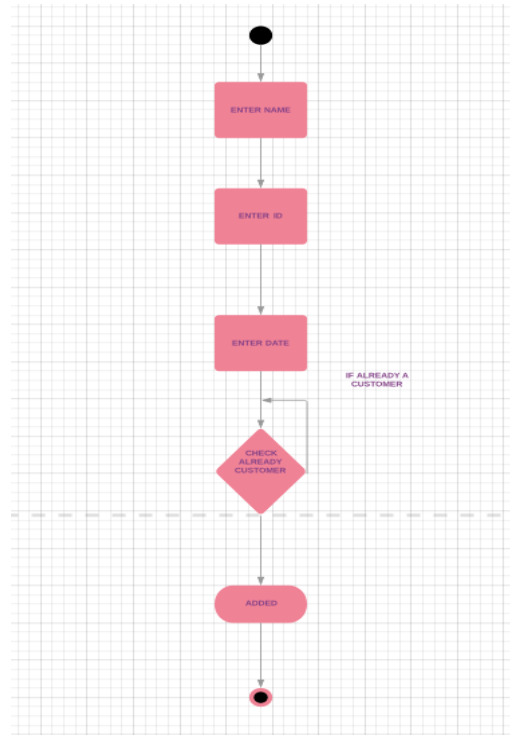


Update Operator:

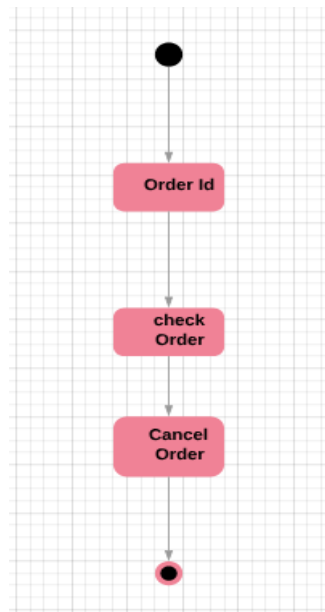


Activity Diagrams

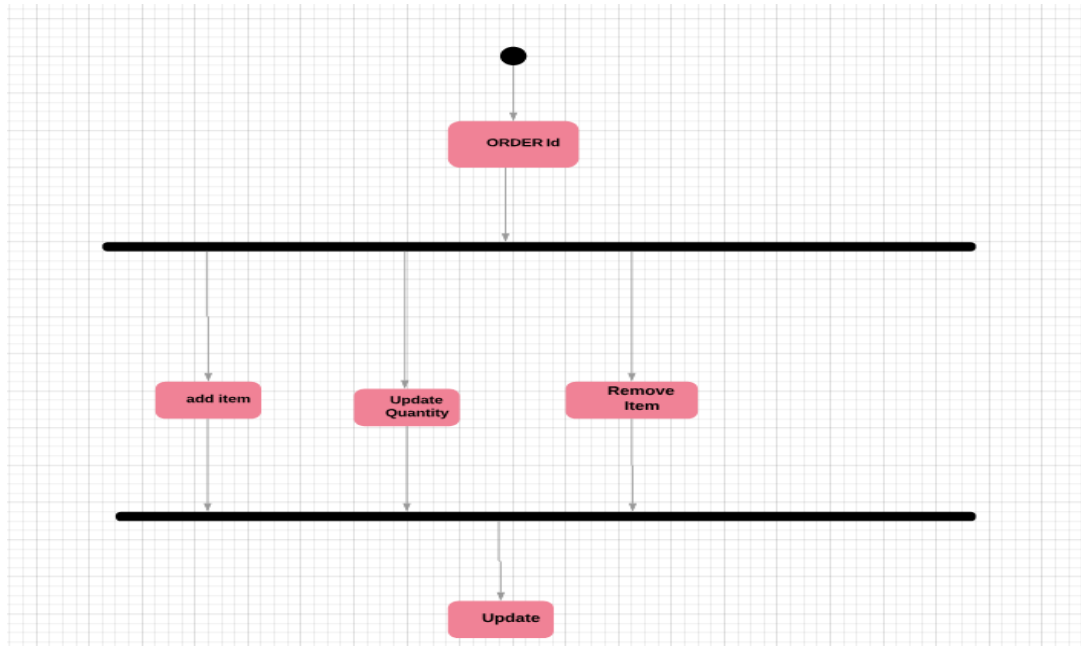
Add Customer:



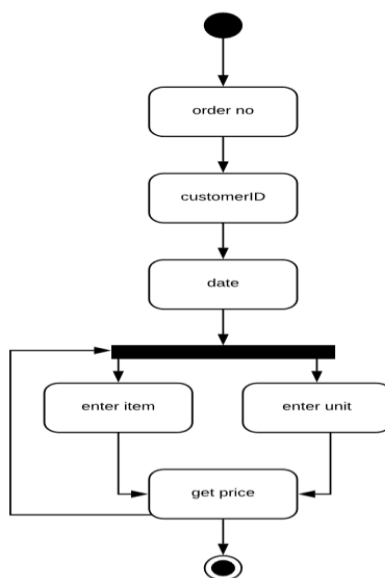
Cancel Order:



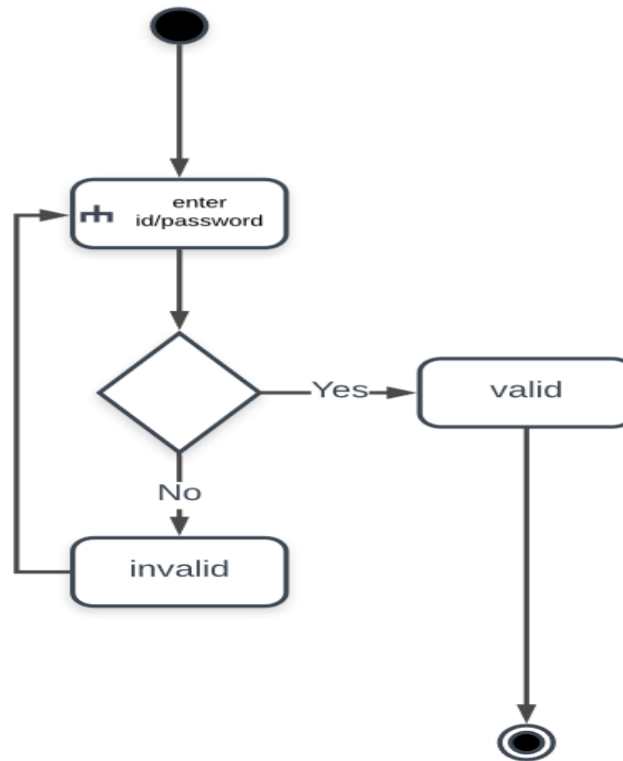
Update Order:



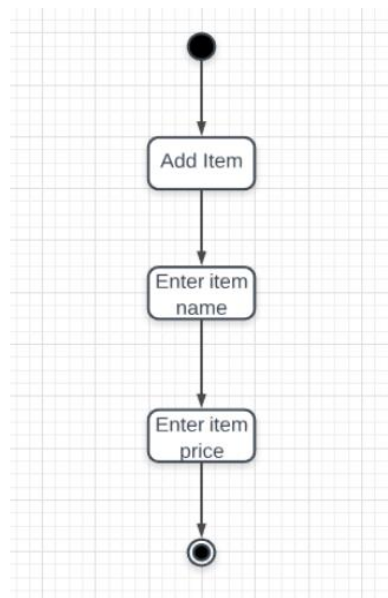
Take Order:



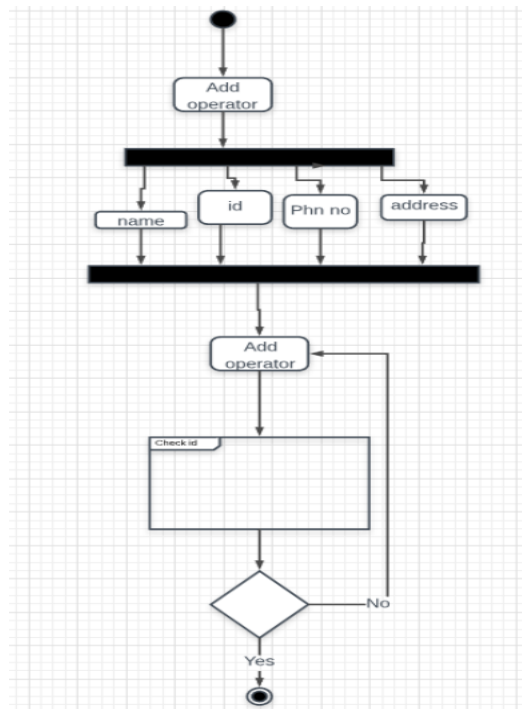
Login:



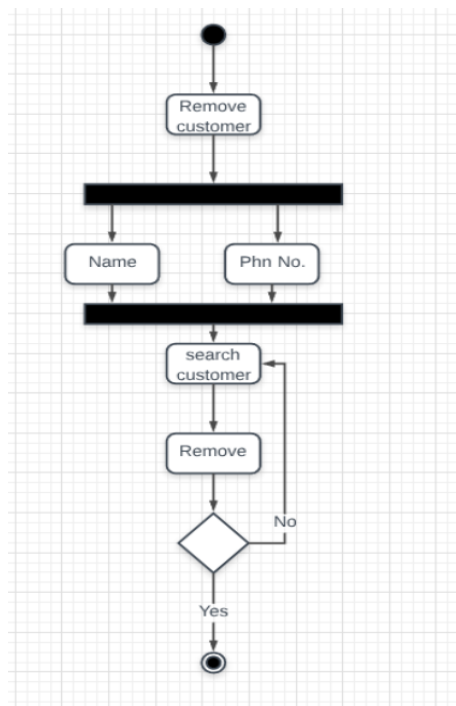
Add Item:



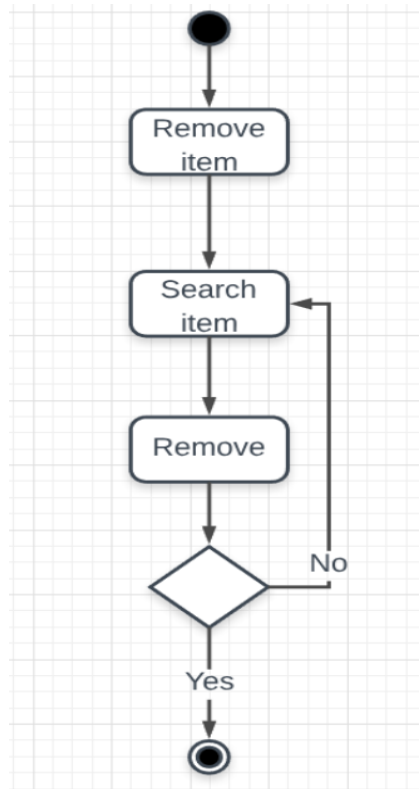
Add Operator:



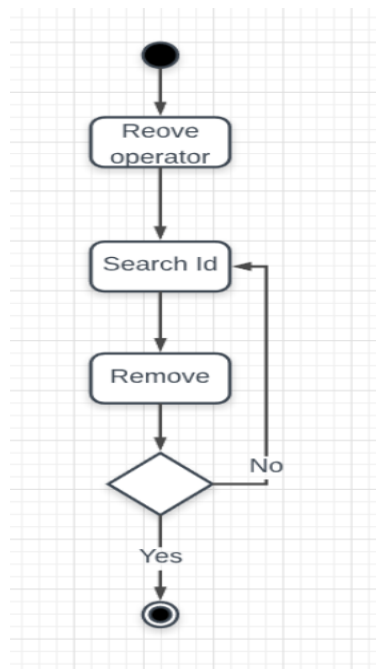
Remove Customer:



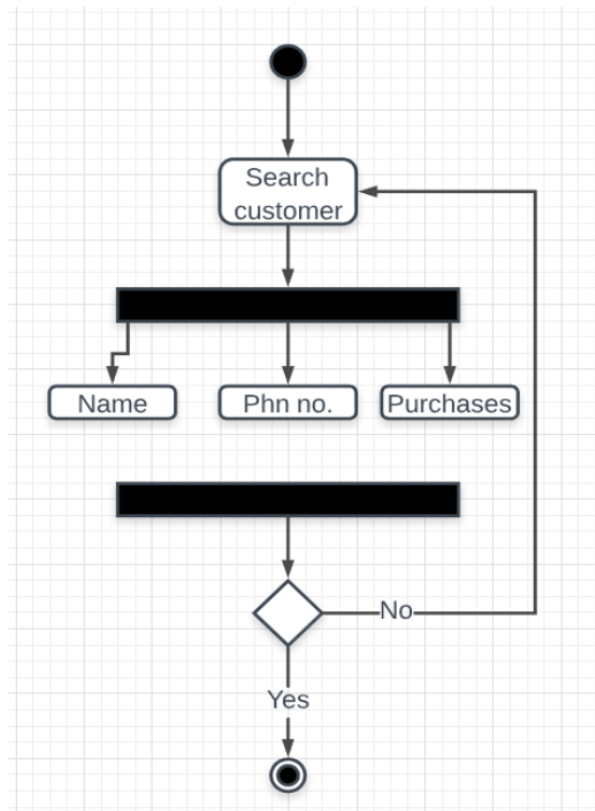
Remove Item:



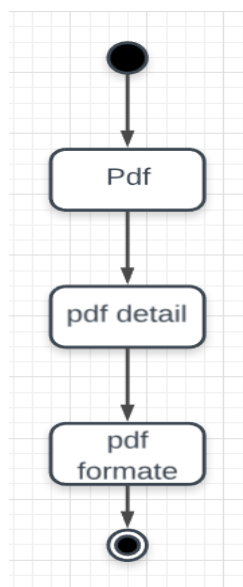
Remove Operator:



Search Customer:



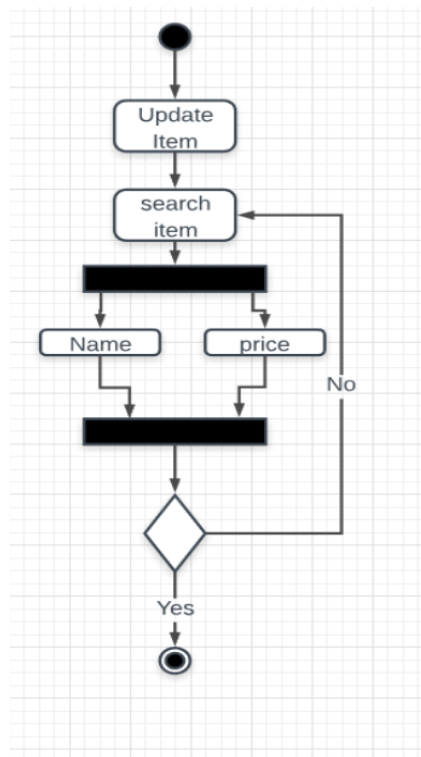
PDF:



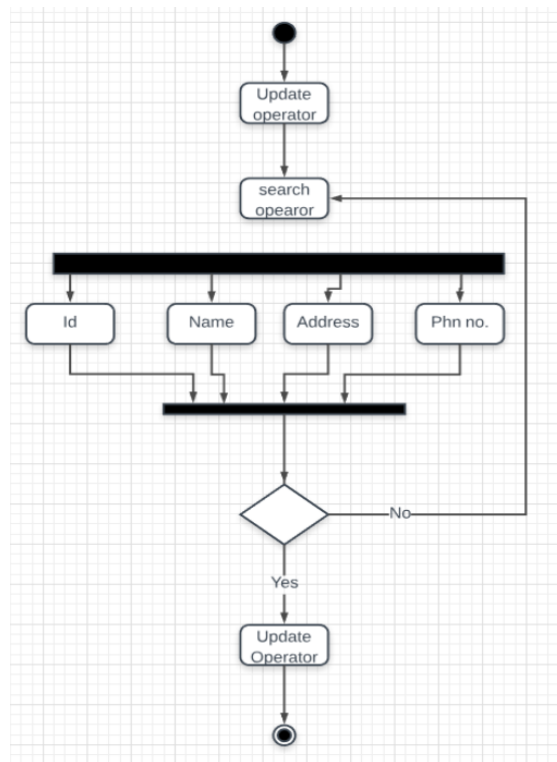
Graph:



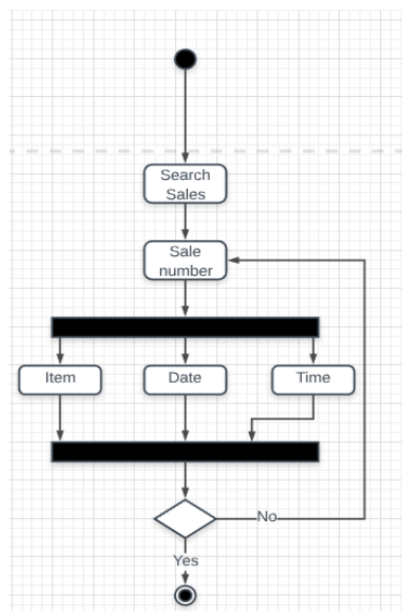
Search Item:



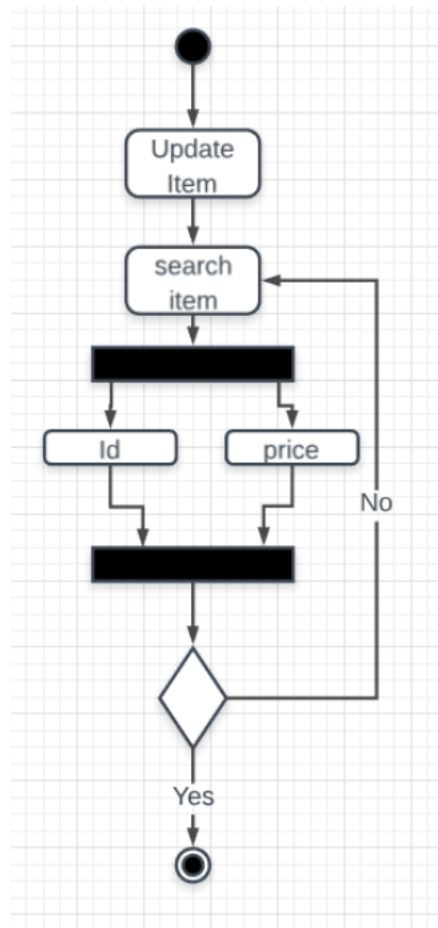
Update Operator:



Search sale:

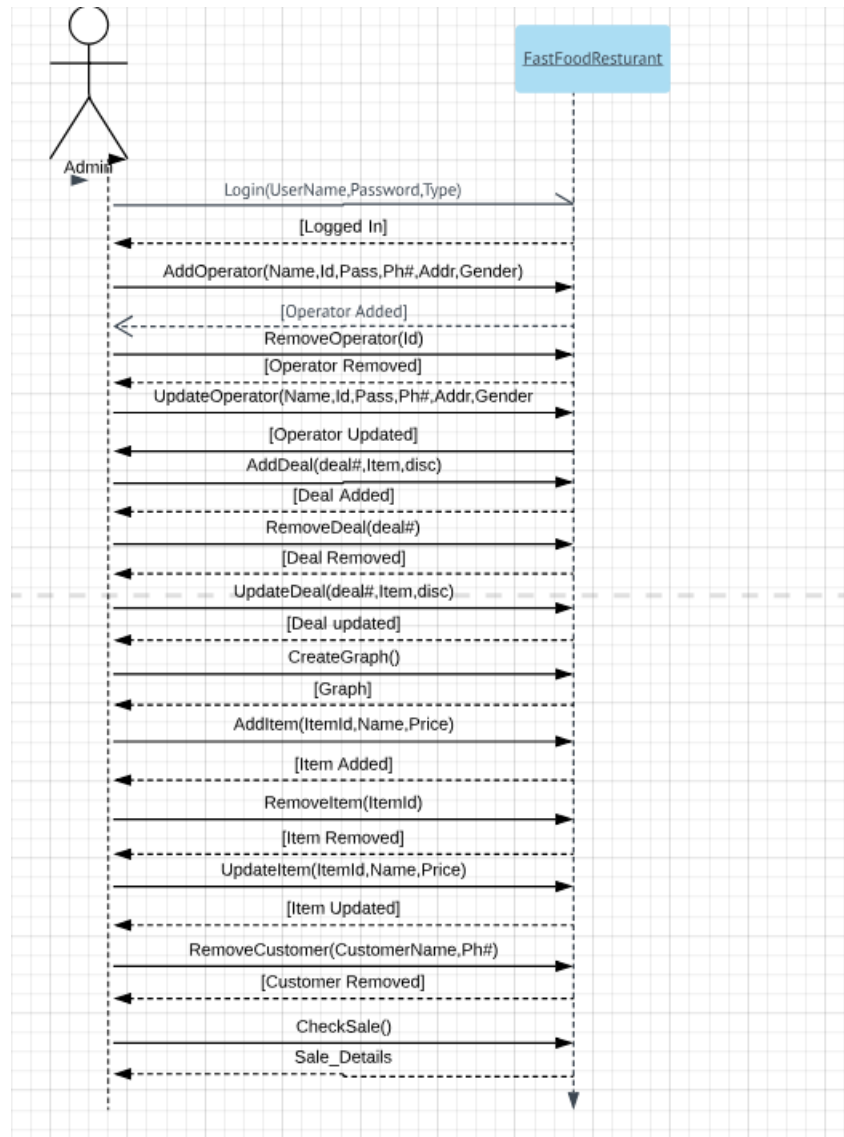


Update Item:

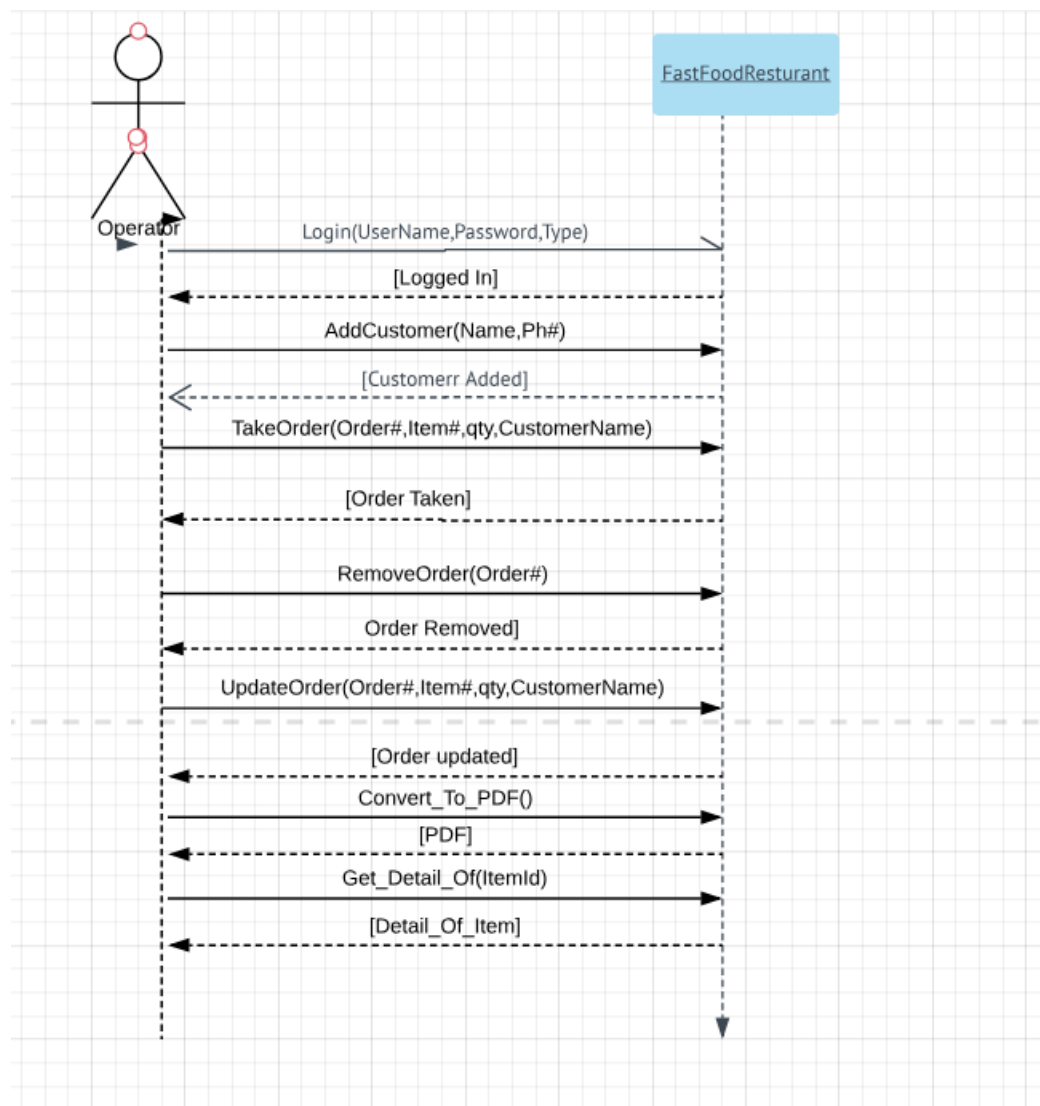


System Sequence Diagram

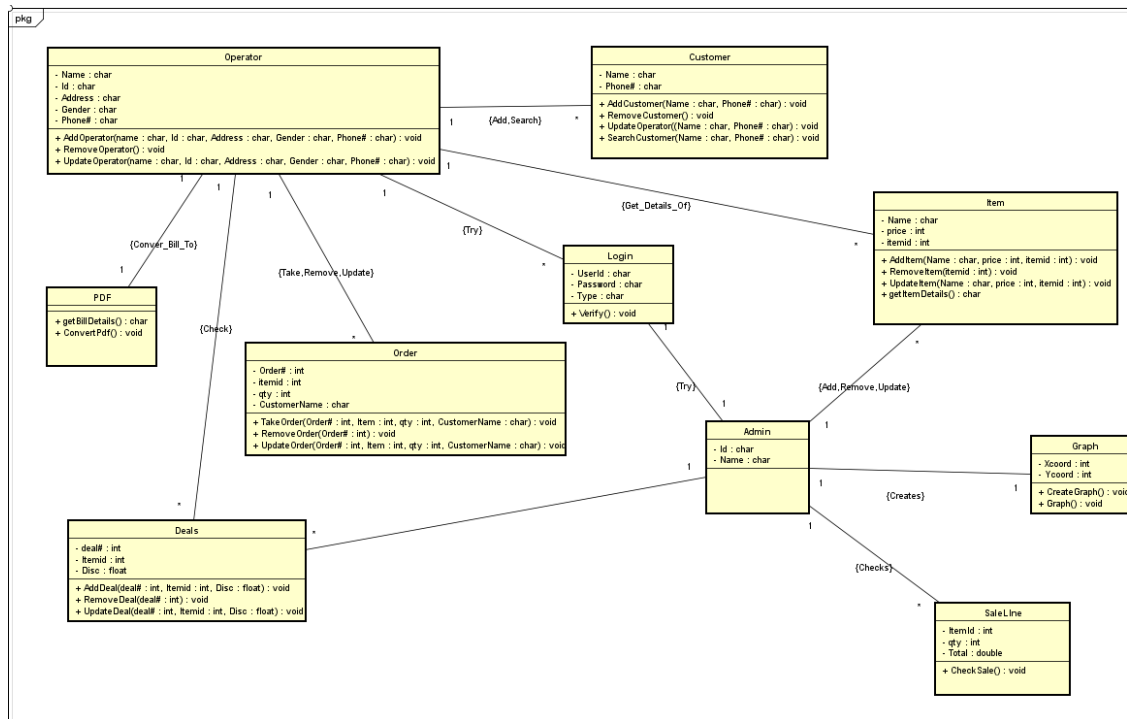
Admin:



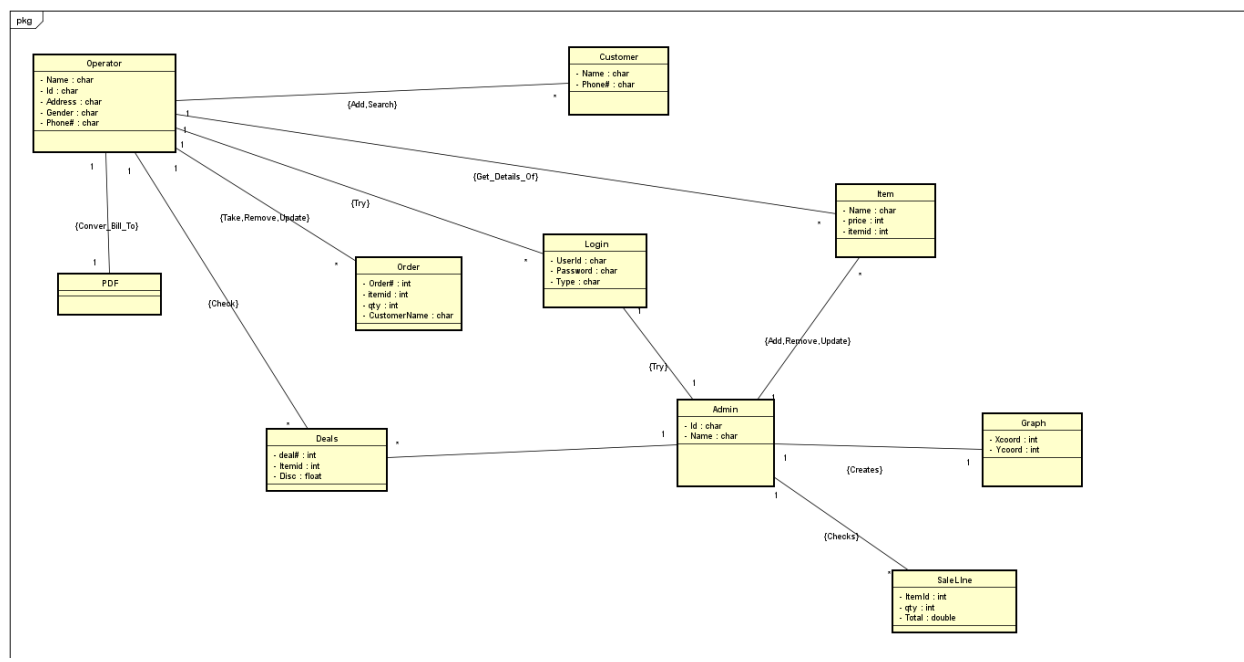
Operator:



Class Diagram:

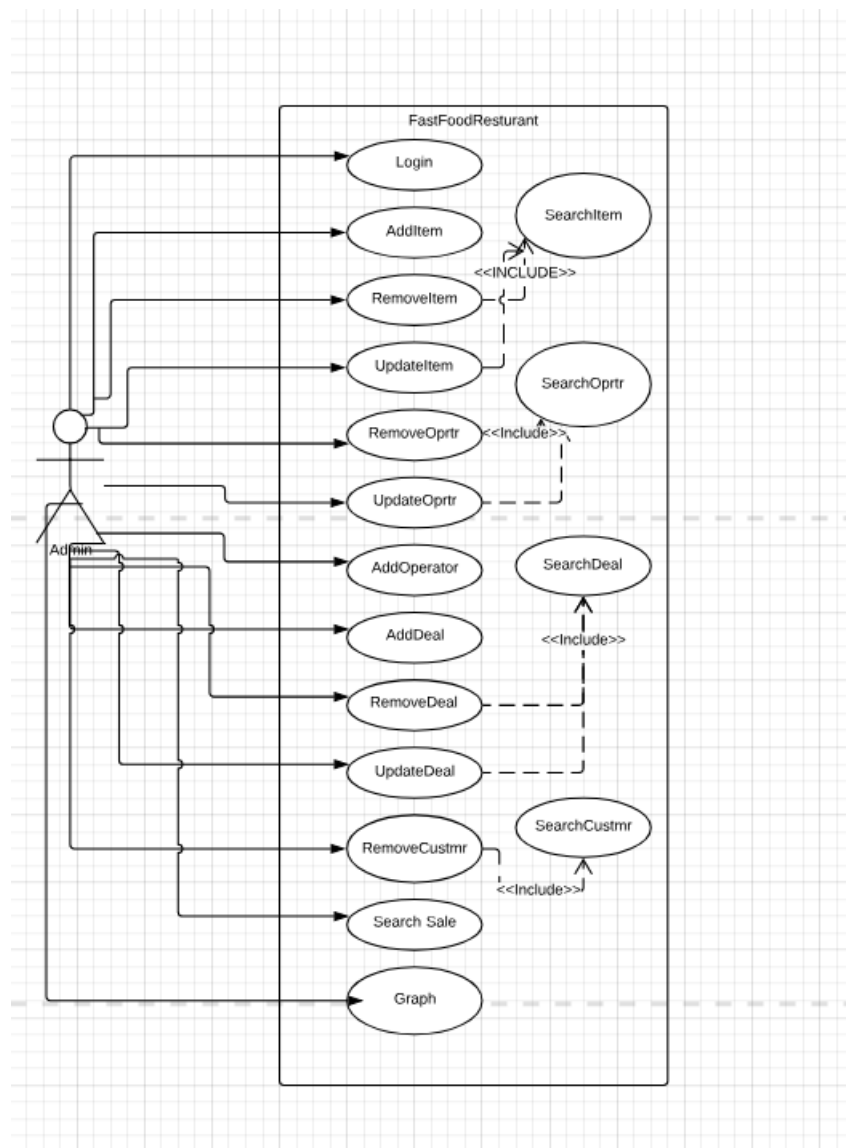


Domain Model:

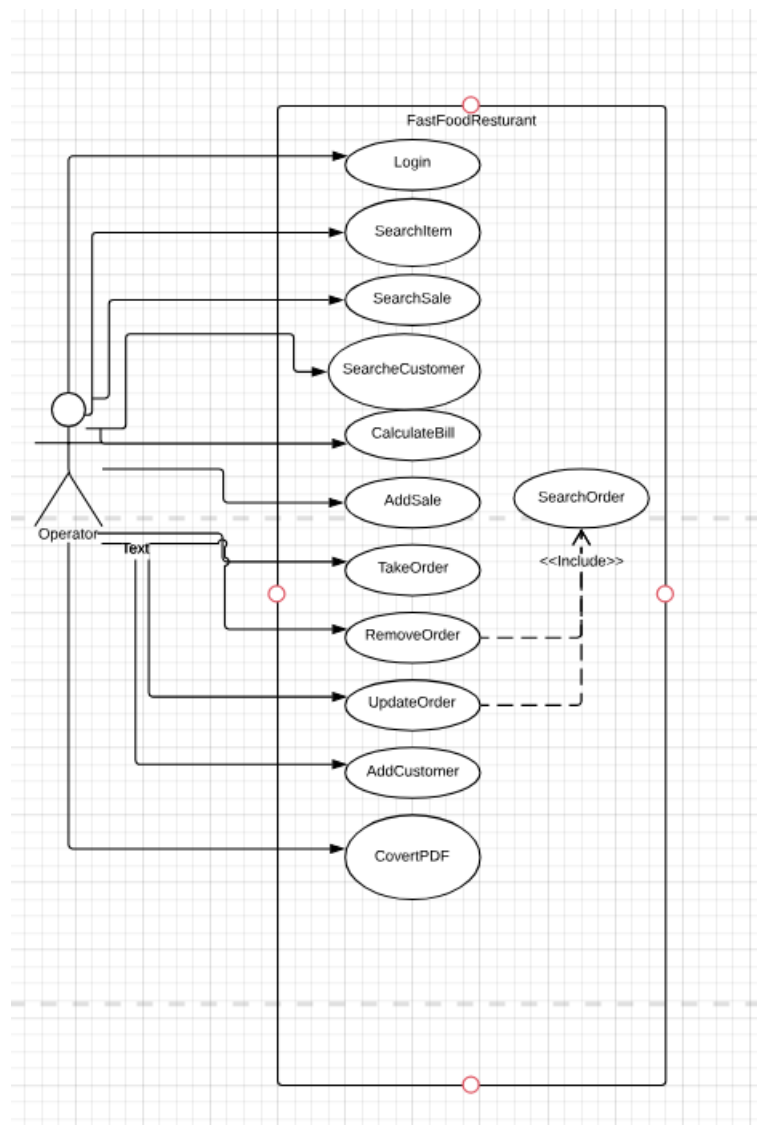


Use Case Diagram

Admin:

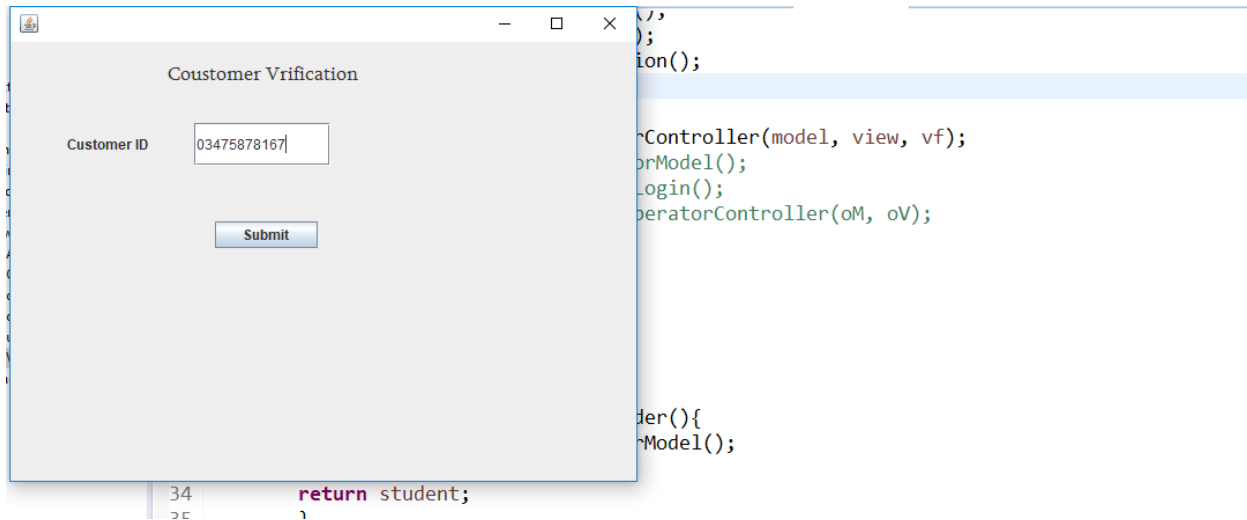


Operator:

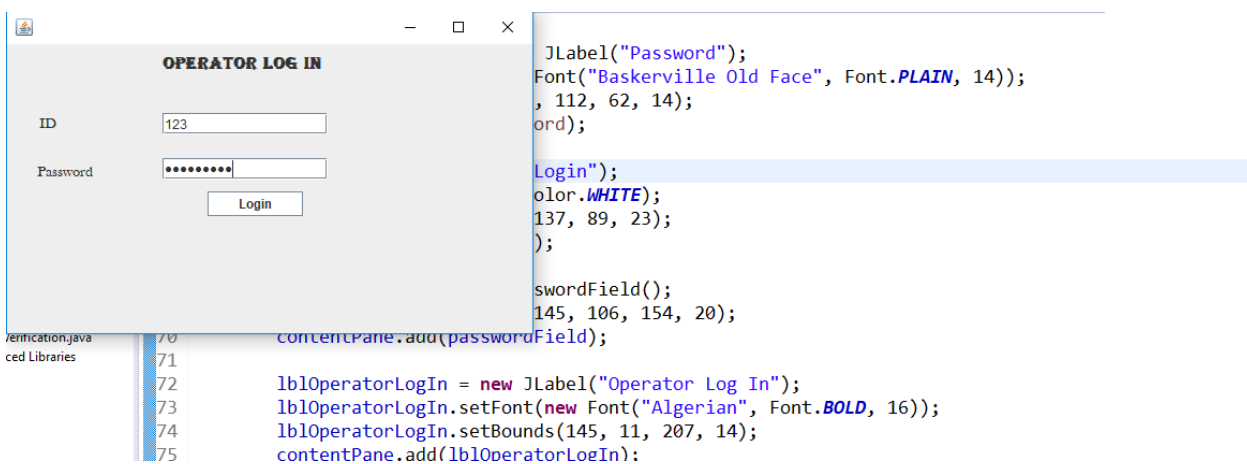


12. Screen Shots of output

Customer Verification:



Login:



Take Order:

Order No. 4

Customer ID 03475878167

Date 2018-54-04

Product Unit Price

Zinger 200 Add

| Product Name | Unit | Price |
|--------------|------|-------|
| Large Pizza | 2 | 2600 |
| Medium Pizza | 1 | 900 |
| Zinger | 2 | 400 |

3900

Save

```

e");
);
, Font.PLAIN, 16));

ingConstants.CENTER);
);
28);

eateLineBorder(Color.WHITE, 5);

);
14);
  
```

Update Order:

Product Unit Price

Update

| Product | Unit | Price |
|-------------|------|-------|
| Large Pizza | 2 | 2600 |

Total 2860

Save

```

ver(),
ew();
cation();

rderController(model, view, vf);
ratorModel();
torLogin();
w operatorController(oM, oV);

1Order()f
  
```

10. Communication Log

First Iteration:

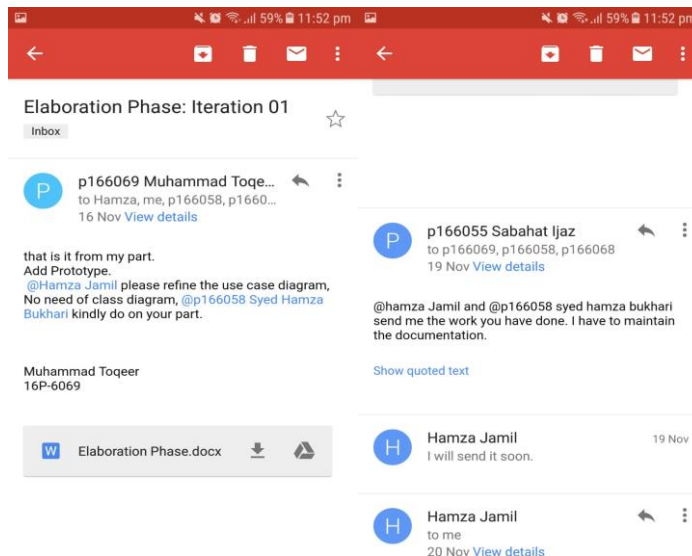
Date: 1st November, 2018

Members attended: Muhammad Touqeer, Sabahat Ijaz, Laiba, Noormah, Hamza Jamil, Hamza Bukhari, Mudassir Athar, Ammar Khan and Yousaf Khan

Agenda discussed: Elaboration phase.

Agreed points: An email group should be made for group discussion. Everyone was agree with their work.

Proof:



Second Iteration:

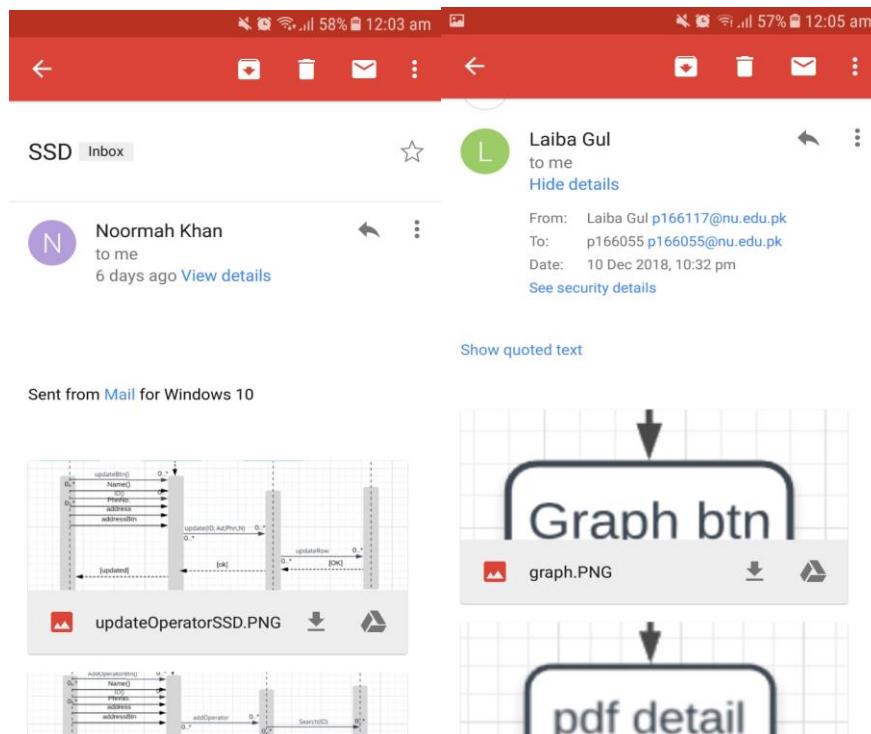
Date: 8th November, 2018

Members attended: Muhammad Touqeer, Sabahat Ijaz, Laiba, Noormah, Hamza Jamil, Hamza Bukhari, Mudassir Athar, Ammar Khan and Yousaf Khan

Agenda discussed: Risks were Highlighted. And Fully dressed Use cases for Basic Requirements were designed.

Agreed points: Everyone was agreed with the solution.

Proof:



Third Iteration:

Date: 19th November, 2018

Members attended: Muhammad Touqeer, Sabahat Ijaz, Laiba, Noormah, Hamza Jamil, Hamza Bukhari, Mudassir Athar, Ammar Khan and Yousaf Khan

Agenda discussed: Prototypes for Project and UML diagrams for Basic Use cases were designed

Agreed points: Everyone was agreed with the solution.

Proof:

| | | | |
|------------|---------|----------------|---------------|
| Noormah | Laiba | Hamza Jamil | Hamza Bukhari |
| Toqeer | Sabahat | Mudassir Athar | Yousaf Khan |
| Ammar Khan | | | |

Forth Iteration:

Date: 29th November, 2018

Members attended: Muhammad Touqeer, Sabahat Ijaz, Laiba, Noormah, Hamza Jamil, Hamza Bukhari, Mudassir Athar, Ammar Khan and Yousaf Khan

Agenda discussed: UML diagrams were designed and Implementation Phase Started.

Agreed points: Everyone was agreed with the solution.

Proof:

Noormah

Laiba

Hamza Jamil

Hamza Bukhari

Toqeer

Sabahat

Mudassir Athar

Yousaf Khan

Ammar Khan
