**TWIRLING FORKS**



Session: 2022 – 2025

**Submitted by:**

AYESHA KHALID 2022-CS-153

**Supervised by:**

DR.AWAIS HASAAN

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

1

Table of Contents

[1. Short Description 3](#_Toc126434090)

[2. Users 3](#_Toc126434091)

[3. Functional Requirements 3](#_Toc126434092)

[4. Wireframes 4](#_Toc126434093)

2

THE TWIRLING FORKS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Short Description

I am trying to sell the noodles of different types so that people can enjoy noodles of best taste .I have used C++ language in this business application. It is important because I as an admin can add different kinds of noodles and can update their stock and price. In this admin can tell the user what service type is and it can add the stock and price of the noodles. And customer can view it. On the other hand customer can add feedback and give tips. While admin can see it. Admin can remove noodles from the stock and can add the discounts. Admin can update the price of noodles and view all the details regarding noodles. Customer can see the menu and can check the details of the noodles. Customer can buy the noodles and can see the bill. It can see the noodles of all the types. It can see the available types of the noodles. And at the end both admin and the customer can LogOut their Ids.

# Users

* Admin
* Customer

# Functional Requirements

***As a I want to perform So that I can***

|  |  |  |
| --- | --- | --- |
| Admin | The task of Adding the noodles | Add the noodles in stock |
| Admin | The task of Deleting the noodles | Delete the noodles which is sold |
| Admin | The task of viewing the feedback | Check the feedbacks from the customers |
| Admin | The task of Updating price | Update the price of the noodles. |
| Admin | The task of Adding discounts | Add the discounts to increase the sale of noodles. |
| Admin | The task of viewing feedback | To view the feedback from customer |
| Admin | The task of Viewing the tips from customers | View the Tips |
| Admin | The task of viewing the noodles details | Check the details of the noodles |
| Admin | The task of giving the services | gives the services to customer |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ayesha Khalid, 2022-CS-153.

3

THE TWIRLING FORKS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***As a I want to perform So that I can***

|  |  |  |
| --- | --- | --- |
| Customer | The task of viewing the service type | See the services of the noodles shop |
| Customer | The task of lbuying products | Buy the products |
| Customer | The task of looking Available types of noodles | Select the available type |
| Customer | The task of looking for the menu | select my favorite noodles |
| Customer | The task of looking for the details | View thedetails |
| Customer | The task of looking for the types of the noodles | View different types of noodles to place my order |
| Customer | The task of looking for the available types | Check the availale types |
| Customer | The task of giving the feedback | Give the Review on the noodles |
| Customer | The task of taking the bill | View the bill |
| Customer | The task of giving the tip | Give the tip |

# Wireframes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Your role:

1: Admin login

2: Customer login

**Figure 1: Log In Screen**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ayesha Khalid, 2022-CS-153. 4

THE TWIRLING FORKS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER ENTERS OPTION 1:

Username = admin

Password = admin123

**Figure 1.1: LogIn Screen for admin**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER ENTERS OPTION 2:

Username = customer

Password = customer123

**Figure 1.2: Log In Screen for customer**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Add noodles in stock.
2. Delete noodles from stock.
3. Update price
4. Add discounts
5. View Noodles details
6. View tips
7. View Feedback
8. Give Services
9. logout

**Figure 2: Admin Main Menu Screen**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ayesha Khalid, 2022-CS-153. 5

THE TWIRLING FORKS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 1.

ADD NOODLES IN STOCK.

New noodle type name = ramen noodles.

Price = 45$

**Figure 2.1: adding noodles in stock Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 2.

DELETE NOODLES FROM STOCK.

Noodles type name = ramen noodles.

Command = Delete.

**Figure 2.2: delete noodles from stock Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 3.

UPDATE PRICE OF NOODLES.

Noodle type name = ramen noodles.

Price = 45$

New price = 50$

**Figure 2.3: updating price of Screen**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ayesha Khalid, 2022-CS-153. 6

THE TWIRLING FORKS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 4.

ADD DICSOUNTS.

Noodle type name = ramen noodles.

Price = 45$

Discount = 10%

**Figure 2.4: adding discount Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 5.

VIEW NOODLES DETAILS

Noodle type name : Price : Stock

EggNoodles : 200 : 10000

**Figure 2.5: view noodles details Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 6

MAKE BILLS

Noodle type name = ramen noodles.

Price of ramen noodles= 45$ and 20% discount

Discount for ramen noodles = 20%

Noodle type name = egg noodles.

Price of egg noodles = 50$ and 10% discount

Bill = (Price of ramen noodles - Discount for ramen) + ( Price of egg noodles - Discount for egg) **Figure 2.6: making bills Screen**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ayesha Khalid, 2022-CS-153. 7

THE TWIRLING FORKS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 7.

FEEDBACK

Customer 1= (feedback from the first customer)

Customer 2 = (feedback from the second customer)

**Figure 2.7: feedback Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 8.

VIEW TIPS

Tip = 1000

**Figure 2.8: checking Tips Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 9.

GIVE SERVICES TYPES

Service type : Indoor service

Service type : Outdoor service

**Figure 2.9: service type Screen**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ayesha Khalid, 2022-CS-153. 8

THE TWIRLING FORKS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 10.

LOGOUT

**Figure 2.10: logout Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Service type
2. Menu
3. Check the Details
4. Buy products
5. Types of all the noodles
6. Available types.
7. Give feedback
8. Bills
9. Tip
10. Logout

**Figure 3: customer Main Menu Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 1.

SERVICE TYPE.

1. Indoor service
2. Outdoor Service

**Figure 3.1: service type Screen**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ayesha Khalid, 2022-CS-153. 9

THE TWIRLING FORKS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 2.

MENU.

1. Egg noodles
2. Ramen noodles
3. Ho fun noodles

**Figure3.2: menu Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 3.

CHECK THE DETAILS

Noodle type name : Price : Stock

EggNoodles : 1000 : 2000

**Figure 3.3: checking details Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 4.

BUY PRODUCTS.

Enter Noodle type name = ramen noodles.

Price = 45

Quantity = 3

Bill = Quantity \* Price – discount;

**Figure 3.4: discounts Screen**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ayesha Khalid, 2022-CS-153. 10

THE TWIRLING FORKS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 5.

TYPE of noodles.

1. Egg noodles
2. Ramen noodles
3. Ho fun noodles
4. Rice stick noodles

**Figure 3.5: types of noodles Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 6.

AVAILABLE TYPES.

1. Egg noodles
2. Ramen noodles
3. Ho fun noodles
4. Rice stick noodles **Figure 3.6: available type Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 7

FEEDBACK

Type our feedback here= \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Figure 3.7: feedback Screen**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ayesha Khalid, 2022-CS-153. 11

THE TWIRLING FORKS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 8

BILLS: Noodle type name = ramen noodles.

Price of ramen noodles= 45

Quantity = 3

Bill = Quantity\*price- discount;

**Figure 3.8: bill Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 9

Tip

Enter the tip= 100 **Figure 3.9: tip Screen**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The twirling forks\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF USER PRESS OPTION 10.

logout

**Figure 3.10: logout Screen**

**Data Structures (Parallel Arrays)**

* + **Variables**
  + Int userArrSize = 10;
  + int size = 50;
  + int x = 0;
  + int noodleCounter = 0;
  + int customerCounter = 0;

THE TWIRLING FORKS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + int sign;
  + int temp;
  + int total;
  + float discount;
  + bool flag = false;
  + bool flagg = true;
  + int usersCount = 0;
  + string name;
  + string password;
  + string role;
  + string option;
  + string message;
  + int newPrice;
  + string day;
  + string record;
  + int price;
  + float bill = 0;
  + float discount1;
  + string confirm;
  + int noodleQuantity;
  + int idx = 0;
  + string type;
  + int commaCount = 1;
  + string item;
  + int length = Password.length();
  + int index = 0;
  + **Arrays**
  + string users[userArrSize];
  + string passwords[userArrSize];
  + string roles[userArrSize];
  + string noodletypesNames[size];
  + int numberOfCustomers[100];
  + int noodlePrice[size];
  + int discountOnNoodles[size];

THE TWIRLING FORKS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + int stockOfNoodles[size];
  + string serviceType[2];
  + int noodlesDeal[size];
  + int tip[size];
  + string feedback[size];
  + int noodlesQuantity[size];
  + bool discountt[size];
  + bool noodless[size];
* **Function Prototypes**
  + void storeData();
  + void storeUsersData();
  + void loadData();
  + string parseRecord(string record, int field);
  + void loadUsersData();
  + void addNoodlesinStock(); //admin can add noodles
  + void removeNoodles(); //admin can remove noodles from stock
  + void updatePrice(); //admin can udate the price of noodles
  + void addDiscount(); //admin can give discount
  + void viewNoodleDetails(); //admin can view the details
  + void viewFeedBack(); //admin can view the feedback of customer
  + void viewTips(); //admin can check the tip given by customer
  + void servicesTypes(); //through which a admin can give the services to customer
  + void viewServicesTypes(); //customer can view services type
  + void viewMenu(); //customer can see menu
  + void viewDetails(); //customer can check the detals
  + void buyProducts(); //customer can buy the noodles
  + void viewNoodlesTypes(); //customer can see all the typesof noodles
  + void viewAvailableTypes(); //customer can see the available types
  + void giveFeedback (); //customer can give feedback
  + void billing(); //customer can view the bills
  + void giveTip(); //customer can give tips
  + void topHeader();
  + void title();
  + void adminsubMenu();
  + string MainMenu();
  + void adminInterface();
  + void clearScreen();
  + int customerMainMenu();
  + void customerInterface();
  + int adminMainMenu();

THE TWIRLING FORKS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + void billing();
  + void subMenu(string submenu);
  + void subMenuBeforeMainMenu(string submenu);
  + bool signUp(string name, string password, string role);
  + string signIn(string name, string password);
  + bool checkPassword(string Password);
  + bool checkName(string name);
* **Functions Working Flow**

LOGIN SCREEN

Exit

signUp()

signIn()

As Customer

As Admin

viewServicesTypes();

removeNoodles();

addNoodlesinStock();

viewMenu();

addDiscount();

updatePrice();

buyProducts();

viewDetails();

viewFeedBack();

viewNoodleDetails();

viewNoodlesTypes();

viewAvailableTypes();

viewTips();

billing();

giveFeedback ();

servicesTypes();

LogOut

LogOut

giveTip();

* **Complete Code of the Business Application**
* #include <fstream>
* #include <iostream>
* #include <conio.h>
* #include <windows.h>
* #include <string>
* using namespace std;
* const int userArrSize = 10;
* const int size = 50;
* int x = 0;
* string users[userArrSize]; // its the ussers array which will count the ussers
* string passwords[userArrSize]; // this the array which will store the passwords of the users
* string roles[userArrSize]; // this is the array which will store the roles
* string noodletypesNames[size] = {"EggNoodles", "RamenNoodles", "HoFunNoodles", "RiceStickNoodles", "TokorotenNoodles"}; // this array is made to store all the types of the noodles
* int numberOfCustomers[100]; // this is the array which will count the number of the customers
* int noodlePrice[size]; // this is the array which will store the price of the noodles
* int discountOnNoodles[size]; // this is the array which will store the discounts on the noodles
* int stockOfNoodles[size]; // this is the array which will store the stock of the noodles
* int oldPrice[size]; // this is the array which will store the old price of the noodles
* string serviceType[2]; // array in which admin can add the services types for the customer
* int noodlesDeal[size]; // this is the array which will check the deals of the noodles
* int tip[size]; // this is the array which will store the tips being added by the customer
* string feedback[size]; // this is the array which will store the feedback of the customers
* int noodlesQuantity[size]; // this is the array which will store the quantity of the noodles added by the customer to buy
* bool discountt[size];
* bool noodless[size];
* void storeData(); // this is the function made to store the data of products in file
* void storeUsersData(); // this is the function made to store the data of users in file
* string parseRecord(string record, int field); // this is the function which is used to count the commas in an array and will return us the desired item
* void loadData(); // this is the function made to load the data of products in file
* void loadUsersData(); // this is the function made to load the data of users in file
* int noodleCounter = 0; // this is the counter for the noodles
* int customerCounter = 0; // this is  the counter for the customers
* int sign;
* int temp;
* int total;
* float discount;
* bool flag = false;
* bool flagg = true;
* int usersCount = 0;
* void addNoodlesinStock(); // it is the first option of the admin which can add noodles in the stock
* void removeNoodles(); // it is the second option of the admin which can remove the quantity of the noodles
* void updatePrice(); // its the third option of the admin which can update the price of the noodles
* void addDiscount(); // its is the function of forth option of the admin through which it can add the discounts on the noodles
* void viewNoodleDetails(); // its the option number 5 of the admin through which it can see the details of the noodles
* void viewFeedBack(); // its the option seven of  the admin through which it can view the feedback of the customers
* void viewTips(); // its the 8th option of the admin through which it can check the tips which are given by the customer
* void servicesTypes(); // its the ninth option of the admin through which it can calculate the monthly profits
* void viewServiceTypes(); // it is the first option of the customer from which customer can see the types of the services
* void viewMenu(); // it is the second option of the customer from which customer can see the menu
* void viewDetails(); // its the third option of the customer from which customer can checkk the details
* void buyProducts(); // its is the function of forth option of the customer from which customer can buy the noodles
* void viewNoodlesTypes(); // its the option number 5 of the customer from which customer can check the types of the noodles
* void viewAvailableTypes(); // its the option number 6 of the customer from which customer can check the type available
* void giveFeedback(); // its the option seven of  the  customer from which customer can give the feedback
* void billing();      // customer can see the bills
* void giveTips(); // its the ninth option of the customer from which customer can give the tips
* void topHeader(); // this function will show the top header of the noodles shop
* void title(); // this function will show the title of the noodle shop which is THE TWIRLING FORKS
* void adminsubMenu(); // this function will show the sub menu of the admin
* string MainMenu(); // this function will show the sign in and sign up screen
* void adminInterface(); // this function will show the admin interface
* void clearScreen(); // this function is made to be used at the end of every functionality to clear up the screen
* int customerMainMenu(); // this function will show the main menu of the customer
* void customerInterface(); // this function will show the customer interface
* int adminMainMenu(); // this function will show the main menu of teh admin
* void billing();      // through which custoner can see its bills
* void subMenu(string submenu); // this function will print the customer or admin "Main Menu" on the console screen
* void subMenuBeforeMainMenu(string submenu); // this function will print the customer or admin "Menu" on the console screen
* bool signUp(string name, string password, string role); // thsi function is made to sign up the customer and the admin
* string signIn(string name, string password); // thsi function is made to sign in the customer and the admin
* bool checkPassword(string Password);
* bool checkName(string name);
* main()
* {
* system("cls"); // system("cls") is used to clear everything from the screen
* loadUsersData();
* loadData();
* getch();
* //  its the option which will be pressed by the user for signin and signup purpose
* string loginOption = "0";
* // this loop is used to ask the user whether he wanna sign in or signup in continuous manner
* while (true)
* {
* title();
* topHeader();
* getch();
* subMenuBeforeMainMenu("Login");
* loginOption = MainMenu();
* // if the user enters 1 then the user will be asked his role and then he will be signed up accordingly
* if (loginOption == "1")
* {
* system("cls");
* string name;
* string password;
* string role;
* topHeader();
* subMenu("SignUp");
* cout << "Enter your Name: " << endl;
* cin >> name;
* bool nameCheck = checkName(name);
* if (nameCheck == true)
* {
* cout << "Enter your Password: " << endl;
* cin >> password;
* bool passwordCheck = checkPassword(password);
* if (passwordCheck == true)
* {
* cout << "Enter your Role (admin or customer): " << endl;
* cin >> role;
* bool isValid = signUp(name, password, role);
* if (isValid)
* {
* cout << "SignedUp Successfully" << endl;
* }
* if (!isValid)
* {
* cout << "Users in the System have exceeded the capacity" << endl;
* }
* }
* }
* else
* {
* cout << "You dont meet the required criteria..try again.." << endl;
* }
* }
* clearScreen();
* if (loginOption == "2")
* {
* system("cls");
* string name;
* string password;
* string role;
* topHeader();
* subMenu("SignIn");
* cout << "Enter your Name: " << endl;
* cin >> name;
* cout << "Enter your Password: " << endl;
* cin >> password;
* role = signIn(name, password);
* if (role == "admin")
* {
* clearScreen();
* adminInterface();
* }
* else if (role == "customer")
* {
* clearScreen();
* topHeader();
* customerInterface();
* }
* else if (role == "undefined")
* {
* cout << " Oops! You have entered wrong details!" << endl;
* clearScreen();
* }
* }
* if (loginOption == "3")
* {
* storeUsersData();
* }
* }
* }
* void topHeader()
* {
* cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;
* cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*THE TWIRLING FORKS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;
* cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;
* }
* void title()
* {
* cout << " \_\_\_\_\_    \_\_\_\_\_    \_\_\_\_\_                 " << endl;
* cout << "|\_   \_|  |\_   \_|  | \_\_\_|    || || ||     " << endl;
* cout << "  | |      | |    | |\_      ||\_|| ||    " << endl;
* cout << " /| |/    /| |/  /|  \_|/    |\_\_  \_\_|    " << endl;
* cout << "  |\_|      |\_|    |\_|          ||       " << endl;
* cout << " \_/ \\\_    \_/ \\\_  \_/ \\\_         ||       " << endl;
* }
* string MainMenu()
* {
* string option;
* cout << "1. SignUp with your Credentials" << endl;
* cout << "2. SignIn to get your Credentials" << endl;
* cout << "3. Exit the Application" << endl;
* cout << endl;
* cout << "Enter the Option Number > ";
* cin >> option;
* return option;
* }
* void clearScreen()
* {
* cout << "Press Any Key to Continue.." << endl;
* getch();
* system("cls");
* }
* void subMenu(string submenu)
* {
* string message = submenu + "Main Menu";
* cout << message << endl;
* }
* void subMenuBeforeMainMenu(string submenu)
* {
* string message = submenu + " Menu";
* cout << message << endl;
* }
* string signIn(string name, string password)
* {
* for (int index = 0; index < usersCount; index++)
* {
* if (users[index] == name && passwords[index] == password)
* {
* return roles[index];
* }
* }
* return "undefined";
* }
* bool signUp(string name, string password, string role)
* {
* if (usersCount < userArrSize)
* {
* users[usersCount] = name;
* passwords[usersCount] = password;
* roles[usersCount] = role;
* usersCount++;
* return true;
* }
* else
* {
* return false;
* }
* }
* void adminInterface()
* {
* int choose = 0;
* while (choose != 10)
* {
* choose = adminMainMenu();
* if (choose == 1)
* {
* system("cls");
* topHeader();
* addNoodlesinStock();
* }
* if (choose == 2)
* {
* system("cls");
* topHeader();
* removeNoodles();
* }
* if (choose == 3)
* {
* system("cls");
* topHeader();
* updatePrice();
* }
* if (choose == 4)
* {
* system("cls");
* topHeader();
* cout << "ADD DISCOUNTS:" << endl;
* addDiscount();
* }
* if (choose == 5)
* {
* system("cls");
* topHeader();
* for (int idx = 0; idx < noodleCounter; idx++)
* {
* cout << "Noodle type name "
* << " \t\t: "
* << "Noodle price "
* << " : "
* << "Stock of noodles" << endl;
* cout << noodletypesNames[idx] << "\t\t:" << noodlePrice[idx] << "\t\t:" << stockOfNoodles[idx] << endl;
* }
* storeData();
* clearScreen();
* }
* if (choose == 7)
* {
* system("cls");
* topHeader();
* for (int i = 0; i < 2; i++)
* {
* cout << feedback[i] << endl;
* }
* clearScreen();
* }
* if (choose == 8)
* {
* system("cls");
* topHeader();
* for (int idx = 0; idx < 1; idx++)
* {
* cout << tip[idx] << endl;
* }
* }
* if (choose == 9)
* {
* system("cls");
* topHeader();
* servicesTypes();
* }
* clearScreen();
* }
* }
* void customerInterface()
* {
* int choose = 0;
* while (choose != 10)
* {
* choose = customerMainMenu();
* if (choose == 1)
* {
* system("cls");
* topHeader();
* viewServiceTypes();
* }
* if (choose == 2)
* {
* system("cls");
* topHeader();
* viewMenu();
* }
* if (choose == 3)
* {
* system("cls");
* topHeader();
* viewDetails();
* }
* if (choose == 4)
* {
* system("cls");
* topHeader();
* buyProducts();
* }
* if (choose == 5)
* {
* system("cls");
* topHeader();
* viewNoodlesTypes();
* }
* if (choose == 6)
* {
* cout << "Noodles type name"
* << "\t"
* << "Noodles Price"
* << "\t\t"
* << " Stock of noodles" << endl;
* for (int idx = 0; idx < noodleCounter; idx++)
* {
* cout << "\t\t\t" << noodletypesNames[idx] << "\t\t: " << noodlePrice[idx] << "\t\t: " << stockOfNoodles[idx] << endl;
* }
* }
* if (choose == 7)
* {
* system("cls");
* topHeader();
* cout << "Feedback:" << endl;
* for (int i = 0; i < 2; i++)
* {
* getline(cin, feedback[i]);
* }
* }
* if (choose == 8)
* {
* system("cls");
* topHeader();
* billing();
* }
* if (choose == 9)
* {
* system("cls");
* topHeader();
* for (int i = 0; i < 1; i++)
* {
* cout << "enter the amount you want to give as a tip= ";
* cin >> tip[i];
* }
* }
* clearScreen();
* }
* }
* int adminMainMenu()
* {
* int option;
* topHeader();
* cout << "1. Add noodles in stock" << endl;
* cout << "2. Remove noodles from the stock" << endl;
* cout << "3. Update price" << endl;
* cout << "4. Add discount" << endl;
* cout << "5. View noodles details" << endl;
* cout << "7. Feedback" << endl;
* cout << "8. Check the tips" << endl;
* cout << "9. Can give service type" << endl;
* cout << "10. LogOut" << endl;
* cout << "Your Option[1 to 10]:";
* cin >> option;
* return option;
* topHeader();
* clearScreen();
* }
* void servicesTypes()
* {
* cout << "Enter the service types for the customer= ";
* for (int i = 0; i < 2; i++)
* {
* getline(cin, serviceType[i]);
* }
* }
* int customerMainMenu()
* {
* int option,
* topHeader();
* cout << "1.Service type" << endl;
* cout << "2.Menu" << endl;
* cout << "3.Check The Details" << endl;
* cout << "4.Buy Noodles" << endl;
* cout << "5.Types of all the noodles" << endl;
* cout << "6.Available types" << endl;
* cout << "7.Feedback" << endl;
* cout << "8.Bills" << endl;
* cout << "9.Can give Tips" << endl;
* cout << "10.LogOut" << endl;
* cout << "Your Option..";
* cin >> option;
* return option;
* }
* void addNoodlesinStock()
* {
* system("cls");
* topHeader();
* string name;
* cout << "Noodles types: " << endl;
* for (int i = 0; i < 5; i++)
* {
* cout << noodletypesNames[i] << endl;
* }
* int count = 0;
* cout << "Enter the type of the noodle: ";
* cin >> name;
* for (int i = 0; i < noodleCounter; i++)
* {
* if (name == noodletypesNames[i])
* {
* count++;
* }
* }
* if (count != 0)
* {
* cout << "Enter the price of this noodle= ";
* cin >> noodlePrice[noodleCounter];
* cout << "Enter the stock of this noodle: ";
* cin >> stockOfNoodles[noodleCounter];
* noodletypesNames[noodleCounter] = name;
* noodleCounter++;
* }
* else
* {
* cout << "WOOPS! YOU HAVE ENTERED WRONG NAME.TRY AGAIN :(" << endl;
* }
* clearScreen();
* }
* void updatePrice()
* {
* system("cls");
* topHeader();
* string name;
* int newPrice;
* cout << "UPDATE PRICE OF NOODLES:" << endl;
* for (int i = 0; i < 5; i++)
* {
* cout << noodletypesNames[i] << endl;
* }
* cout << "Noodles type name= ";
* cin >> name;
* cout << "New price= ";
* cin >> newPrice;
* noodleCounter++;
* if (name == noodletypesNames[noodleCounter])
* {
* noodleCounter++;
* }
* if (newPrice == noodlePrice[noodleCounter])
* {
* noodleCounter++;
* }
* clearScreen();
* }
* void addDiscount()
* {
* string day;
* cout << "Noodles types: " << endl;
* for (int i = 0; i < 5; i++)
* {
* cout << noodletypesNames[i] << endl;
* }
* cout << "Noodles type name= ";
* cin >> noodletypesNames[noodleCounter];
* cout << "Enter the day= ";
* cin >> day;
* cout << "Enter the discount= ";
* cin >> discount;
* if (day == "saturday" && day == "Saturday" || day == "sunday" && day == "Sunday")
* {
* cout << "Todays discount is= " << discount << endl;
* }
* }
* void viewNoodleDetails()
* {
* cout << "View noodles details:";
* adminOption4();
* adminOption2();
* cin >> noodletypesNames[noodleCounter];
* cin >> noodlePrice[noodleCounter];
* cin >> stockOfNoodles[noodleCounter];
* }
* void removeNoodles()
* {
* system("cls");
* topHeader();
* string name;
* cout << "Noodles types: " << endl;
* for (int i = 0; i < 5; i++)
* {
* cout << noodletypesNames[i] << endl;
* }
* cout << "REMOVE THE NOODLES QUANTITY:" << endl;
* cout << "Noodles type name you want to remove= ";
* cin >> name;
* for (int i = 0; i < noodleCounter; i++)
* {
* if (name == noodletypesNames[i])
* {
* cout << "stock= ";
* cin >> stockOfNoodles[i];
* cout << "enter the quantity you want to delete= ";
* cin >> noodlesQuantity[i];
* if (noodlesQuantity[i] < stockOfNoodles[i])
* {
* stockOfNoodles[i] = stockOfNoodles[i] - noodlesQuantity[i];
* cout << "Stock left= " << stockOfNoodles[i] << endl;
* }
* else
* {
* cout << "This much stock is not available :(";
* }
* }
* else
* {
* cout << "This stock is not available yet";
* }
* }
* clearScreen();
* }
* void viewServiceTypes()
* {
* cout << "Service type: " << endl;
* for (int i = 0; i < 2; i++)
* {
* cout << serviceType[i] << endl;
* }
* }
* void viewMenu()
* {
* cout << "MENU:" << endl;
* for (int i = 0; i < 5; i++)
* {
* cout << noodletypesNames[i] << endl;
* }
* }
* void viewDetails()
* {
* cout << "Noodles type name"
* << "\t"
* << "Noodles Price"
* << "\t\t"
* << " Stock of noodles" << endl;
* for (int idx = 0; idx < noodleCounter; idx++)
* {
* cout << "\t\t\t" << noodletypesNames[idx] << "\t\t: " << noodlePrice[idx] << "\t\t: " << stockOfNoodles[idx] << endl;
* }
* }
* void customerOption4()
* {
* string day;
* cout << "Enterthe day= ";
* cin >> day;
* if (day == "saturday" && day == "sunday")
* {
* discount = 0.30;
* discountOnNoodles[noodleCounter] = discount;
* cout << "Todays discount is= " << discount << endl;
* }
* else
* {
* cout << "No discount today :(";
* }
* }
* void viewNoodlesTypes()
* {
* cout << "Types of all the noodles:" << endl;
* for (int i = 0; i < 5; i++)
* {
* cout << noodletypesNames[i] << endl;
* }
* }
* void viewAvailableTypes()
* {
* system("cls");
* topHeader();
* for (int i = 0; i < 5; i++)
* {
* cout << noodletypesNames[i] << endl;
* }
* }
* void storeData()
* {
* fstream file;
* file.open("noodlesayesha.txt", ios::out);
* int x = 0;
* while (x < noodleCounter)
* {
* file << noodletypesNames[x] << ",";
* file << feedback[x] << ",";
* file << numberOfCustomers[x] << ",";
* file << noodlePrice[x] << ",";
* file << discountOnNoodles[x] << ",";
* file << stockOfNoodles[x] << ",";
* file << oldPrice[x] << ",";
* file << noodlesDeal[x] << ",";
* file << tip[x] << ",";
* file << noodlesQuantity[x] << endl;
* x++;
* }
* file.close();
* }
* void loadData()
* {
* string record;
* fstream file;
* file.open("noodlesayesha.txt", ios::in);
* while (getline(file, record))
* {
* noodletypesNames[noodleCounter] = parseRecord(record, 1);
* feedback[noodleCounter] = parseRecord(record, 2);
* numberOfCustomers[noodleCounter] = stoi(parseRecord(record, 3));
* noodlePrice[noodleCounter] = stoi(parseRecord(record, 4));
* discountOnNoodles[noodleCounter] = stoi(parseRecord(record, 5));
* stockOfNoodles[noodleCounter] = stoi(parseRecord(record, 6));
* oldPrice[noodleCounter] = stoi(parseRecord(record, 7));
* noodlesDeal[noodleCounter] = stoi(parseRecord(record, 8));
* tip[noodleCounter] = stoi(parseRecord(record, 9));
* noodlesQuantity[noodleCounter] = stoi(parseRecord(record, 10));
* noodleCounter++;
* }
* file.close();
* }
* void storeUsersData()
* {
* fstream myfile;
* myfile.open("userayesha.txt", ios::out);
* int x = 0;
* while (x < noodleCounter)
* {
* myfile << users[x] << ",";
* myfile << passwords[x] << ",";
* myfile << roles[x] << endl;
* x++;
* }
* myfile.close();
* }
* void loadUsersData()
* {
* string record;
* fstream myfile;
* myfile.open("userayesha.txt", ios::in);
* while (getline(myfile, record))
* {
* users[noodleCounter] = parseRecord(record, 1);
* passwords[noodleCounter] = parseRecord(record, 2);
* roles[noodleCounter] = parseRecord(record, 3);
* noodleCounter++;
* }
* myfile.close();
* }
* void billing()
* {
* cout << "Noodle type name"
* << ":"
* << " Quantity of noodles"
* << ":"
* << " Price" << endl;
* for (int i = 0; i < noodleCounter; i++)
* {
* cout << noodletypesNames[i] << "\t\t: " << noodlesQuantity[i] << "\t\t: " << noodlePrice[i] << endl;
* }
* int price;
* float bill = 0;
* string day;
* float discount1;
* cout << "Enter the day= ";
* cin >> day;
* getch();
* for (int idx = 0; idx < noodleCounter; idx++)
* {
* if ((day == "Sunday" || day == "Saturday") && discountt[idx] == true && noodless[idx] == true)
* {
* discount1 = noodlesQuantity[idx] \* noodlePrice[idx] \* 0.30;
* price = noodlePrice[idx] \* noodlesQuantity[idx];
* bill = price - discount1;
* cout << "Total bill of " << noodletypesNames[idx] << " is=  " << bill << endl;
* discountt[idx] = false;
* noodless[idx] = false;
* total = total + bill;
* }
* }
* cout << "Your total bill is : " << total << endl;
* }
* void buyProducts()
* {
* int idx = 0;
* string confirm;
* int noodleQuantity;
* string type;
* while (idx != 5)
* {
* cout << "Noodles Type"
* << "\t:\t"
* << " Price"
* << "\t:\t"
* << " Stock" << endl;
* for (int idx = 0; idx < noodleCounter; idx++)
* {
* cout << noodletypesNames[idx] << "\t\t: " << noodlePrice[idx] << "\t\t: " << stockOfNoodles[idx] << endl;
* }
* cout << "Wanna buy noodles?(yes/no)=  ";
* cin >> confirm;
* if (confirm == "yes")
* {
* cout << "Enter the noodles type= ";
* cin >> type;
* cout << "Enter the quantity of the noodles= ";
* cin >> noodleQuantity;
* for (int i = 0; i < 5; i++)
* {
* if (type == noodletypesNames[i])
* {
* noodlesQuantity[i] = noodleQuantity;
* stockOfNoodles[i] = stockOfNoodles[i] - noodlesQuantity[i];
* discountt[i] = true;
* noodless[i] = true;
* }
* else
* {
* cout << "No match for this noodle type :(";
* }
* cout << noodletypesNames[i] << " : " << noodlesQuantity[i] << endl;
* }
* }
* else if (confirm == "no")
* {
* break;
* }
* else
* {
* cout << "Woops! Seems like you didn't meet the required criteria!Try again :(" << endl;
* }
* clearScreen();
* idx++;
* }
* }
* string parseRecord(string record, int field)
* {
* int commaCount = 1;
* string item;
* for (int x = 0; x < record.length(); x++)
* {
* if (record[x] == ',')
* {
* commaCount = commaCount + 1;
* }
* else if (commaCount == field)
* {
* item = item + record[x];
* }
* }
* return item;
* }
* bool checkPassword(string Password)
* {
* bool flag = false;
* int length = Password.length();
* int index = 0;
* if (length > 5)
* {
* for (int idx = 0; idx < usersCount; idx++)
* {
* if (Password == passwords[idx])
* {
* return flag;
* }
* }
* while (index < length)
* {
* if ((Password[index] > 63 && Password[index] < 91) || (Password[index] > 96 && Password[index] < 123) || (Password[index] > 47 && Password[index] < 58))
* {
* index++;
* }
* else
* {
* return flag;
* }
* }
* flag = true;
* }
* return flag;
* }
* bool checkName(string name)
* {
* bool flag = false;
* int length = name.length();
* int index = 0;
* if (length > 5)
* {
* for (int idx = 0; idx < usersCount; idx++)
* {
* if (name == users[idx])
* {
* return flag;
* }
* }
* while (index < length)
* {
* if ((name[index] > 63 && name[index] < 91) || (name[index] > 96 && name[index] < 123))
* {
* index++;
* }
* else
* {
* return flag;
* }
* }
* flag = true;
* }
* return flag;
* }
* while (flagg)
* {
* for (int i = 0; i < name.length(); i++)
* {
* if (!((name[i] >= 97 && name[i] <= 122) || (name[i] >= 65 && name[i] <= 90)))
* {
* cout << "Invalid!! Enter Again" << endl;
* cin >> name;
* break;
* }
* else
* {
* flagg = false;
* }
* }
* }
* **Weakness in the Business Application**
  + My weakness is I coulnt add all the validations. also I am having problem with filehandling. I could make only few int/float or string type functions.Mostly all were void.
* **Future Directions**
  + I will add all the validations and I will do the file handling correctly. And I will try to avoid Void kinda functions.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ayesha Khalid, 2022-CS-153. 12