UMM Distribution Company

Project Supervisor

Mr.Samyan Qayyum Wahla

Project Members(CS261F22FinalPID33)

Mutaiba Mohsin	2021-CS-63
Uswa Arif	2021-CS-77
Mahnoor Hassan	2021-CS-86

Department of Computer Science
University of Engineering and Technology, Lahore
Pakistan

Contents

1 Project Description 5
2 Project Features 6
3 System Requirements 6
4 Actors and Stakeholders 7
5 Use Cases 7
6 Interfaces 30
7 Data Structure: 70
8 Exceptions: 71
9 Data Storage: 71
10 Email Sending: 73
11 Object Oriented Features: 73
12 Inheritance: 77
13 Polymorphism: 77
14 Detailed Object Oriented Design: 80
15 OOP Status Table 80
16 Analytical Reports: 81
17 New Project Plan 83

List of Tables

1 System Requirements . . . 6
2 U01 . . . 7
3 U02 . . . 8
4 U03 . . . 9
5 U04 . . . 9
6 U05 . . . 10
7 U06 . . . 11
8 U07 . . . 12
9 U08 . . . 13
10 U09 . . . 13
11 U10 . . . 14
12 U11 . . . 14
13 U12 . . . 15
14 U13 . . . 16
15 U14 . . . 16
16 U15 . . . 17
17 U16 . . . 17
18 U17 . . . 18
19 U18 . . . 18
20 U19 . . . 19
21 U20 . . . 19
22 U21 . . . 20
23 U22 . . . 21
24 U23 . . . 22
25 U24 . . . 22
26 U25 . . . 23
27 U26 . . . 23
28 U27 . . . 24
29 U28 . . . 24
30 U29 . . . 25
31 U30 . . . 25
32 U31 . . . 25
33 U32 . . . 26
34 U33 . . . 26
35 U34 . . . 27
36 U35 . . . 27
37 U36 . . . 28
38 U37 . . . 28
39 Status Table . . . 29
40 I01 . . . 30
41 I02 . . . 31
42 I03 . . . 32
43 I04 . . . 33
44 I05 . . . 34
45 I06 . . . 35
46 I07 . . . 36
47 I08 . . . 37
48 I09 . . . 38
49 I10 . . . 39
50 I11 . . . 40

51 I12 . . . 41
52 I13 . . . 42
53 I14 . . . 43
54 I15 . . . 44
55 I16 . . . 45
56 I17 . . . 46
57 I18 . . . 47
58 I19 . . . 48
59 I20 . . . 49
60 I21 . . . 50
61 I22 . . . 51
62 I23 . . . 52
63 I24 . . . 53
64 I25 . . . 54
65 I26 . . . 55
66 I27 . . . 56
67 I28 . . . 57
68 I29 . . . 58
69 I30 . . . 59
70 I31 . . . 60
71 I32 . . . 61
72 I33 . . . 61
73 I34 . . . 62
74 I35 . . . 63
75 I36 . . . 64
76 I37 . . . 65
77 I38 . . . 66
78 I39 . . . 67
79 I40 . . . 68
80 I41 . . . 69
81 Data Structures . . . 70
82 Exceptions . . . 71
83 OOP Status Table . . . 80
84 Project Plan . . . 83

1 Project Description

Since the internet usage worldwide has made 90% of the world population to depend on it. Its increasing usage day by day has made people rely on the internet to look for products and services available through internet. This reliance was a great opportunity for the companies, who has maintained its reputation in the real world, to step in this digital world.

Keeping these in mind, we have chosen to work on massively popular product, "Tea", available even in your nearby stores. Our project allows the manager, employees, and riders to perform the function that not helps in management of the company but also helps them to perform a wide range of operations to facilitate our customers. The Head office of the company may be located in Lahore, Pakistan. The head office get it's supplied from the factory located at Karachi, Pakistan. The product manufactured in the factory then goes to the head office from where it is distributed to retail stores all over Pakistan.

The manufacturing of products with different sizes is done. The company works like all typical companies resulting in oriented working environment, the rider gets the area or areas assigned by employee, who is authorized and responsible for assigning the locations, and it's his responsibility to get an order from the customer and deliver it within the specific duration. The riders' location is being tracked until the order is delivered, this is done by the tracking Id assigned to every rider uniquely, and the rider can see the location of the customer and the customer can enter the Id and see where he is.

The rider can collect the cash from the customer and maintain the ledger of the customer respectively. If the payment is paid for the past purchases the rider can update the history, the payment goes to the company. The rider can inform the employee that the order is delivered that can be confirmed from its location. Hence the project can also help the rider to pull up customer delivery information, update the delivery status and update the payment history.

The rider is responsible for Order getting and delivering likewise the company is responsible for the maintenance of the vehicle used by the individual rider, and also the fuel consumed during the duty hours. The employee is in charged to look after these expenses, and these expenses can be written off as Business Expenses. The employee can increase the stock as there is increase in the orders. The stock is removed as soon as the expiry date is reached. As in winters, the demand of the product increases so the production hyped up during these specific months, in this case company has more profit. The salary of the employees are managed by the manager as manager has access to all over the company and the employee manages the salary of the rider.

All these employees related tasks are easily managed using the features available in our projects. The Head work is to work in the benefit of the company this electronic order booking helps to manage the data easily and with the proof that product has been delivered. By making the improvement in the system, entering data in the system, controlling stock limit, increasing the product production, maintaining the profit and salary of all the people related to the company, all these features make it at least 90% better than the manual working. Also increases the monthly earning by reducing the cost of the labor, giving details of the most sold product and increasing its production.

The extra expenses deducted from the company earning is the cost of the fuel, which can be reduced using our efficient map that gives the shortest distance from one location to another. The data can be sorted providing number of sale highest from which frequenter, allows the company to make strong relations with that retailer and where there are less sales to identify the reason and improve it. Another user is Customer, allowing customer to inform about their orders through emails, also the history of past payments pending and placing the order digitally. The customer can track the order, if the rider is nearby or not.

Providing all these features is not the only task but also making it choice of a favorable data structure making it fast. Linear and Non-Linear Data Structures will be used such as Array, Stack, Linked-List, List, Queue and graph. The usage is depending on the function such as for the function where delivery is done, queue is used, FIFO the one placed before gets the delivery first. The project will good enough to increase the growth and flexibility of the business. It helps you simplify day-to-day tasks, get more customers and deliver more goods. A perfect place to manage the products, orders and payments.

2 Project Features

In Distribution companies, the complete process in which the products successfully travel from company to the wholesalers, retailers and local stores, requires complete management. This includes Employers, Riders and Manager. A software for distribution companies involves a lot of features for the proper flow of these activities. The features include:

- Complete Data Management of Employers, Riders, Products, Customers and Orders. Theses includes Add, Edit, Remove, View and Search.
- Complete Order Management modules to collect ordered data of customers. All orders are maintained by the rider. All the data of the orders such as Order Delivered, Orders which are still left to be delivered.
- Location of riders can be tracked by map, giving the path between two points and their latitude and longitude.
- Riders can see customer's location and Customers have also an access to the map.
- The Software also offers all Customer's data relevant to the orders by creating details of customer's data which includes their placed orders, payment according to the order placed and orders delivered.
- Stock is also maintained by Refilling the products and removing the expired products.
- Email is also send to the customers about their order placed.
- Complete Expenses Management which includes Profit and Loses of the company per month.

3 System Requirements

Table 1: System Requirements

Language	C Sharp
IDEs	Windows Form App (.net Framework) Visual Studio 2019 StarUML JustInMind Latex
Packages	GMap.NET.Core GMap.NET.WinForms Google.Apis Google.Apis.Auth Google.Apis.Core Google.Apis.Gmail.v1 EntityFramework Newtonsoft.Json Stub.System.Data.SQLite.Core.NetFramework System.Data.SqlClient System.Data.SQLite System.Data.SqlClient.Core System.Data.SqLite.EF6 System.Data.SqLite.Linq System.Security.Principal.Windows

Libraries	GMap.NET GMap.NET.MapProviders GMap.NET.WindowsForms GMap.NET.WindowsForms.Markers Google.Apis.Auth.OAuth2 Google.Apis.Gmail.v1 Google.Apis.Services Google.Apis.Util.Store System.IO System.Threading System System.Collections.Generic System.Windows.Forms

4 Actors and Stakeholders

Actors are the entities that directly interact and act within the system.

- **Employees:** The project team or employees are the actors as they are also interacted by the project. They are hired for controlling inventory system that determines the items present in stocks and to manage expenses. They have access on riders.
- **Manager:** The Project manager is the actor that is responsible for implementation and completion of project by the project team. It has access to all over the company and project. They have to keep an eye on every element taking place in the company.
- **Rider:** The riders are the actors of the project which takes and delivers the orders to their respective place. They have access to the location of the customers and they maintain all the data related to the orders and respective customers.

Stakeholders are the entities that does not interact directly to the system.

- **Customers:** Customers can order the products. They also receive an email of their order. Their data is also maintained by the system.

5 Use Cases

Use Case 1(Sign-in Page):

Table 2: U01

Use Case ID	U01
Name	Sign-in page
Actor	Manger, Employee, Rider
Description	The Users have to login to get access the functionality of system. They first have to fill up this page. This is the login page where users have to enter their username and password in order to enter into the system.

Flow	Basic Flow: 1: Users enters the system. 2: They want to login into their accounts. 3: Users enter their username. 4: Users enters the passwords. 5: Then hit the login button. 6: System checks the username and password matches and they are signed into their accounts. Alternative Flow: 1: Users enters the system. 6a: Users enters the invalid username. 1: System shows an error message. 2: User again enters the correct username. 6b: Users enters the invalid password. 1: System shows an error message. 2: User again enters the correct password. 7: System recognizes the data and give login to the user.

Use Case 2(Create Account):

Table 3: U02

Use Case ID	U02
Name	Create account
Actor	Employee, Rider
Description	The new employee or rider have to add in the system. At first, they have to create their accounts. They have to sign up to the system. In order to get signed up, they have to fill the username, password and confirm password options. An account of the employee or rider will be created.
Flow	Basic Flow: 1: Employees/riders arrives at the company. 2: They have to create their accounts at first. 3: Press the create account link button on the login page. 4: A sign up page opens. 5: Employees/riders enters the new username. 6: Employees/riders enters the new password. 7: Employees/riders enters the confirmed password. 8: Press the sign up button. 9: System checks the password and confirm password entered by the employees/riders is same. 10: An account of a person has been created. Alternative Flow: 1: Employees/riders creates an account. 6: System gives error about password is not strong. 1: Employees/riders again enters the password and creates account. 2: System creates a new account for the user and the use case ended. 9: System gives error that password and confirmed password is not same. 1: Employees/riders again entered the password and confirmed password. 2: System creates a new account for the user and the use case ended.

Use Case 3(Reset Password):

Table 4: U03

Use Case ID	U03
Name	Reset Password.
Actor	Employee, Rider
Description	Employees or Riders logins to get access the functionality of the System. Account login gets username and password. If Employees or Riders forget the password, then System will not login the users until password is not entered. There is an option of the reset password. Employees or Riders can again create their account password and get logins to the System.
Flow	Basic Flow: 1: Employees/riders opens the Sign-In page. 2: Employees/riders forgets the password. 3: Employees/riders clicks on the forget-password option and a Forget-Password page is opened. 4: Employees/riders enters the new password. 5: Employees/riders enters the confirmed password same as the new password. 6: Employees/riders clicks on the Reset button. 7: System checks the password and confirmed password. 8: A new password for the employees/riders will be created. Alternative Flow: 1: Employees/riders creates a new password. 4: System gives error about password is not strong. 1: Employees/riders again enters the password and creates new one. 2: System creates a new password for the user and the use case ended. 5: System gives error that password and confirmed password is not same. 1: Employees/riders again entered the password and confirmed password. 2: System creates a new password for the user and the use case ended.

Use Case 4(Add Rider):

Table 5: U04

Use Case ID	U04
Name	Add Rider
Actor	Employee, Manager
Description	Manager/Employee can add a rider into the system where he has to enter his name, CNIC, Id, phone number, salary and vehicle number. A new rider would be added to the system.

Flow	Basic Flow: 1: A new rider arrives at distribution company. 2: Rider is hired and added to the system. 3: Manager/Employee enters the name of rider. 4: Manager/Employee enters the CNIC of rider. 5: Manager/Employee enters the Id given to rider by company. 6: Manager/Employee enters the rider's phone number. 7: Manager/Employee enters the rider's salary. 8: Manager/Employee enters the vehicle number given to rider. 9: Manager/Employee clicks the add button. 10: System checks for the duplicate name and phone number. 11: System checks the phone number an CNIC format. Alternative Flow: 1: Manager/Employee enters the whole data of rider. 10: System detects duplicate name and phone number. 1: It gives the error that rider already exists. 11a: Manager/Employee writes incorrect format of rider's phone number 1: System do not except the format of phone number and shows a message to rewrite the phone number in a correct format. 2: Manager/Employee enters the correct format of rider's phone number. 11b: Manager/Employee writes incorrect format of rider's CNIC. 1: System do not except the format of CNIC and shows a message to rewrite in a correct format. 2: Manager/Employee enters the correct format of rider's CNIC. 12: System founds duplicate vehicle number. 1: System shows that vehicle number is already assigned to another rider. 2: Manager/Employee again enters the other vehicle that is not assigned to some other riders.

Use Case 5(Edit Rider):

Table 6: U05

Use Case ID	U05
Name	Edit Rider
Actor	Employee, Manager
Description	Manager/Employee can edit a rider of the system that was previously added to the system. If the rider's name, CNIC, Id, phone number, salary or vehicle number has to be changed then the manager/employee has to use this edit rider option where he can change the rider's information and save this new information to the system.

Flow	Basic Flow: 1: A rider was added to the system previously. 2: Manager/Employee see all rider's information in a grid view table. 3: Manager/employee wants to change a rider's information. 4: Manager/employee presses the edit button against the information of the respective rider. 5: Employee re-enters the information that is needed to be change. 6: Employee hits the edited button. 7: System checks for the duplicate name and phone number with other riders. 8: System checks the phone number and CNIC format. 9: System checks for the duplicate vehicle number. 10: The newly entered information would then be updated against the respective rider into the system. Alternative Flow: 1: Manager/Employee enters the whole edited data of rider. 7: System detects duplicate name and phone number. 1: It gives the error that rider already exists. 8a: Manager/Employee writes incorrect format of rider's phone number. 1: System do not except the format of phone number and shows a message to rewrite the phone number in a correct format. 2: Manager/Employee enters the correct format of rider's phone number. 8b: Manager/Employee writes incorrect format of rider's CNIC. 1: System do not except the format of CNIC and shows a message to rewrite in a correct format. 2: Manager/Employee enters the correct format of rider's CNIC. 9: System founds duplicate vehicle number. 1: System shows that vehicle number is already assigned to another rider. 2: Manager/Employee again enters the other vehicle that is not assigned to some other riders. 10: System will save the edited data. 11: This new edit data is accessed everywhere on System.

Use Case 6(Remove Rider):

Table 7: U06

Use Case ID	U06
Name	Remove Rider
Actor	Employee, Manager
Description	Manager/Employee can remove a rider that was a part of the system. A grid view table is shown in which all rider's information is shown. Manager/Employee removes the rider and it gets deleted from the system.

Flow	Basic Flow: 1: The Employee wants to delete a rider from the system. 2: Employee see all rider's information in a grid view table. 3: Employee presses the remove button. 4: The System requires a confirmation to delete a rider. 5: User confirms to delete rider. 4: The respective rider and his information is deleted from the system. 5: The orders given to that rider is transferred to other rider. Alternative Flow: 1: Manager/Employee deletes the rider's data and check the confirmation of deletion. 2: System do not delete the rider's data due to slow down of software or other reasons. 3: Manager/Employee again deletes the rider's data. 4: System deletes rider and the use case ends.

Use Case 7(Search Riders):

Table 8: U07

Use Case ID	U07
Name	Search Riders
Actor	Employee/Manager
Description	Manager/Employee wants to search for a rider from the system. Manager/employee searches the rider by Id given to that rider.
Flow	Basic Flow: 1: The Manager/employee wants to search a rider from the system. 2: Manager/employee clicks on the search button present in View Rider page. 3: Manager/employee enters the rider's Id to search. 4: Manager/employee enters the search button. 5: System checks for the rider by Id given by Manager/employee. 6: System shows the searched rider's all information in a grid view table. 7: Manager/employee sees searched rider's information in a grid view table and use case ends. Alternative Flow: 3: Manager/employee enters invalid Id of rider to search. 4: System checks for rider. 1: System gives message that the Id entered is not present in system. 2: Manager/employee again enters the Id of rider to search.

Use Case 8(View Riders):

Table 9: U08

Use Case ID	U08
Name	View Riders
Actor	Employee, Manager
Description	Manager/Employee views all rider from the system. A grid view table is shown in which all rider's information is shown. The table shows rider's name, CNIC, Id, salary, phone number and vehicle number.
Flow	Basic Flow: 1: The Manager/employee wants to view all rider from the system. 2: Manager/employee clicks on the View riders button that is available in manager/employee's menu. 3: Manager/employee see all rider's information in a grid view table.

Use Case 9(Add Customer):

Table 10: U09

Use Case ID	U09
Name	Add Customer
Actor	Rider
Description	Rider can add a customer into the system where he has to enter his name, CNIC, email, ledger, frequenter, total orders and address. A new customer would be added to the system.
Flow	Basic Flow: 1: A new rider arrives at distribution company. 2: Customer is hired and added to the system. 3: Rider enters the name of customer. 4: Rider enters the CNIC of customer. 5: Rider enters the ledger of customer. 6: Rider enters the customer is frequenter or not. 7: Rider enters the rider's email. 8: Rider enters the total orders of customer. 9: Rider enters the rider's address. 10: Rider clicks the add button. 11: System checks for the duplicate CNIC and email. 12: System checks the CNIC and email format. Alternative Flow: 1: Rider enters the whole data of customer. 11: System detects duplicate CNIC and email. 1: It gives the error that customer already exists. 12a: Rider writes incorrect format of customer's CNIC. 1: System do not except the format of CNIC and shows a message to rewrite the CNIC in a correct format. 2: Rider enters the correct format of customer's CNIC. 12b: Rider writes incorrect format of customer's email. 1: System do not except the format of email and shows a message to rewrite the email in a correct format. 2: Rider enters the correct format of customer's email.

Use Case 10(Edit Customer):

Table 11: U10

Use Case ID	U10
Name	Edit Customer
Actor	Rider
Description	Rider can edit a customer of the system that was previously added to the system. If the customer's name, CNIC, email, ledger, frequenter, total orders and address has to be changed then the Rider has to use this edit customer option where he can change the customer's information and save this new information to the system.
Flow	Basic Flow: 1: A customer was added to the system previously. 2: Rider see all customer's information in a grid view table. 3: Rider wants to change a customer's information. 4: Rider presses the edit button against the information of the respective customer. 5: Rider re-enters the information that is needed to be change. 6: Rider hits the edited button. 7: System checks for the duplicate CNIC and email with other customers. 8: System checks the CNIC and email format. 9: The newly entered information would then be updated against the respective rider into the system. Alternative Flow: 1: Rider enters the whole edited data of customer. 7: System detects duplicate CNIC and email with other customers. 1: It gives the error that customer already exists. 8a: Rider writes incorrect format of customer's CNIC. 1: System do not except the format of CNIC and shows a message to rewrite the CNIC in a correct format. 2: Rider enters the correct format of customer's CNIC. 8b: Rider writes incorrect format of customer's email. 1: System do not except the format of email and shows a message to rewrite the email in a correct format. 2: Rider enters the correct format of customer's email. 9: System will save the edited data. 10: This new edit data is accessed everywhere on System.

Use Case 11(Remove Customer):

Table 12: U11

Use Case ID	U11
Name	Remove Customer
Actor	Rider
Description	Rider can remove a customer that was a part of the system. A grid view table is shown in which all customer's information is shown. Rider removes the customer and it gets deleted from the system.

Flow	Basic Flow: 1: The Rider wants to delete a customer from the system. 2: Rider see all customer's information in a grid view table. 3: Rider presses the remove button. 4: The System requires a confirmation to delete a customer. 5: Rider confirms to delete customer. 4: The respective customer and his information is deleted from the system. 5: System also deletes all orders of that customer also. Alternative Flow: 1: Rider deletes the customer's data and check the confirmation of deletion. 2: System do not delete the customer's data due to slow down of software or other reasons. 3: Rider again deletes the customer's data. 4: System deletes customer and the use case ends.

Use Case 12(Search Customer):

Table 13: U12

Use Case ID	U12
Name	Search Customer
Actor	Rider
Description	Rider wants to search for a customer from the system. Rider searches the customer by CNIC of that customer.
Flow	Basic Flow: 1: The Rider wants to search a customer from the system. 2: Rider clicks on the search button present in View Customer page. 3: Rider enters the customer's CNIC to search. 4: Rider enters the search button. 5: System checks for the customer by CNIC. 6: System shows the searched customer's all information in a grid view table. 7: Rider sees searched customer's information in a grid view table and use case ends. Alternative Flow: 3: Rider enters incorrect CNIC of customer to search. 4: System checks for that customer. 1: System gives message that the CNIC entered is not present in system. 2: Rider again enters the CNIC of customer to search.

Use Case 13(View Customers):

Table 14: U13

Use Case ID	U13
Name	View Customers
Actor	Employee, Manager, Rider
Description	Manager/Employee/Rider views all customer from the system. A grid view table is shown in which all customer's information is shown. The table shows customer's name, CNIC, email, ledger, frequenter, total orders and address.
Flow	Basic Flow: 1: The Manager/employee/rider wants to view all customer from the system. 2: Manager/employee/rider clicks on the View Customers button that is available in manager/employee/rider's menu. 3: Manager/employee/rider see all customer's information in a grid view table.

Use Case 14(Add Order):

Table 15: U14

Use Case ID	U14
Name	Add Order
Actor	Rider
Description	Rider is able to see a table in which all the orders placed were shown. If the rider wants to enter a new order, he has to press the add button and add order page opens where he enters the Order Id, product, month and date of order placed.
Flow	Basic Flow: 1: A grid view table of all the orders placed is shown to the rider. 2: Rider wants to add a new order. 3: Rider presses the add button. 4: Rider enters Order Id of customer. 5: Rider enters product name that the customer wants to order. 6: Rider enters month in which order is placed. 7: Rider enters date to which order will delivers. 8: Rider presses the added button. 9: System checks for the product is present in stock or not. 10: System checks the date entered is correct or not. 11: System adds the order of customer. Alternative Flow: 1: Rider enters the whole data of order. 9: System found product is not available in stock. 1: System gives message that product is not available. 2: Rider do not place the order. 10: System founds incorrect date. 1: System gives message that the date entered is incorrect. 2: Rider again enters the correct date. 11: System adds the order of customer.

Use Case 15(Edit Order):

Table 16: U15

Use Case ID	U15
Name	Edit Order
Actor	Rider
Description	Rider is shown a table in which all the orders placed were shown. If the rider wants to edit an order placed before, he has to press the edit button, the edit order page opens where he re-enters the Order Id, product, month and date of order placed which is to be changed.
Flow	Basic Flow: 1: A grid view table of all the orders placed is shown to the rider. 2: Rider wants to edit an order. 3: Rider presses the edit button present in View Orders page. 4: An Edit Order page is opened. 5: Rider makes changes to the order placed. 6: Rider hits the edited button. 7: System checks for the product is present in stock or not. 8: System checks the date entered is correct or not. 9: System edits the order of customer. Alternative Flow: 7: System found product is not available in stock. 1: System gives message that product is not available. 2: Rider do not place the order. 8: System founds incorrect date. 1: System gives message that the date entered is incorrect. 2: Rider again enters the correct date. 9: System will save the edited data. 10: This new edit data is accessed everywhere on System.

Use Case 16(Remove Order):

Table 17: U16

Use Case ID	U16
Name	Remove Order
Actor	Rider
Description	Rider is shown a table in which all the orders placed were shown. If the Rider wants to delete an order, he has to press the remove button. The order placed against that remove button will be deleted from the order list.

Flow	Basic Flow: 1: A grid view table of all the orders placed is shown to the rider. 2: Rider wants to delete an order. 3: Rider presses the remove button. 4: The order against this remove button will be deleted from the order list. Alternative Flow: 1: Rider deletes the order's data. 2: System check the confirmation of deletion. 3: System do not delete the order's data due to slow down of software or other reasons. 4: Rider again deletes the order's data. 5: System deletes it and the use case ends.

Use Case 17(Search Order):

Table 18: U17

Use Case ID	U17
Name	Search Order
Actor	Rider
Description	Rider wants to search for order from the system. Rider searches the order by Id given to that Order.
Flow	Basic Flow: 1: The Rider wants to search Order of a customer from the system. 2: Rider clicks on the search button present in View Orders page. 3: Rider enters the Order Id to search. 4: Rider enters the search button. 5: System checks for that Order Id. 6: System shows the searched Order's all information in a grid view table. 7: Rider sees searched Order's information in a grid view table and use case ends. Alternative Flow: 3: Rider enters invalid Order Id to search. 4: System checks for that Id. 1: System gives message that the Id entered is not present in system. 2: Rider again enters the Order Id to search.

Use Case 18(View Order):

Table 19: U18

Use Case ID	U18
Name	View Order
Actor	Employee, Manager, Rider
Description	Manager/Employee/Rider views all orders from the system. A grid view table is shown in which all order's information is shown.
Flow	Basic Flow: 1: The Manager/employee/Rider wants to view all Order from the system. 2: Manager/employee/Rider clicks on the View Order button that is available in manager/employee/rider's menu. 3: Manager/employee/rider see all Order's information in a grid view table.

Use Case 19(Add Employee):

Table 20: U19

Use Case ID	U19
Name	Add Employee
Actor	Manager
Description	Manager can add employee into the system where he has to enter his name, CNIC, phone number, salary and Id. A new employee would be added to the system.
Flow	Basic Flow: 1: A new employee arrives at distribution company. 2: Employee is hired and added to the system. 3: The manager enters the employee's name. 4: The manager enters the employee's CNIC. 5: The manager enters the employee's phone number. 6: The manager enters the employee's salary. 7: The manager enters the employee's Id. 8: The manager hits the add button. 9: System checks for the duplicate name and phone number. 10: System checks the phone number and CNIC format. 11: System checks for the duplicate Id. Alternative Flow: 1: Manager enters the whole data of employee. 9: System detects duplicate name and phone number. 1: It gives the error that rider already exists. 10a: Manager writes incorrect format of employee's phone number. 1: System do not except the format of phone number and shows a message to rewrite the phone number in a correct format. 2: Manager enters the correct format of employee's phone number. 10b: Manager writes incorrect format of employee's CNIC. 1: System do not except the format of CNIC and shows a message to rewrite the CNIC in a correct format. 2: Manager enters the correct format of employee's CNIC. 11: System founds duplicate Id. 1: System shows that Id is already assigned to another employee. 2: Manage again enters the Id that is not assigned to some other employees.

Use Case 20(Edit Employee):

Table 21: U20

Use Case ID	U20
Name	Edit Employee
Actor	Manager
Description	Manager can edit employee of the system that was previously added to the system. If the employee's name, CNIC, phone number, salary or Id has to be changed then the manager has to use this edit employee option where he can change the employee's information and saves this new information to the system.

Flow	Basic Flow: 1: Employee was added to the system previously. 2: Manager see all employee's information in a grid view table. 3: Manage wants to change employee's information. 4: Manager presses the edit button against the information of the respective employee. 5: Manager re-enters the information that is needed to be change. 6: Manager hits the edited button. 7: System checks for the duplicate name and phone number with other employees. 8: System checks the phone number and CNIC format. 9: System checks for the duplicate Id. 10: The newly entered information would then be updated against the respective employee into the system. Alternative Flow: 1: Manager enters the whole edited data of employee. 7: System detects duplicate name and phone number. 1: It gives the error that employee already exists. 8a: Manager writes incorrect format of employee's phone number. 1: System do not except the format of phone number and shows a message to rewrite the phone number in a correct format. 2: Manager again enters the correct format of employee's phone number. 8b: Manager writes incorrect format of employee's CNIC. 1: System do not except the format of CNIC and shows a message to rewrite the CNIC in a correct format. 2: Manager again enters the correct format of employee's CNIC. 9: System founds duplicate Id. 1: System shows that Id is already assigned to another employee. 2: Manager again enters the other vehicle that is not assigned to some other employee. 10: System will save the edited data. 11: This new edit data is accessed everywhere on System.

Use Case 21(Remove Employee):

Table 22: U21

Use Case ID	U21
Name	Remove Employee
Actor	Manager
Description	Manager can remove employee from the system that was a part of the system. In order to remove employee from the system, manager presses the remove button and the employee was deleted from the system.

Flow	Basic Flow: 1: The manager wants to delete employee from the system. 2: Manager sees all rider's information in a grid view table. 3: Manager presses the remove button. 4: The System requires a confirmation to delete employee. 5: User confirms to delete employee. 6: The respective employee and his information is deleted from the system. Alternative Flow: 1: Manager deletes the employee's data and check the confirmation of deletion. 2: System do not delete the employee's data due to slow down of software or other reasons. 3: Manager again deletes the employee's data by pressing on delete button. 4: System deletes it and the use case ends.

Use Case 22(Search Employee):

Table 23: U22

Use Case ID	U22
Name	Search Employee
Actor	Manager
Description	Manager wants to search for employee from the system. Manager searches the employee by Id given to that employee.
Flow	Basic Flow: 1: The Manager wants to search employee from the system. 2: Manager clicks on the search button present in View Employee page. 3: Manager enters the employee's Id to search. 4: Manager enters the search button. 5: System checks for the employee by Id given by Manager. 6: System shows the searched employee's all information in a grid view table. 7: Manager sees searched employee's information in a grid view table and use case ends. Alternative Flow: 3: Manager enters invalid Id of employee to search. 4: System checks for employee. 1: System gives message that the Id entered is not present in system. 2: Manager again enters the Id of employee to search.

Use Case 23(View Employees):

Table 24: U23

Use Case ID	U23
Name	View Employees
Actor	Manager
Description	Manage views all employees from the system. A grid view table is shown in which all employee's information is shown.
Flow	Basic Flow: 1: The Manager wants to view all employees from the system. 2: Manager clicks on the View employees button that is available in manager's menu. 3: Manager see all employee's information in a grid view table.

Use Case 24(Add Product):

Table 25: U24

Use Case ID	U24
Name	Add Product
Actor	Manager
Description	Manager can add products into the system where he has to enter name, stock, threshold, price, total product sold, expiry date and size of product. Then press the add button. A new product would be added to the system.
Flow	Basic Flow: 1: A new product added at distribution company. 2: The manager enters the product's name. 3: The manager enters the total stock of product. 4: The manager enters the product's threshold price. 5: The manager enters the product's original price. 6: The manager enters the total sold products. 7: The manager enters the expiry date of that product. 8: The manager enters the product's size. 9: The manager hits the add button. 10: System checks for the duplicate size of the product. 11: System checks for the expiry date of product. Alternative Flow: 1: Manager enters the whole data of product. 10: System detects duplicate size of product. 1: It gives the message that product already exists. 11: Manager writes incorrect format of expiry date of product. 1: System do not except the format of expiry date and shows a message to rewrite the expiry date in a correct format. 2: Manager enters the correct format of product's expiry date.

Use Case 25(Edit Product):

Table 26: U25

Use Case ID	U25
Name	Edit Product
Actor	Manager
Description	Manager can edit a product of the system that was previously added to the system. If the name, stock, threshold, price, total product sold, expiry date and size of product has to be changed then the manager has to use this edit product option where he can change the product's information and saves this new information to the system.
Flow	Basic Flow: 1: Product was added to the system previously. 2: Manager see all product's information in a grid view table. 3: Manage wants to change product's information. 4: Manager presses the edit button against the information of the respective product. 5: Manager re-enters the information that is needed to be change. 6: Manager hits the edited button. 7: System checks for the duplicate size of products with other products. 8: System checks the expiry date of product. 9: The newly entered information would then be updated against the respective product into the system. Alternative Flow: 1: Manager enters the whole edited data of product. 7: System detects for duplicate size of product with other products. 1: It gives the error that product already exists. 8: Manager writes incorrect expiry date of product. 1: System do not except the format of expiry date and shows a message to rewrite the expiry date in a correct format. 2: Manager again enters the correct format of expiry date of product. 9: System will save the edited data. 10: This new edit data is accessed everywhere on System.

Use Case 26(Remove Product):

Table 27: U26

Use Case ID	U26
Name	Remove Product
Actor	Manager
Description	Manager can remove a product from the system that was a part of the system. In order to remove a product from the system, manager presses the remove button and the product was deleted from the system.

Flow	Basic Flow: 1: The Manager wants to delete a product from the system. 2: see all product's information in a grid view table. 3: If a product has to be removed from the system then the manager presses the remove button. 4: The respective product and his information is deleted from the system. Alternative Flow: 1: Manager deletes the product's data and check the confirmation of deletion. 2: System do not confirm the deletion of product's data due to slow down of software. 3: Manager again deletes the product's data by pressing on delete button. 4: System deletes it and the use case ends.

Use Case 27(Search Product):

Table 28: U27

Use Case ID	U27
Name	Search Product
Actor	Manager
Description	Manager wants to search for product from the system. Manager searches the product by size of that product.
Flow	Basic Flow: 1: The Manager wants to search product from the system. 2: Manager clicks on the search button present in View product page. 3: Manager enters the product's size to search. 4: Manager enters the search button. 5: System checks for the product by size given by Manager. 6: System shows the searched product all information in a grid view table. 7: Manager sees searched product's information in a grid view table and use case ends. Alternative Flow: 3: Manager enters invalid size of product to search. 4: System checks for product. 1: System gives message that the Id entered is not present in system. 2: Manager again enters the size of product to search.

Use Case 28(View Product):

Table 29: U28

Use Case ID	U28
Name	View Products
Actor	Manager
Description	Manager views all products from the system. A grid view table is shown in which all product's information is shown.
Flow	Basic Flow: 1: The Manager wants to view all products from the system. 2: Manager clicks on the View products button that is available in manager/employee's menu. 3: Manager see all product's information in a grid view table.

Use Case 29(Manage Expenses):

Table 30: U29

Use Case ID	U29
Name	Manage Expenses
Actor	Manager
Description	Manager can record all expenses, profits, loses and bonus occurs at the company.
Flow	Basic Flow: 1: The manager records the expenses of riders such as vehicle given to them, fuels used by vehicle. 2: Manager also records the salary given to riders and employees. 3: Manager record all bonus to riders or employees and all profits, loses occurs at company.

Use Case 30(Give Salary to Riders):

Table 31: U30

Use Case ID	U30
Name	Give Salary to Riders
Actor	Employee
Description	Employee maintain salary status of all riders. If rider delivers more orders to the customer before time, a bonus is also given to the riders. This bonus is also maintained by the employee.
Flow	Basic Flow: 1: Employee gives salary to all riders. 2: Employee opens Riders salary status page. 3: Employee enters the name of rider to which the salary is given. 4: Employee enters rider's Id. 5: Employee clicks on the Bonus status (Clicks on Yes if bonus is given, otherwise No). 6: System calculates the salary of rider with or without bonus.

Use Case 31(Give Salary to Employees):

Table 32: U31

Use Case ID	U31
Name	Give Salary to Employees
Actor	Manager
Description	Manager maintain salary status of all employees. If employee gives salaries to the riders on time and well manage rider's responsibilities, a bonus is also given to employees. This bonus is also maintained by the manager.
Flow	Basic Flow: 1: Manager gives salary to all employees. 2: Manager opens Employees salary status page. 3: Manager enters the name of employee to which the salary is given. 4: Manager enters employee's Id. 5: Manager clicks on the Bonus status (Clicks on Yes if bonus is given, otherwise No). 6: System calculates the salary of employee with or without bonus.

Use Case 32(Send Email):

Table 33: U32

Use Case ID	U32
Name	Send Email
Actor	Manager, Employee
Description	Manager/Employee can send email to the customer for thanking them to purpose stock from their company. They can also send email to customers: if customers any previous record of payment is left, if the payment is clear or if there is any problem in sending late orders to customer.
Flow	Basic Flow: 1: Manager/Employee opens the Email Sending page. 2: Manager/Employee enters the name to send email to the customer. 3: Manager/Employee writes Subject ad body of email for why they are sending email to customer. 4: Manager/Employee enters the send button. 5: System checks the customer email. 6: A message is shown that the email is send and it is delivered to the respective customer. Alternative Flow: 1: Manager/Employee open Email page to send email to customer. 6: Manager/Employee enters invalid or wrong email of customer. 1: System shows the error that email is incorrect as there is no customer email present that manager or employee writes. 2: Manager/Employee enters the correct email of that customer and send email. 3: System verifies the email. 4: A message is shown that the email is sent.

Use Case 33(make graph):

Table 34: U33

Use Case ID	U33
Name	make graph
Actor	Manager
Description	Manager can see the financial report of profits or loss in Graph form where the percentages is given for profits and loss in company expenses management.
Flow	Basic Flow: 1: Manager opens financial report and wants to see analytical report of finances. 2: Manager clicks on the analytical button. 3: System shows a graph where visualized profits or losses are put together. 4: Manager can see all reports of loss and profits.

Use Case 34(Deliver Ordered Products):

Table 35: U34

Use Case ID	U34
Name	Deliver Ordered Products
Actor	Rider
Description	Riders have completely visible data of orders. Riders have total number of ordered products that delivers to the customers. Orders are delivered on exact dates given with them. If ordered products are delivered, a check box is ticked by the rider.
Flow	Basic Flow: 1: Rider's work is to deliver all the ordered products of customers. 2: System presents all ordered products done by the customers. 3: Rider checkout all orders. 4: Rider starts from the top most orders of customer and delivers to that customer. 5: Customer receives the orders. 6: Rider tick on the check box to assure that the product is delivered. 7: System shows that the ordered product is delivered to the customer. Alternative Flow: 1: If the order is not delivered, product again added to the orders list of all customers.

Use Case 35(Show location of Customer to deliver products):

Table 36: U35

Use Case ID	U35
Name	Show location of Customer to deliver products
Actor	Rider
Description	Rider watch the location of customer by map. Rider delivers the ordered product to customer.
Flow	Basic Flow: 1: Rider starts a new sale. 2: System shows the location of customer to the rider. 3: Rider watch the location by map and delivers the ordered products to customer. Alternative Flow: 1: The location entered is wrong. 2: System updated the location.

Use Case 36(update payments of Customer):

Table 37: U36

Use Case ID	U36
Name	update payments of Customer
Actor	Rider
Description	Rider updates the payment by just tick on check box.
Flow	Basic Flow: 1: Rider delivers the order. 2: Customer wants to pay the payment of Order. 3: Rider opens the Customer Payment page. 4: Rider enters the Customer's Order Id. 5: Rider enters the email of Customer. 6: Rider Enters the amount paid by customer and clicks on the check box amount paid. 7: System checks for the correct Order Id and email of customer. 8: System records the amount paid. Alternative Flow: 7: System found incorrect Order Id or email. 1: System gives message of incorrect Id or email. 2: Rider again enters the above incorrect option.

Use Case 37(Designing UI):

Table 38: U37

Use Case ID	U37
Name	Designing UI
Actor	Employee, Rider
Description	The form overall designing will be done which display all the available options.
Flow	Basic flow: 1: The user logins into the account and the menu is displayed. 2: If User is employee, the option given to that user is displayed. 3: If User is ride, the option given to rider is displayed. 4: If User is manager, the option given to that user is displayed.

Status Table

Table 39: Status Table

Use Case ID	Assigned To	Status	% implemented	Comments
U01	Mahnoor Hassan	implemented	100	
U02	Mahnoor Hassan	Implemented	100	
U03	Mahnoor Hassan	Implemented	100	
U04	Uswa Arif	Implemented	100	
U05	Uswa Arif	Implemented	100	
U06	Uswa Arif	Implemented	100	
U07	Uswa Arif	Implemented	100	
U08	Uswa Arif	Implemented	100	
U09	Mutaiba Mohsin	Implemented	100	
U10	Mutaiba Mohsin	Implemented	100	
U11	Mutaiba Mohsin	Implemented	100	
U12	Mutaiba Mohsin	Implemented	100	
U13	Mutaiba Mohsin	Implemented	100	
U14	Mutaiba Mohsin	Implemented	100	
U15	Mutaiba Mohsin	Implemented	100	
U16	Mutaiba Mohsin	Implemented	100	
U17	Mutaiba Mohsin	Implemented	100	
U18	Mutaiba Mohsin	Implemented	100	
U19	Mahnoor Hassan	Implemented	100	
U20	Mahnoor Hassan	Implemented	100	
U21	Mahnoor Hassan	Implemented	100	
U22	Mahnoor Hassan	Implemented	100	
U23	Mahnoor Hassan	Implemented	100	
U24	Mutaiba Mohsin	Implemented	100	
U25	Mutaiba Mohsin	Implemented	100	
U26	Mutaiba Mohsin	Implemented	100	
U27	Mutaiba Mohsin	Implemented	100	
U28	Mutaiba Mohsin	Implemented	100	
U29	Uswa Arif	Implemented	100	
U30	Uswa Arif	Implemented	100	
U31	Uswa Arif	Implemented	100	
U32	Uswa Arif	Implemented	100	
U33	Uswa Arif	Implemented	90	
U34	Uswa Arif	Implemented	100	
U35	Uswa Arif	Implemented	70	
U36	Mahnoor Hassan	Implemented	100	
U37	Mahnoor Hassan	Implemented	100	

6 Interfaces

Table 40: I01

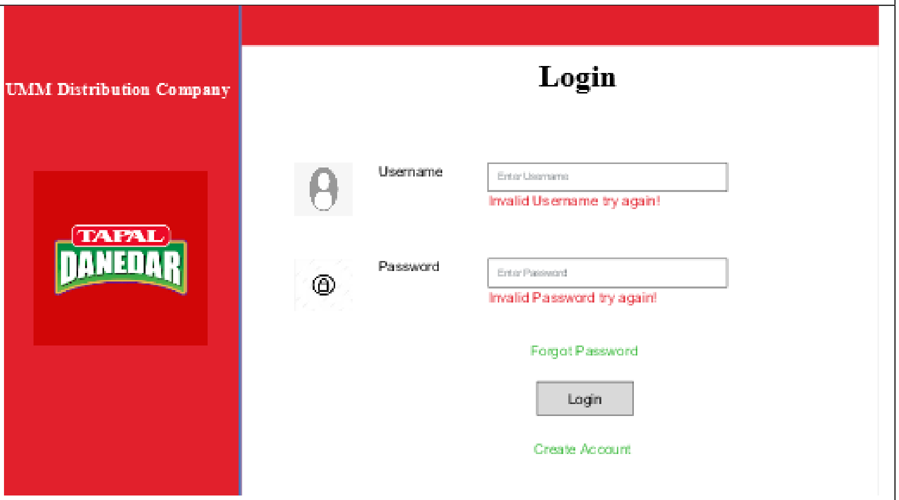
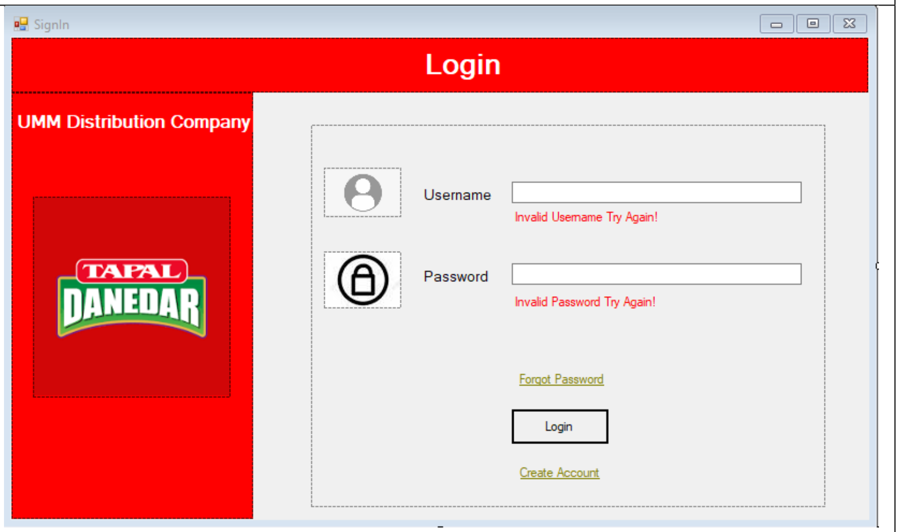
Interface ID	I01
Name	Login
Linked Use Case	U01
UI Screen (JustInMind)	 UMM Distribution Company TAPAL DANEDAR Login Username Enter Username Password Enter Password Forgot Password Login Create Account Invalid Username try again! Invalid Password try again!
UI Screen	 SignIn Login UMM Distribution Company TAPAL DANEDAR Username Password Forgot Password Login Create Account Invalid Username Try Again! Invalid Password Try Again!
Validators	1: Invalid UserName 2: Invalid Password 3: Account not created

Table 41: I02

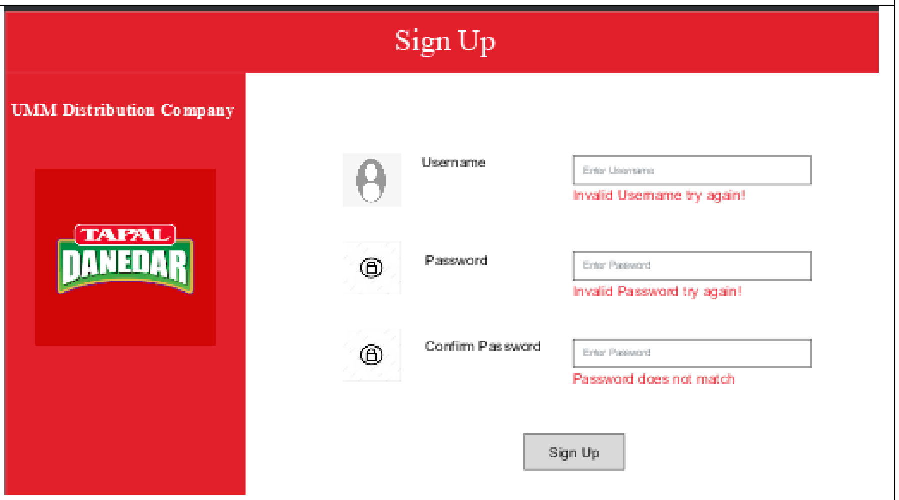
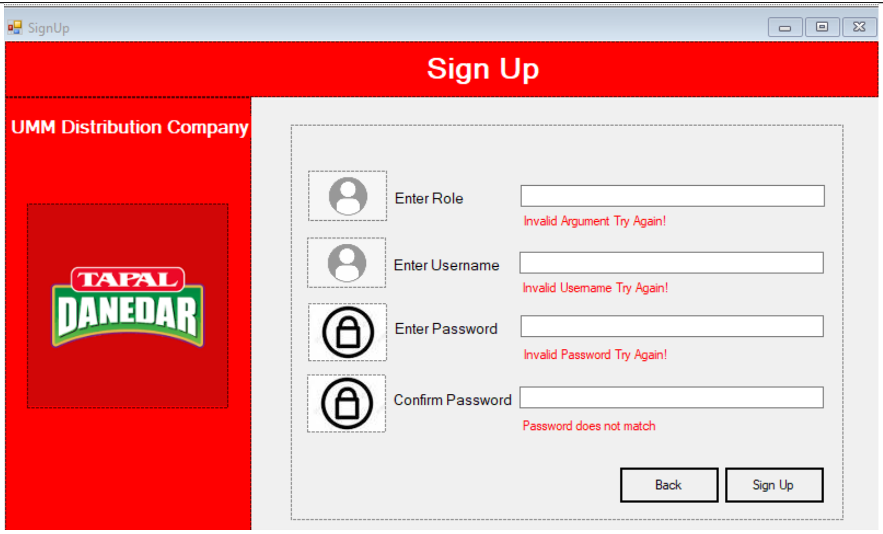
Interface ID	I02
Name	Sign Up
Linked Use Case	U02
UI Screen (JustInMind)	 Sign Up UMM Distribution Company TAPAL DANEDAR Username Enter Username Password Enter Password Confirm Password Enter Password Sign Up Invalid Username try again! Invalid Password try again! Password does not match
UI Screen	 SignUp Sign Up UMM Distribution Company TAPAL DANEDAR Enter Role Enter Username Enter Password Confirm Password Back Sign Up Invalid Argument Try Again! Invalid Username Try Again! Invalid Password Try Again! Password does not match
Validators	1: Invalid UserName 2: Invalid Password 3: Confirm Password condition is wrong

Table 42: I03

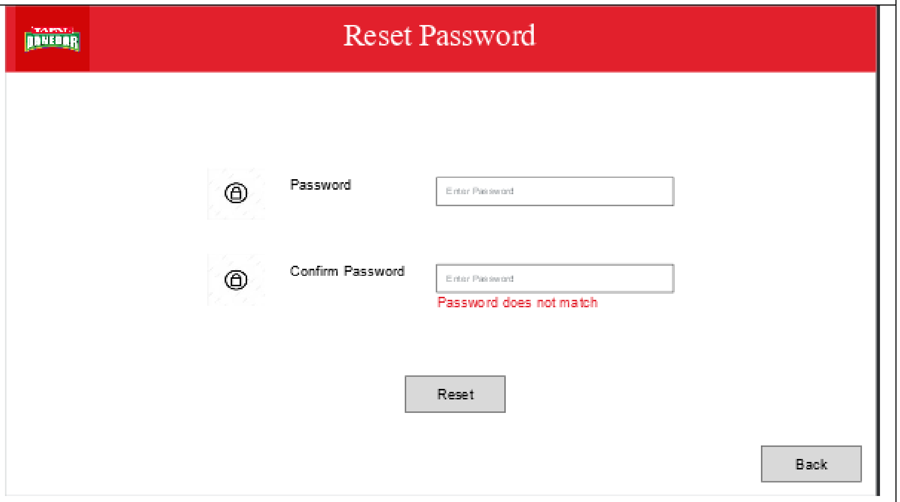
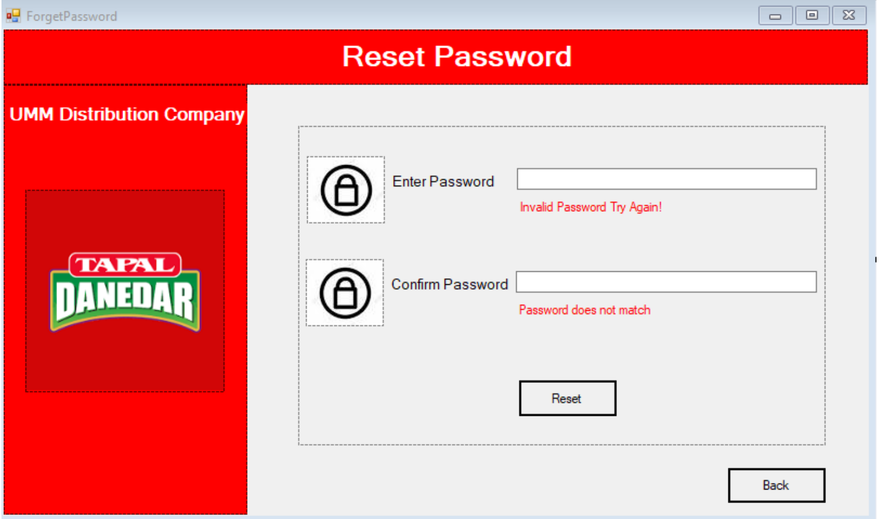
Interface ID	I03
Name	Reset Password
Linked Use Case	U03
UI Screen (JustInMind)	 Reset Password Password Enter Password Confirm Password Enter Password Reset Back Password does not match
UI Screen	 ForgetPassword Reset Password UMM Distribution Company TAPAL DANEDAR Enter Password Confirm Password Reset Back Invalid Password Try Again! Password does not match
Validators	1: Invalid Password 2: Reset Button doesn't work

Table 43: I04

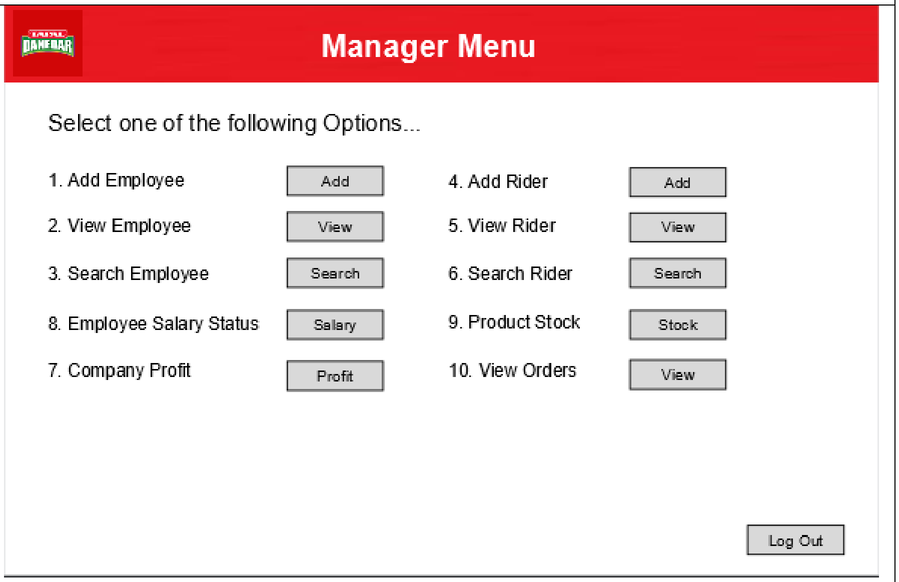
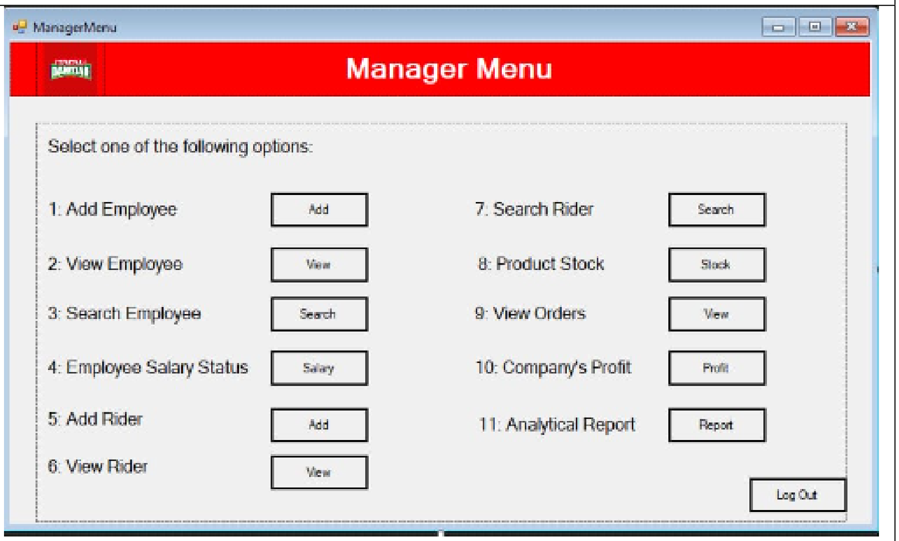
Interface ID	I04
Name	Manager Menu
Linked Use Case	U37
UI Screen (JustInMind)	 Manager Menu Select one of the following Options... 1. Add Employee Add 4. Add Rider Add 2. View Employee View 5. View Rider View 3. Search Employee Search 6. Search Rider Search 8. Employee Salary Status Salary 9. Product Stock Stock 7. Company Profit Profit 10. View Orders View Log Out
UI Screen	 ManagerMenu Manager Menu Select one of the following options: 1: Add Employee Add 7: Search Rider Search 2: View Employee View 8: Product Stock Stock 3: Search Employee Search 9: View Orders View 4: Employee Salary Status Salary 10: Company's Profit Profit 5: Add Rider Add 11: Analytical Report Report 6: View Rider View Log Out
Validators	1: Can't Open the required page.

Table 44: I05

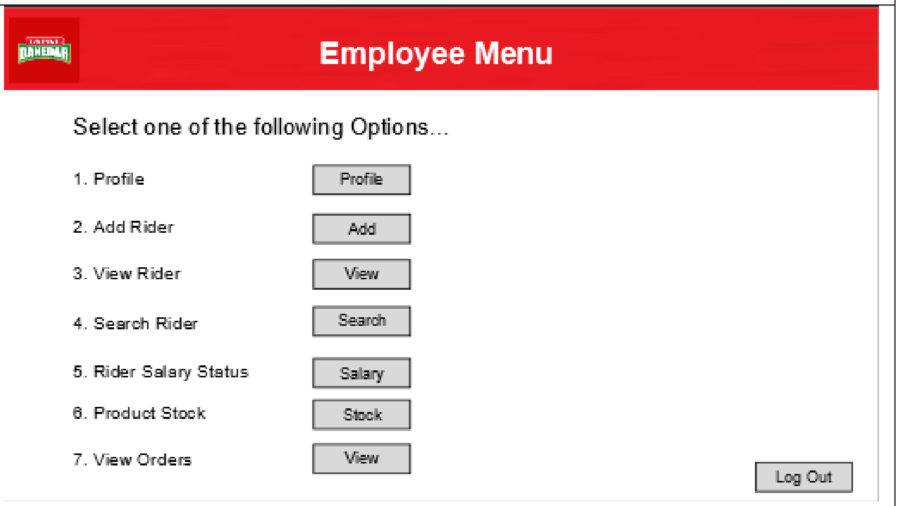
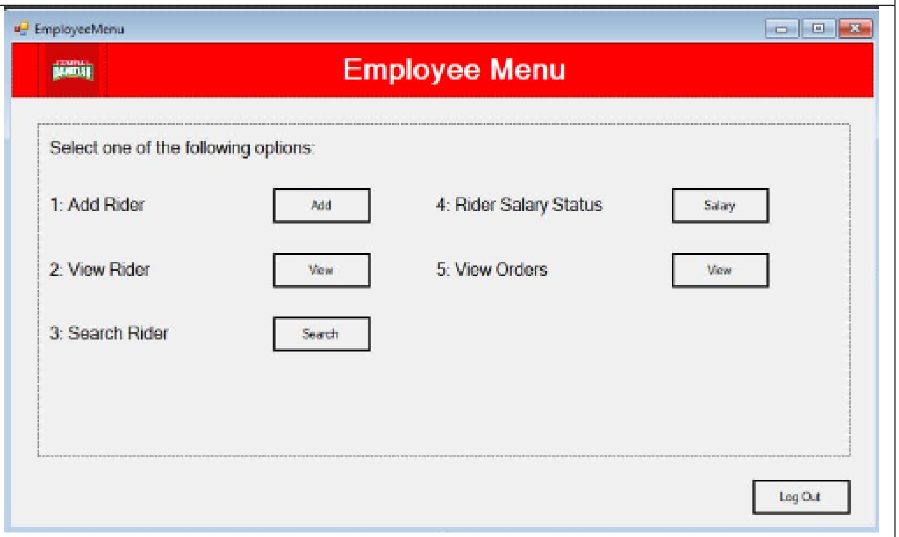
Interface ID	I05
Name	Employee Menu
Linked Use Case	U37
UI Screen (JustInMind)	 Select one of the following Options... Employee Menu 1. Profile Profile 2. Add Rider Add 3. View Rider View 4. Search Rider Search 5. Rider Salary Status Salary 6. Product Stock Stock 7. View Orders View Log Out
UI Screen	 Select one of the following options: EmployeeMenu Employee Menu 1: Add Rider Add 2: View Rider View 3: Search Rider Search 4: Rider Salary Status Salary 5: View Orders View Log Out
Validators	1: Can't Open the required page.

Table 45: I06

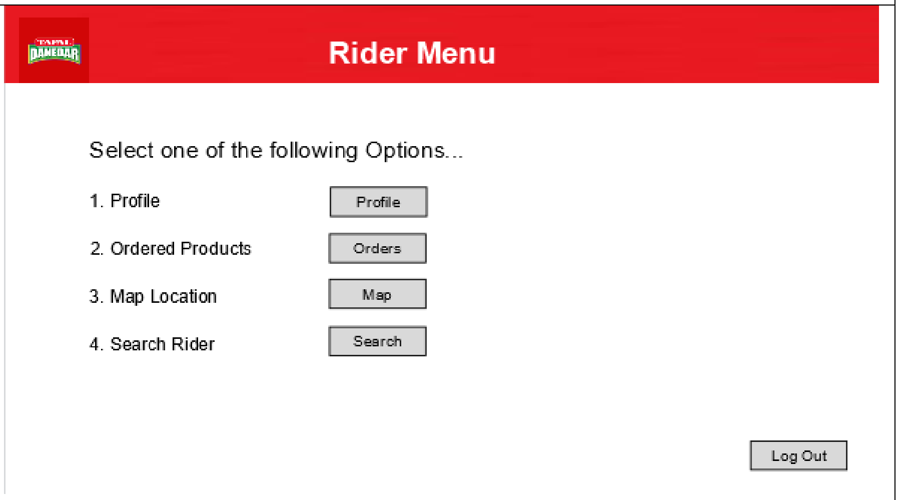
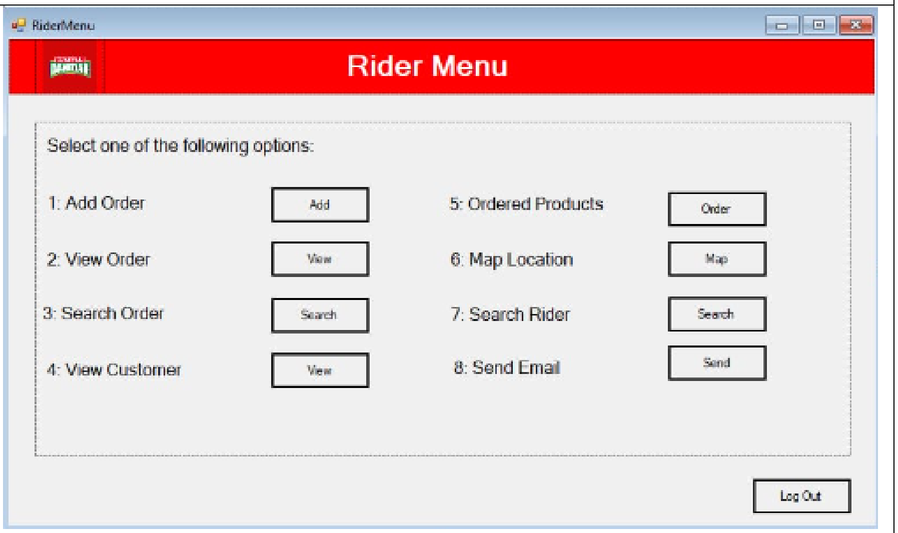
Interface ID	I06
Name	Rider's Menu
Linked Use Case	U37
UI Screen (JustInMind)	 Select one of the following Options... Rider Menu 1. Profile Profile 2. Ordered Products Orders 3. Map Location Map 4. Search Rider Search Log Out
UI Screen	 Select one of the following options: RiderMenu Rider Menu 1: Add Order Add 5: Ordered Products Order 2: View Order View 6: Map Location Map 3: Search Order Search 7: Search Rider Search 4: View Customer View 8: Send Email Send Log Out
Validators	1: Can't Open the required page.

Table 46: I07

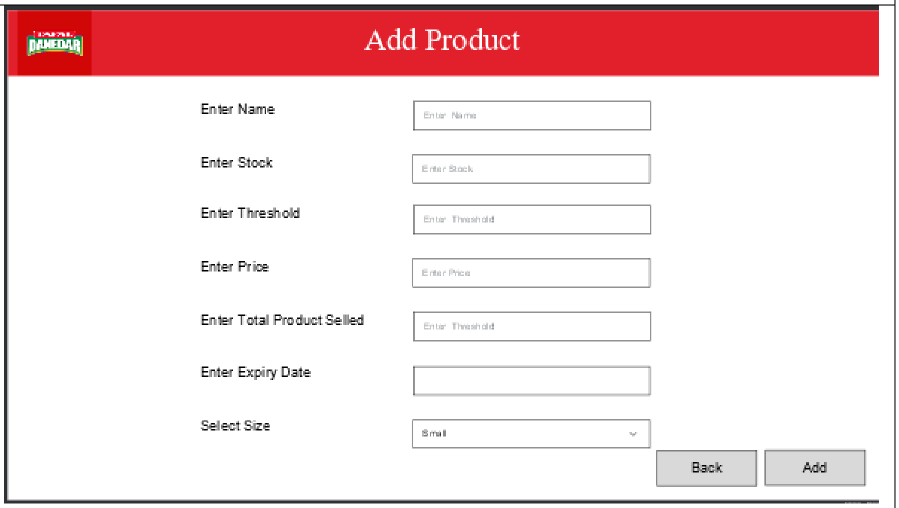
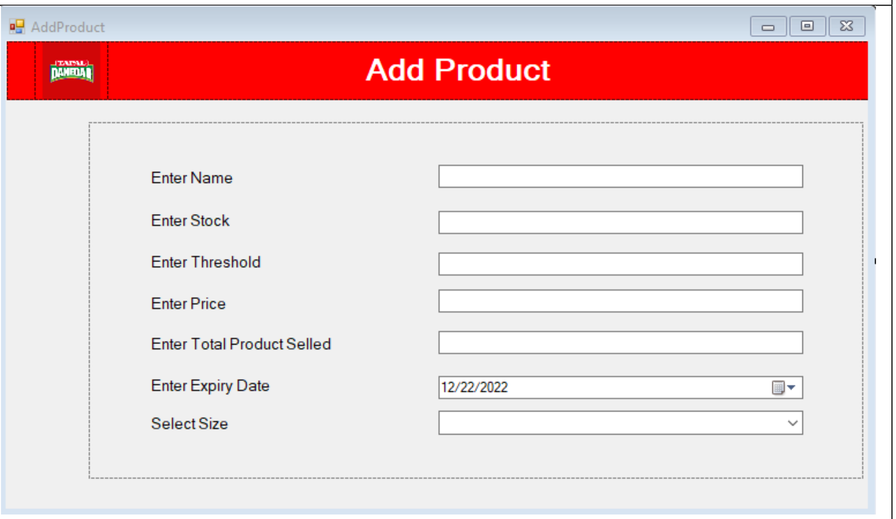
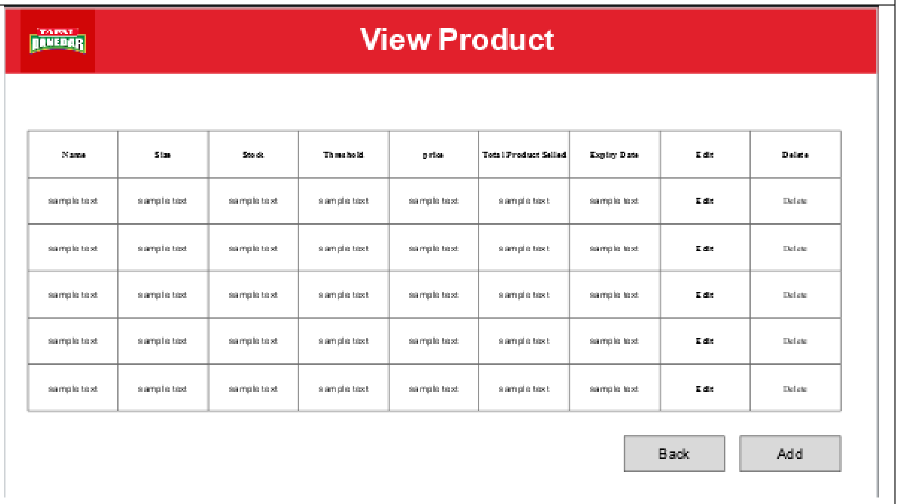
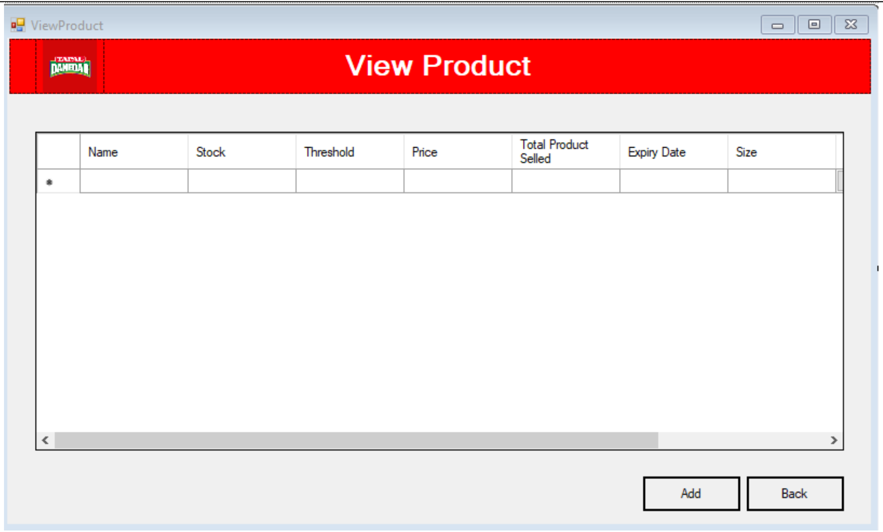
Interface ID	I07
Name	Add product
Linked Use Case	U24
UI Screen (JustInMind)	 Add Product Enter Name: Enter Name Enter Stock: Enter Stock Enter Threshold: Enter Threshold Enter Price: Enter Price Enter Total Product Selled: Enter Threshold Enter Expiry Date: Select Size: Small
UI Screen	 Add Product Enter Name: Enter Stock: Enter Threshold: Enter Price: Enter Total Product Selled: Enter Expiry Date: 12/22/2022 Select Size:
Validators	1: Product cannot be added.

Table 47: I08

<table>
<tr><td>Interface ID</td><td>I08</td></tr>
<tr><td>Name</td><td>View product</td></tr>
<tr><td>Linked Use Case</td><td>U28 , U26</td></tr>
<tr><td>UI Screen (JustInMind)</td><td>

View Product
<table>
<tr><th>Name</th><th>Size</th><th>Stock</th><th>Threshold</th><th>price</th><th>Total Product Selled</th><th>Expiry Date</th><th>Edit</th><th>Delete</th></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
</table>

</td></tr>
<tr><td>UI Screen</td><td>

ViewProduct
View Product
<table>
<tr><th></th><th>Name</th><th>Stock</th><th>Threshold</th><th>Price</th><th>Total Product Selled</th><th>Expiry Date</th><th>Size</th></tr>
<tr><td>*</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

</td></tr>
<tr><td>Validators</td><td>1: Cannot go to the Add Page.
2: Cannot Edit a product.
3: Cannot Delete a Product.
4: Data is not shown in the table.
5: File does not exists.</td></tr>
</table>

Table 48: I09

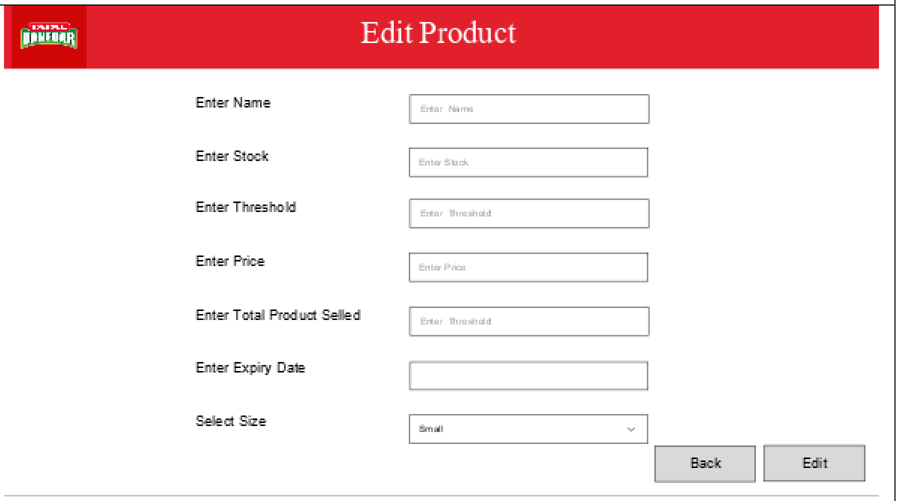
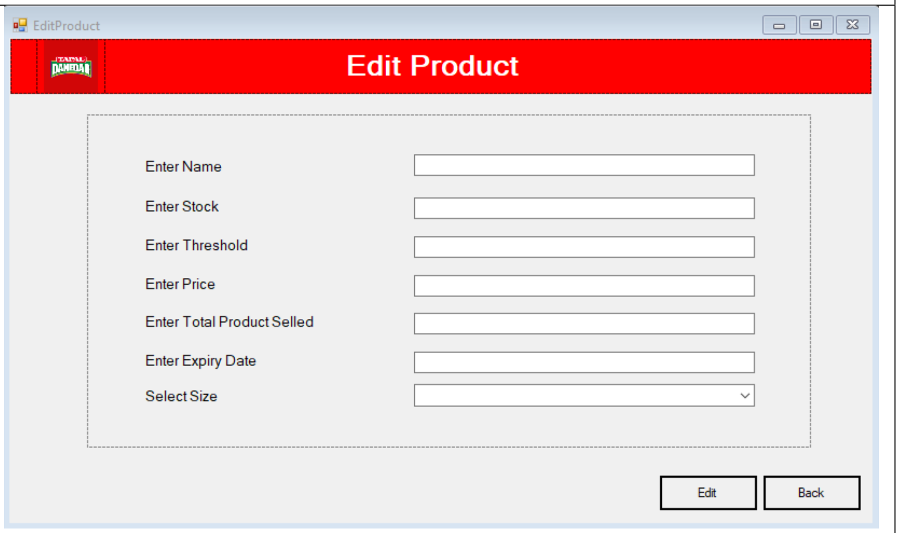
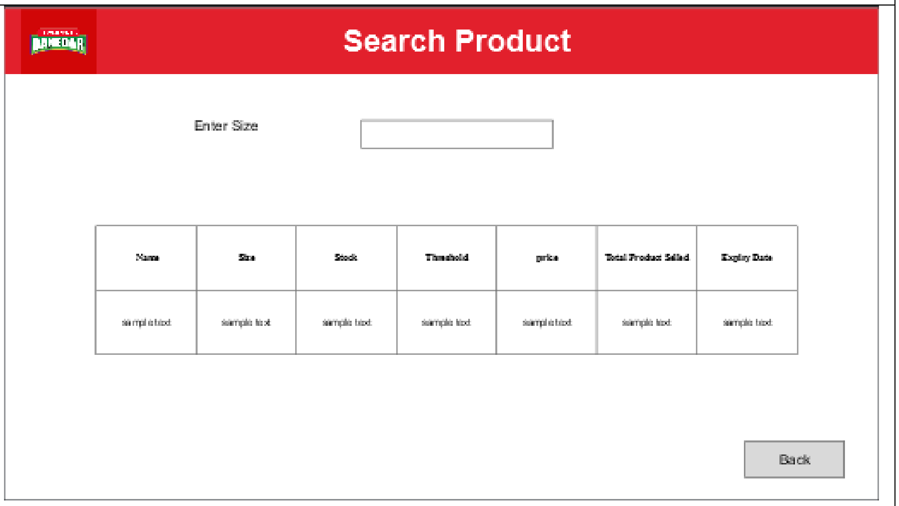
Interface ID	I09
Name	Edit product
Linked Use Case	U25
UI Screen (JustInMind)	 Edit Product Enter Name: Enter Name Enter Stock: Enter Stock Enter Threshold: Enter Threshold Enter Price: Enter Price Enter Total Product Selled: Enter Threshold Enter Expiry Date: Select Size: Small
UI Screen	 EditProduct Edit Product Enter Name Enter Stock Enter Threshold Enter Price Enter Total Product Selled Enter Expiry Date Select Size
Validators	1: Products cannot be edited. 2: Edit button doesn't changes the already present products.

Table 49: I10

<table>
<tr><td>Interface ID</td><td>I10</td></tr>
<tr><td>Name</td><td>Search product</td></tr>
<tr><td>Linked Use Case</td><td>U27</td></tr>
<tr><td>UI Screen (JustInMind)</td><td>

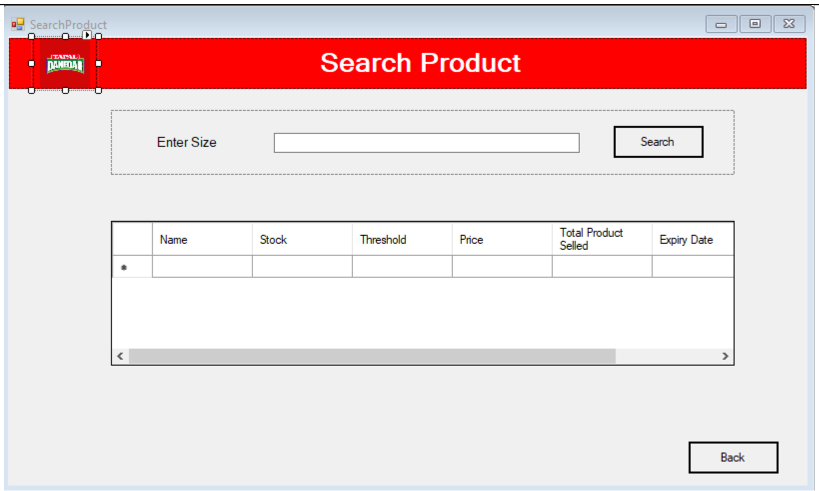
<table>
<tr><th>Name</th><th>Size</th><th>Stock</th><th>Threshold</th><th>price</th><th>Total Product Selled</th><th>Expiry Date</th></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td></tr>
</table></td></tr>
<tr><td>UI Screen</td><td>
SearchProduct
Search Product
Enter Size
Search
Back
<table>
<tr><th></th><th>Name</th><th>Stock</th><th>Threshold</th><th>Price</th><th>Total Product Selled</th><th>Expiry Date</th></tr>
<tr><td>*</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td>Validators</td><td>1: Products cannot be searched.</td></tr>
</table>

Table 50: I11

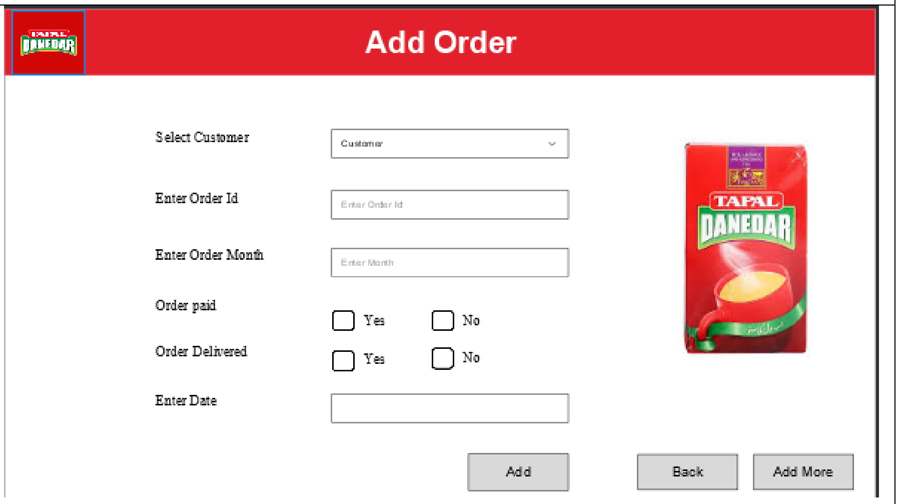
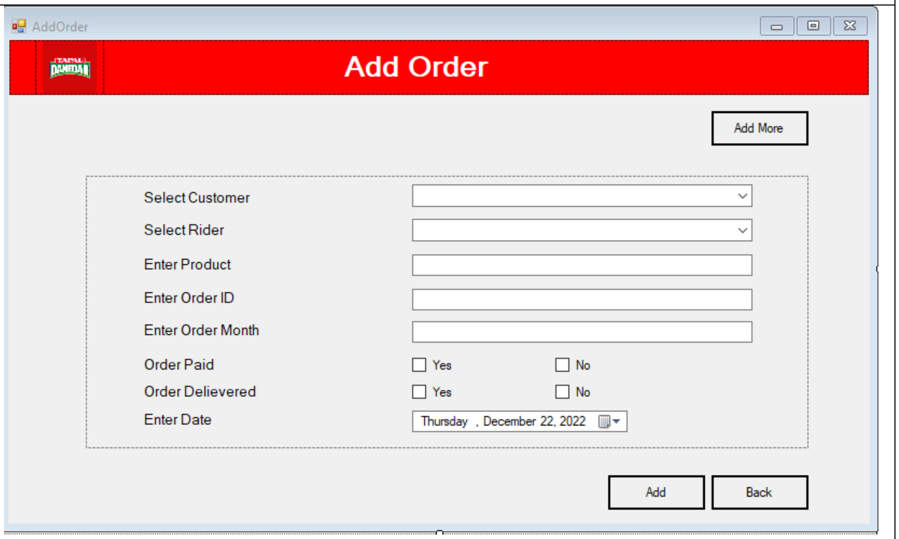
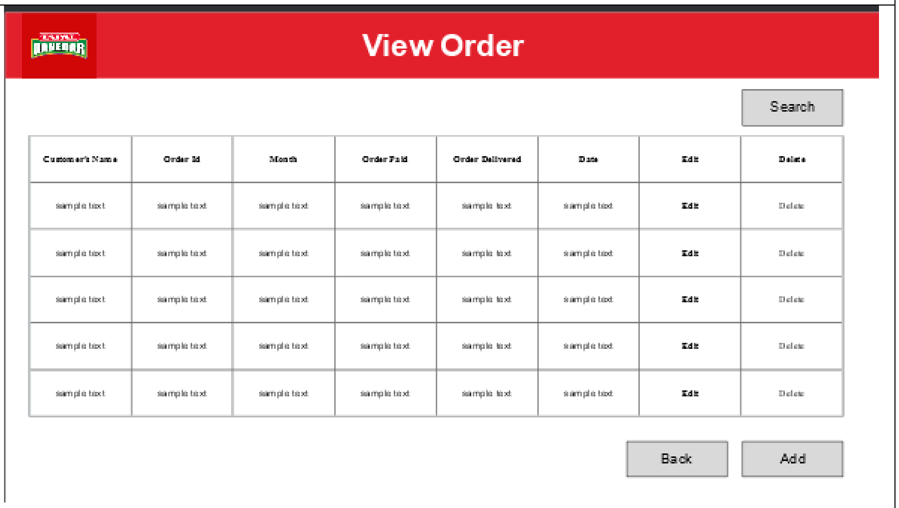
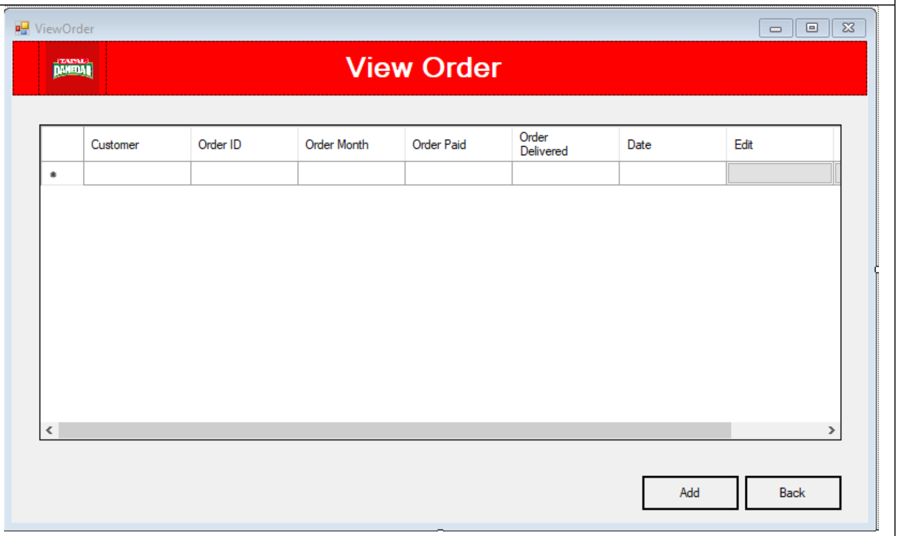
Interface ID	I11
Name	Add Order
Linked Use Case	U14
UI Screen (JustInMind)	 Add Order Select Customer Customer Enter Order Id Enter Order Id Enter Order Month Enter Month Order paid □ Yes □ No Order Delivered □ Yes □ No Enter Date Add Back Add More
UI Screen	 AddOrder Add Order Add More Select Customer Select Rider Enter Product Enter Order ID Enter Order Month Order Paid □ Yes □ No Order Delievered □ Yes □ No Enter Date Add Back Thursday , December 22, 2022
Validators	1: Add more option doesn't work. 2: Product cannot be added.

Table 51: I12

<table>
<tr><td>Interface ID</td><td>I12</td></tr>
<tr><td>Name</td><td>View Order</td></tr>
<tr><td>Linked Use Case</td><td>U16, U18</td></tr>
<tr><td>UI Screen (JustInMind)</td><td>

<table>
<tr><th>Customer's Name</th><th>Order Id</th><th>Month</th><th>Order Paid</th><th>Order Delivered</th><th>Date</th><th>Edit</th><th>Delete</th></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
</table></td></tr>
<tr><td>UI Screen</td><td>

<table>
<tr><th></th><th>Customer</th><th>Order ID</th><th>Order Month</th><th>Order Paid</th><th>Order Delivered</th><th>Date</th><th>Edit</th></tr>
<tr><td>*</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td>Validators</td><td>1: Can't go to search page.
2: Can't opens up the Add page.
3: Can't Edit the Order.
4: Can't Delete the order.
5: Data is not shown in the table.
6: File does not exists.</td></tr>
</table>

Table 52: I13

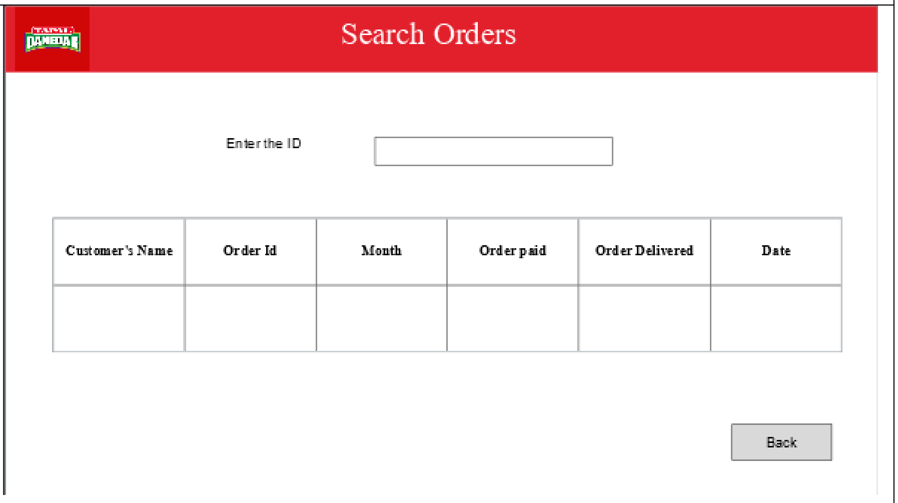
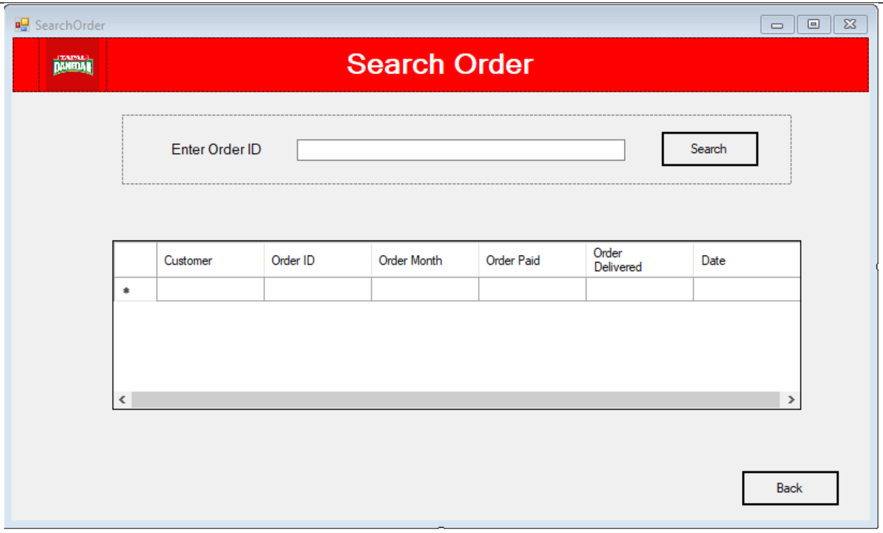
Interface ID	I13
Name	Search Order
Linked Use Case	U17
UI Screen (JustInMind)	 Search Orders Enter the ID Customer's Name Order Id Month Order paid Order Delivered Date Back
UI Screen	 SearchOrder Search Order Enter Order ID Search Customer Order ID Order Month Order Paid Order Delivered Date Back
Validators	1: There is no order placed against the entered ID. 2: Order cannot be shown in the table. 3: File doesn't exists.

Table 53: I14

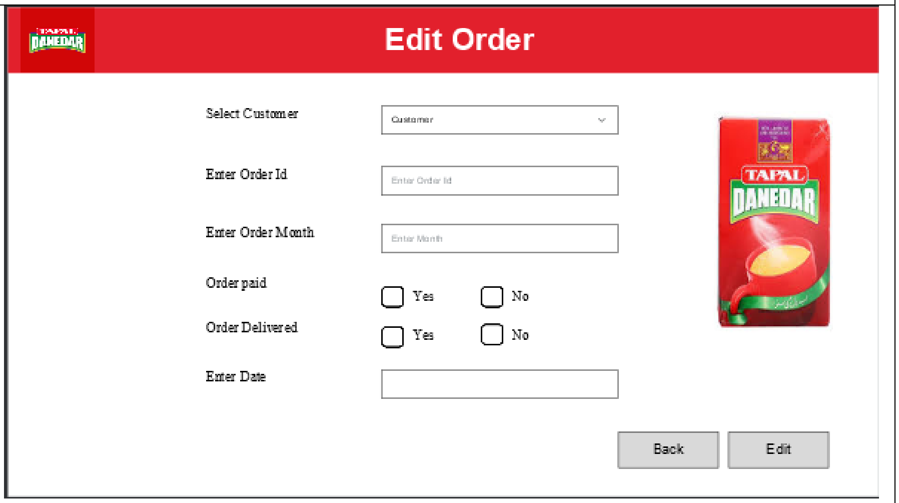
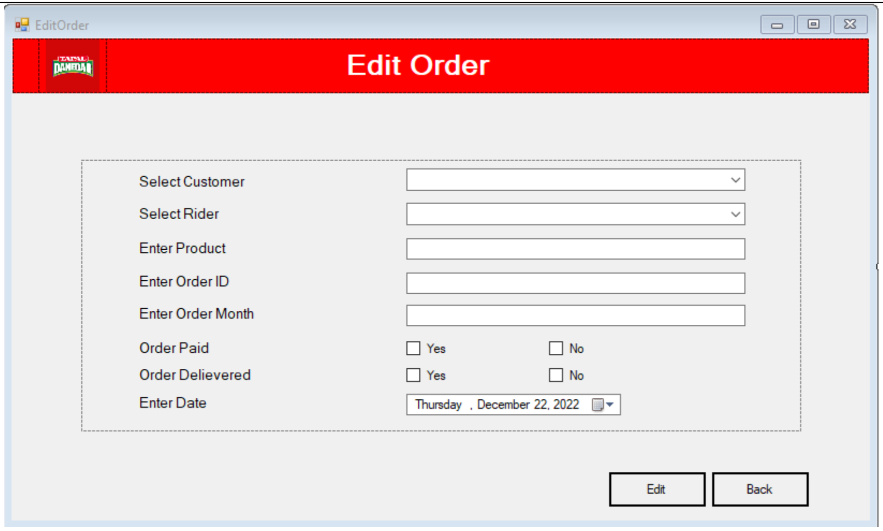
Interface ID	I14
Name	Edit Order
Linked Use Case	U15
UI Screen (JustInMind)	 Edit Order Select Customer Customer Enter Order Id Enter Order Id Enter Order Month Enter Month Order paid □ Yes □ No Order Delivered □ Yes □ No Enter Date
UI Screen	 EditOrder Edit Order Select Customer Select Rider Enter Product Enter Order ID Enter Order Month Order Paid □ Yes □ No Order Delievered □ Yes □ No Enter Date Thursday , December 22, 2022
Validators	1: Order cannot be edited. 2: Edit button doesn't changes the already present orders.

Table 54: I15

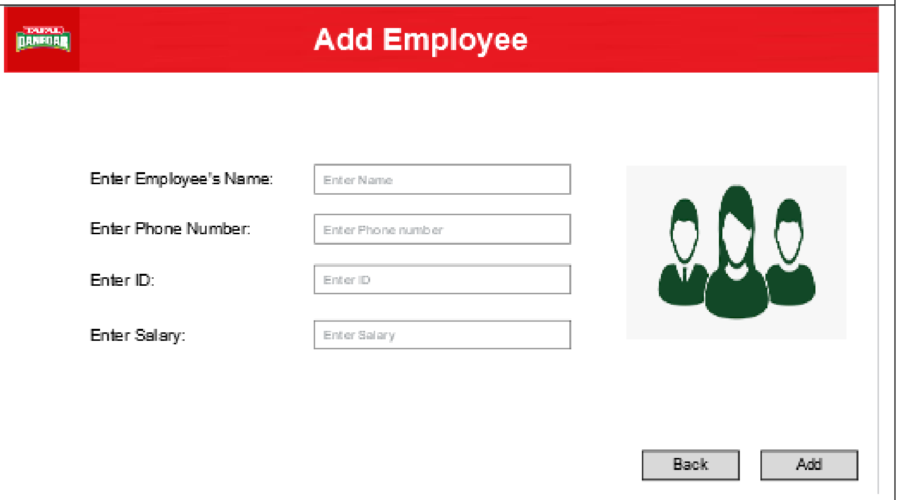
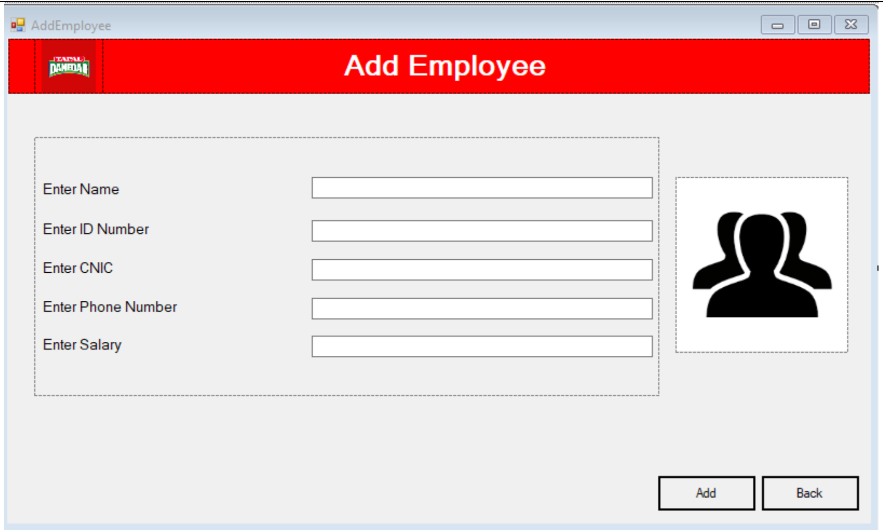
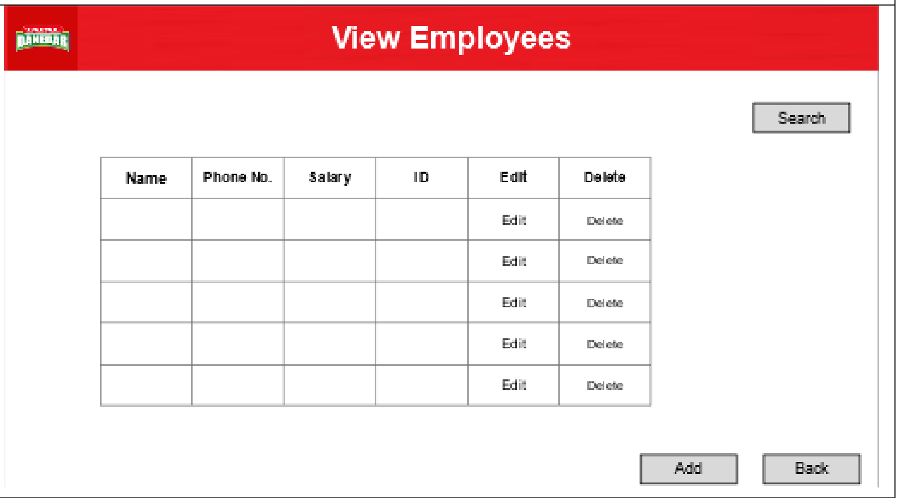
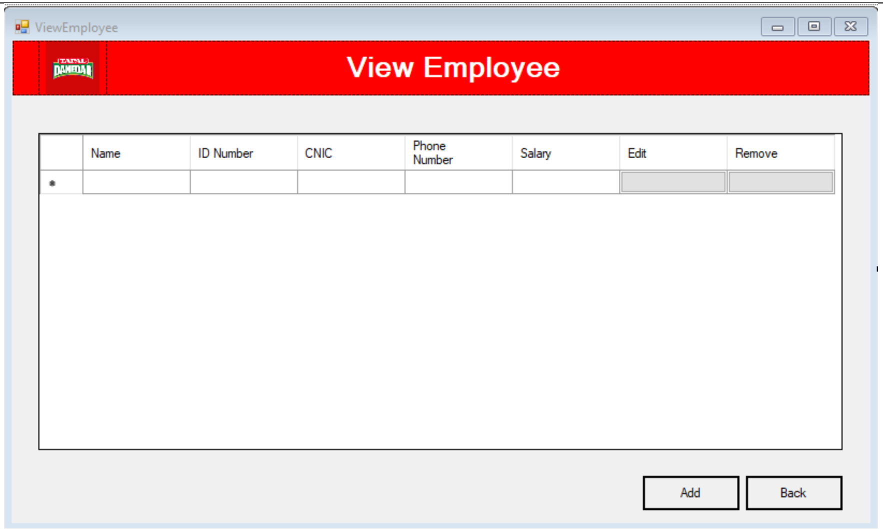
Interface ID	I15
Name	Add Employee
Linked Use Case	U19
UI Screen (JustInMind)	 Add Employee Enter Employee's Name: Enter Name Enter Phone Number: Enter Phone number Enter ID: Enter ID Enter Salary: Enter Salary Back Add
UI Screen	 AddEmployee Add Employee Enter Name Enter ID Number Enter CNIC Enter Phone Number Enter Salary Add Back
Validators	1: Employee cannot be added.

Table 55: I16

<table>
<tr><td>Interface ID</td><td>I16</td></tr>
<tr><td>Name</td><td>View Employee</td></tr>
<tr><td>Linked Use Case</td><td>U21, U23</td></tr>
<tr><td>UI Screen (JustInMind)</td><td>

<table>
<tr><th>Name</th><th>Phone No.</th><th>Salary</th><th>ID</th><th>Edit</th><th>Delete</th></tr>
<tr><td></td><td></td><td></td><td></td><td>Edit</td><td>Delete</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Edit</td><td>Delete</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Edit</td><td>Delete</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Edit</td><td>Delete</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Edit</td><td>Delete</td></tr>
</table></td></tr>
<tr><td>UI Screen</td><td>

<table>
<tr><th></th><th>Name</th><th>ID Number</th><th>CNIC</th><th>Phone Number</th><th>Salary</th><th>Edit</th><th>Remove</th></tr>
<tr><td>*</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td>Validators</td><td>1: Search page doesn't opens.
2: Add button doesn't show add page.
3: Edit option doesn't work.
4: Delete option doesn't work.
5: Data is not shown in the table.
6: File does not exists.</td></tr>
</table>

Table 56: I17

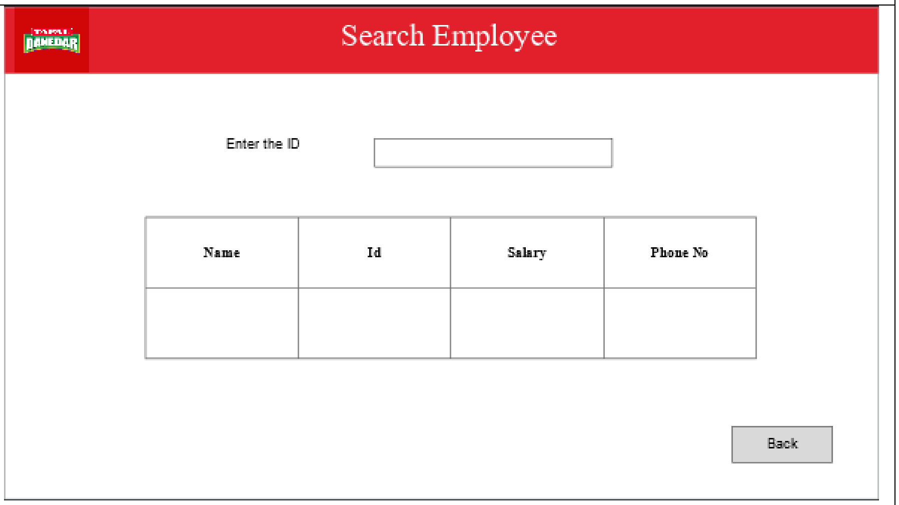
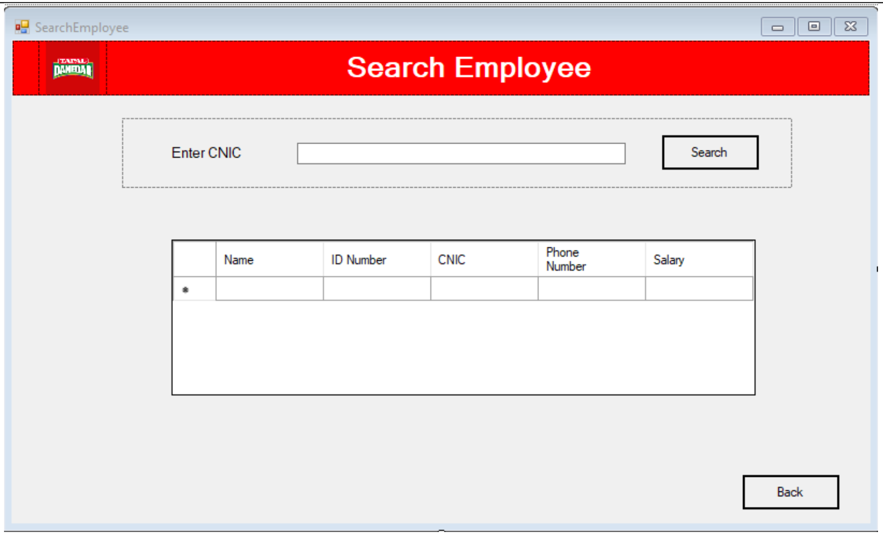
Interface ID	I17
Name	Search Employee
Linked Use Case	U22
UI Screen (JustInMind)	 Search Employee Enter the ID Name Id Salary Phone No Back
UI Screen	 SearchEmployee Search Employee Enter CNIC Search Name ID Number CNIC Phone Number Salary Back
Validators	1: There is no employee present against the entered ID. 2: Employee cannot be shown in the table. 3: File doesn't exists.

Table 57: I18

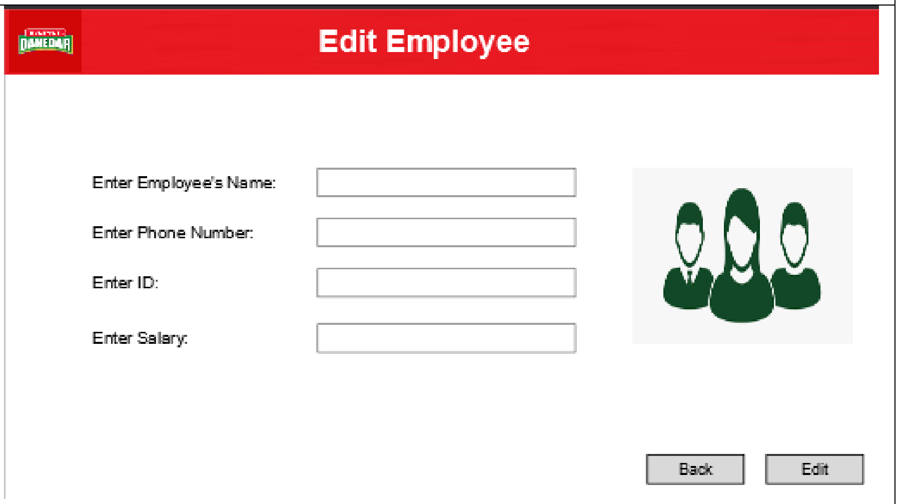
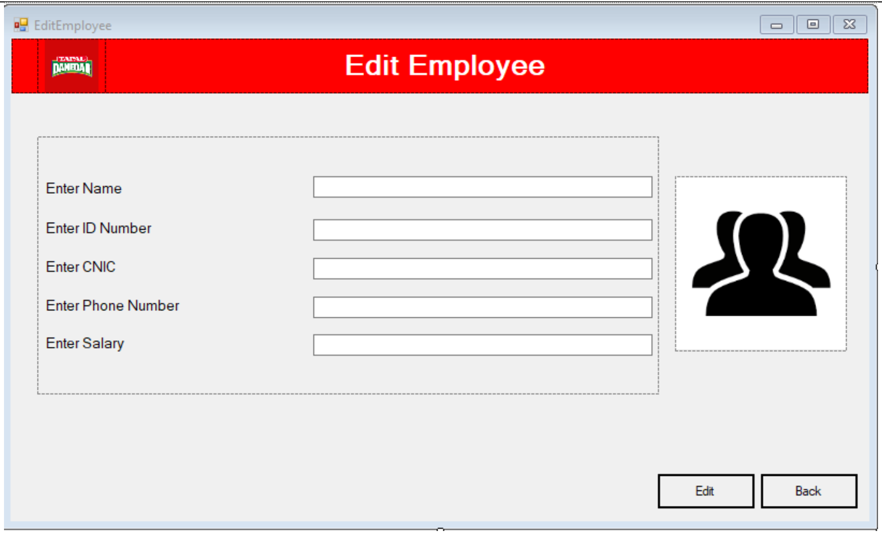
Interface ID	I18
Name	Edit Employee
Linked Use Case	U20
UI Screen (JustInMind)	 Edit Employee Enter Employee's Name: Enter Phone Number: Enter ID: Enter Salary: Back Edit
UI Screen	 EditEmployee Edit Employee Enter Name Enter ID Number Enter CNIC Enter Phone Number Enter Salary Edit Back
Validators	1: Employee cannot be edited. 2: Edit button doesn't change the already present employee.

Table 58: I19

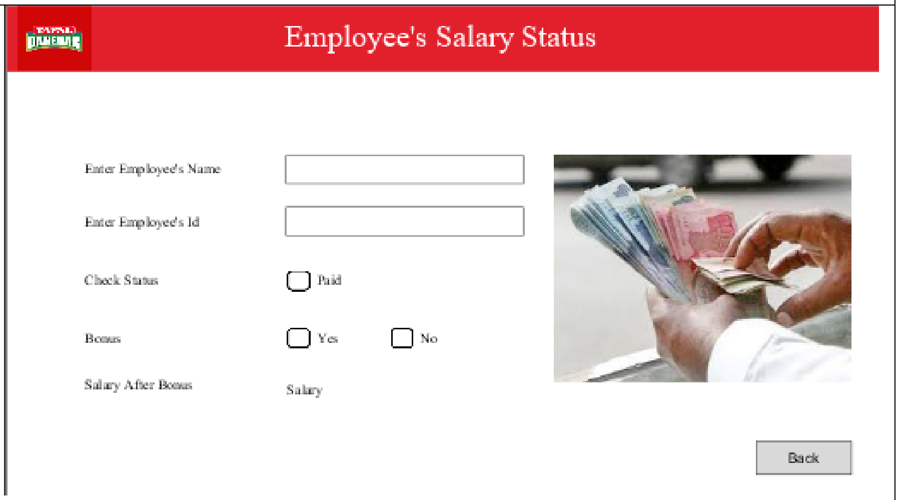
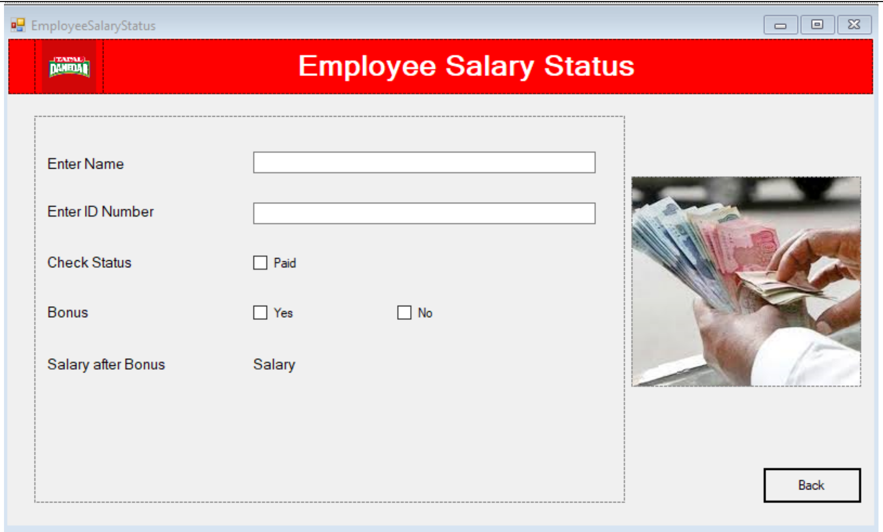
Interface ID	I19
Name	Employee Salary
Linked Use Case	U31
UI Screen (JustInMind)	 Employee's Salary Status Enter Employee's Name Enter Employee's Id Check Status □ Paid Bonus □ Yes □ No Salary After Bonus Salary Back
UI Screen	 EmployeeSalaryStatus Employee Salary Status Enter Name Enter ID Number Check Status □ Paid Bonus □ Yes □ No Salary after Bonus Salary Back
Validators	1: There is no employee present against the entered Id. 2: Check box error. 3: Error in Salary calculation.

Table 59: I20

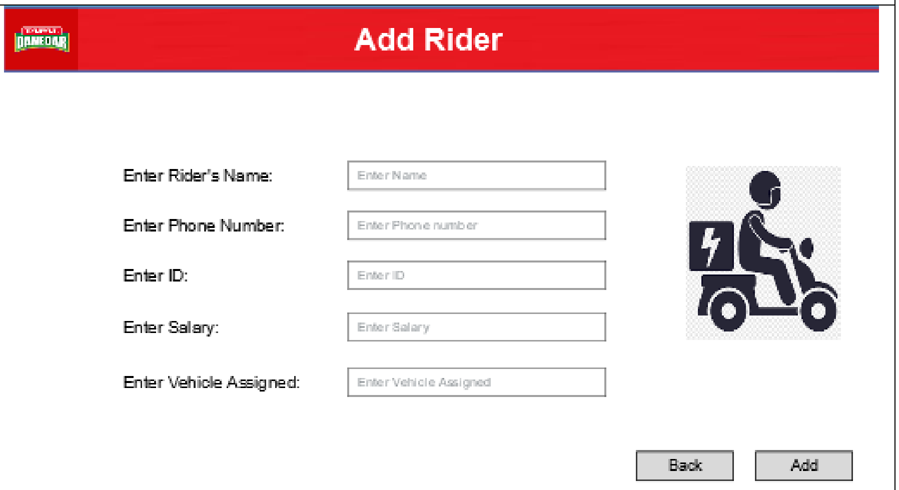
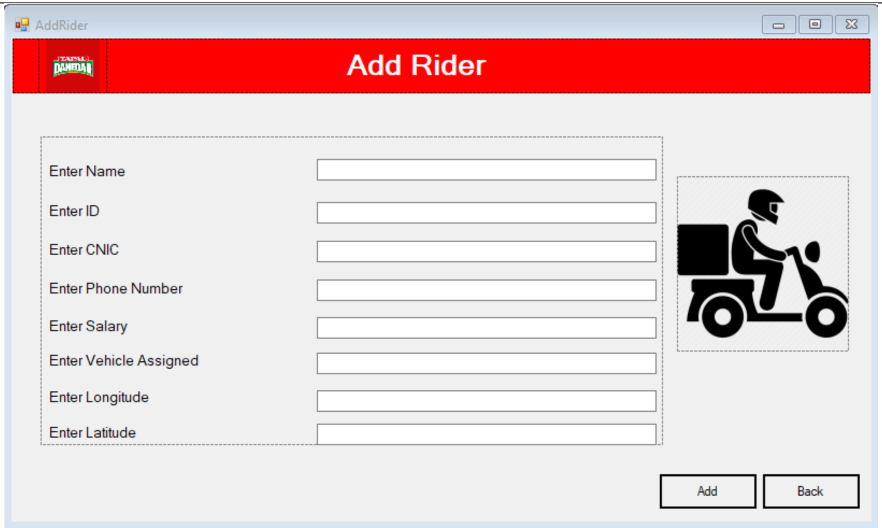
Interface ID	I20
Name	Add Rider
Linked Use Case	U04
UI Screen (JustInMind)	 Add Rider Enter Rider's Name: Enter Name Enter Phone Number: Enter Phone number Enter ID: Enter ID Enter Salary: Enter Salary Enter Vehicle Assigned: Enter Vehicle Assigned Back Add
UI Screen	 AddRider Add Rider Enter Name Enter ID Enter CNIC Enter Phone Number Enter Salary Enter Vehicle Assigned Enter Longitude Enter Latitude Add Back
Validators	1: Rider cannot be added.

Table 60: I21

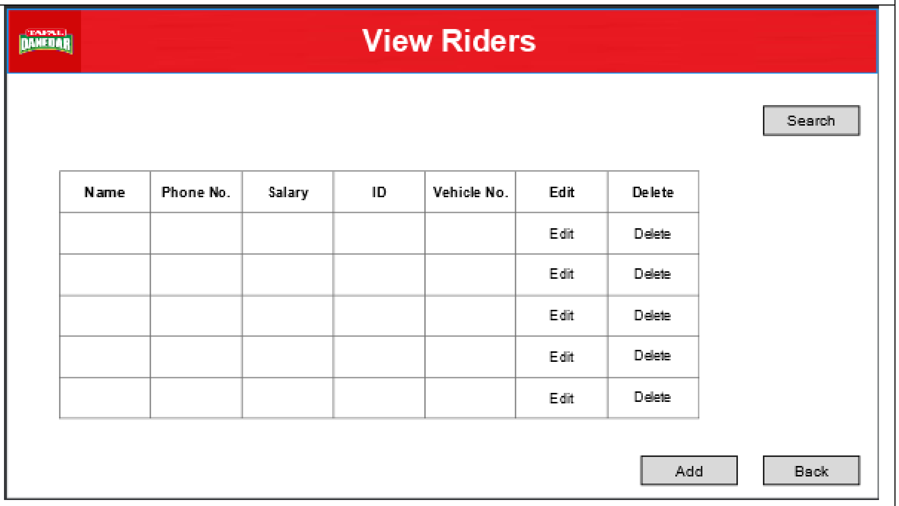
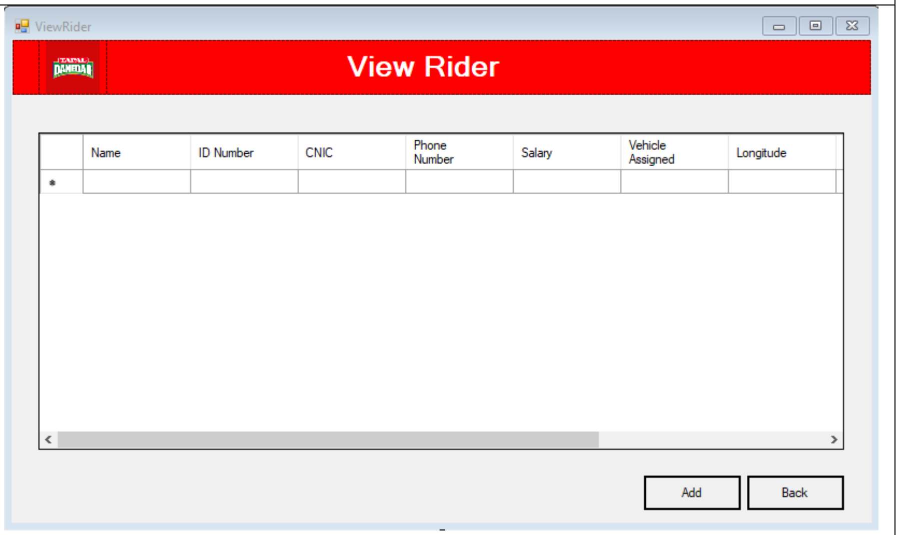
Interface ID	I21
Name	View Rider
Linked Use Case	U06, U08
UI Screen (JustInMind)	 View Riders Search Name Phone No. Salary ID Vehicle No. Edit Delete Edit Delete Edit Delete Edit Delete Edit Delete Edit Delete Add Back
UI Screen	 ViewRider View Rider Name ID Number CNIC Phone Number Salary Vehicle Assigned Longitude Add Back
Validators	1: Search page doesn't opens. 2: Add button doesn't show add page. 3: Edit option doesn't work. 4: Delete option doesn't work. 5: Data is not shown in the table. 6: File does not exists.

Table 61: I22

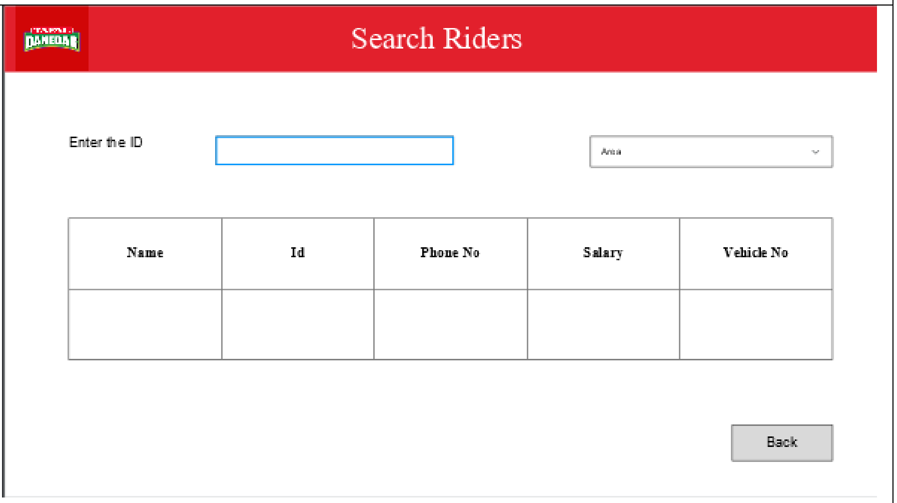
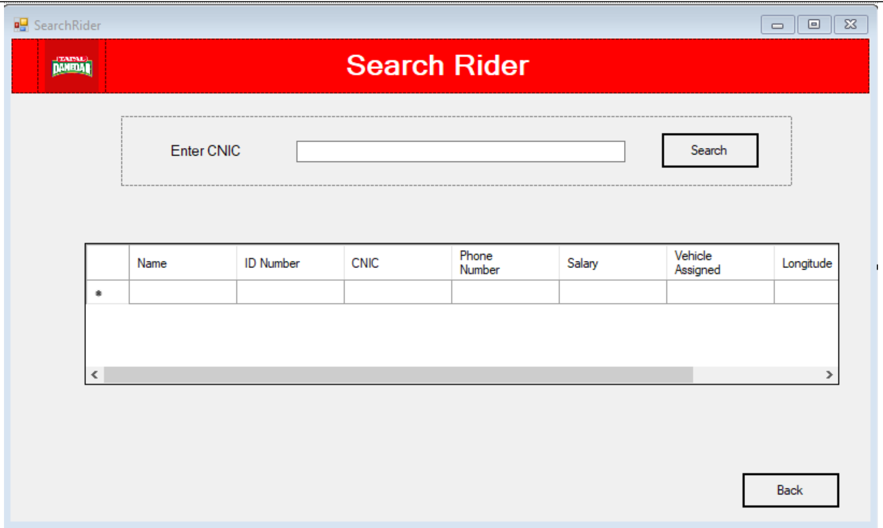
Interface ID	I22
Name	Search Rider
Linked Use Case	U07
UI Screen (JustInMind)	 Search Riders Enter the ID Area Name Id Phone No Salary Vehicle No Back
UI Screen	 SearchRider Search Rider Enter CNIC Search Name ID Number CNIC Phone Number Salary Vehicle Assigned Longitude Back
Validators	1: There is no rider present against the entered ID. 2: Rider cannot be shown in the table. 3: File doesn't exists.

Table 62: I23

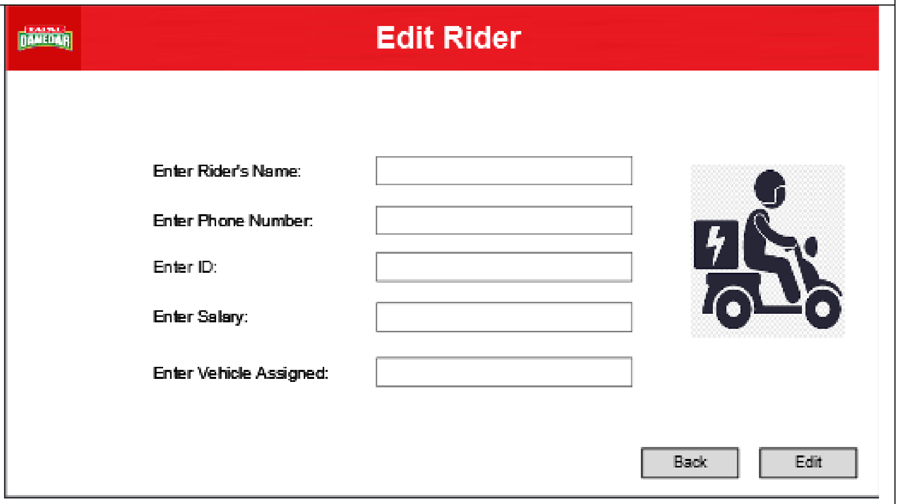
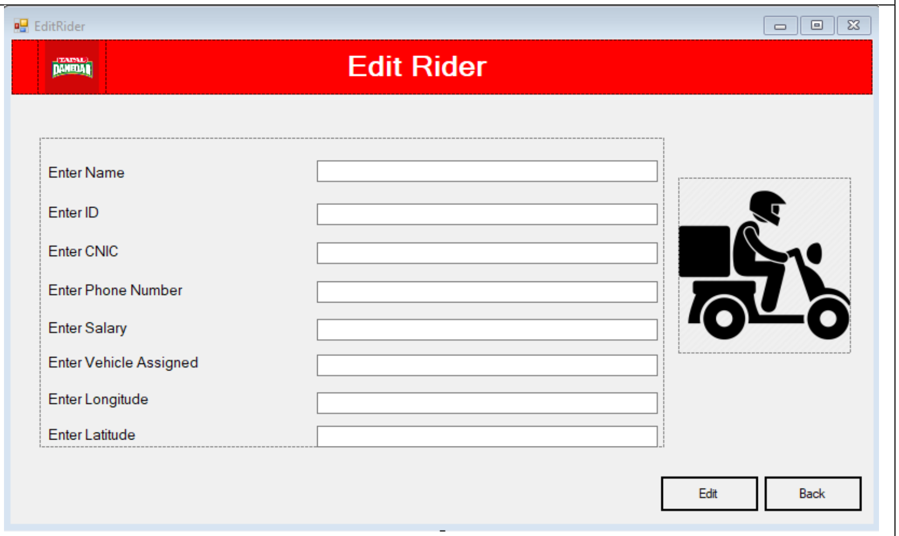
Interface ID	I23
Name	Edit Rider
Linked Use Case	U05
UI Screen (JustInMind)	 Edit Rider Enter Rider's Name: Enter Phone Number: Enter ID: Enter Salary: Enter Vehicle Assigned:
UI Screen	 Edit Rider Enter Name Enter ID Enter CNIC Enter Phone Number Enter Salary Enter Vehicle Assigned Enter Longitude Enter Latitude
Validators	1: Rider cannot be edited. 2: Edit button doesn't change the already present rider.

Table 63: I24

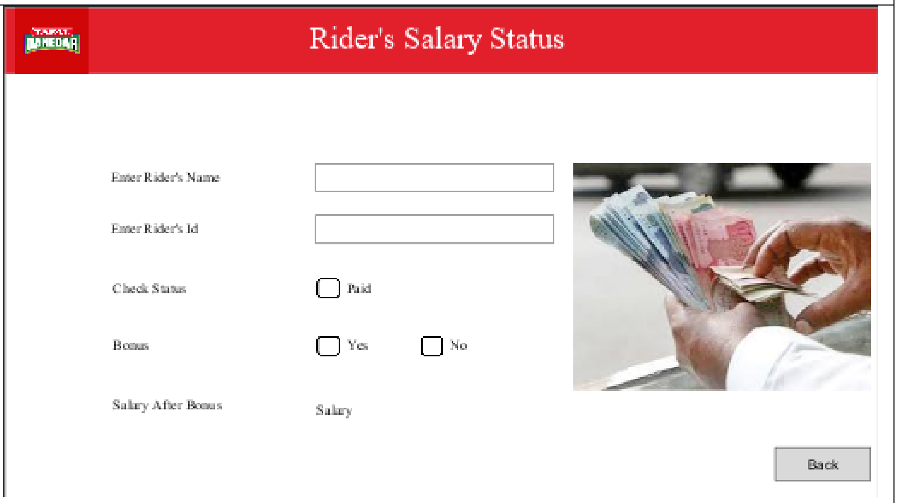
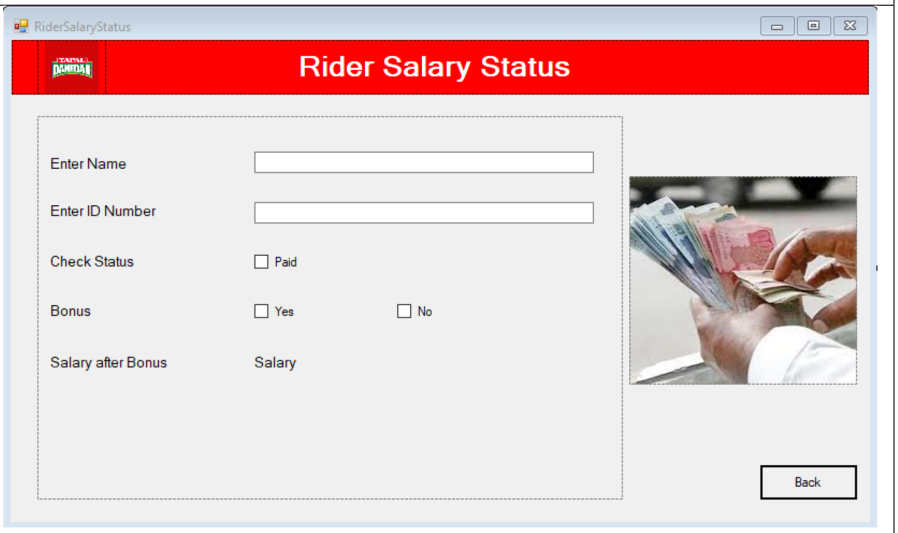
Interface ID	I24
Name	Rider's Salary
Linked Use Case	U30
UI Screen (JustInMind)	 Rider's Salary Status Enter Rider's Name Enter Rider's Id Check Status □ Paid Bonus □ Yes □ No Salary After Bonus Salary Back
UI Screen	 RiderSalaryStatus Rider Salary Status Enter Name Enter ID Number Check Status □ Paid Bonus □ Yes □ No Salary after Bonus Salary Back
Validators	1: There is no rider present against the entered Id. 2: Check box error. 3: Error in Salary calculation.

Table 64: I25

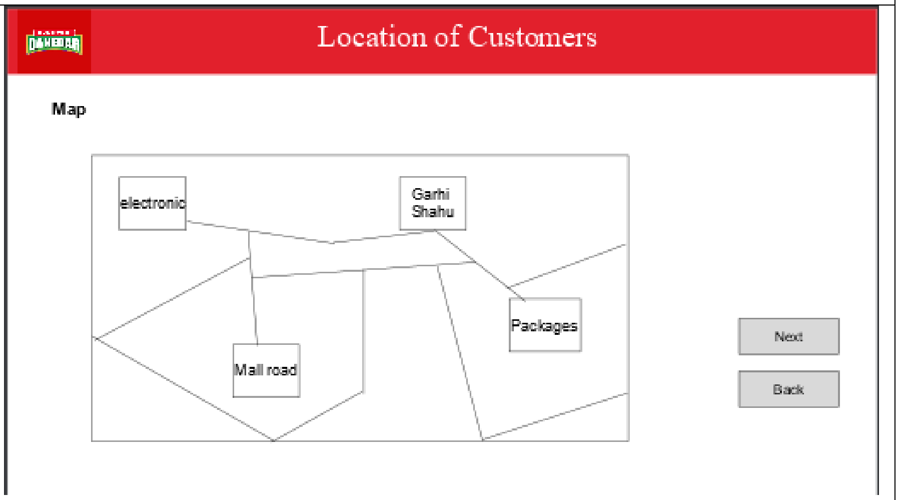
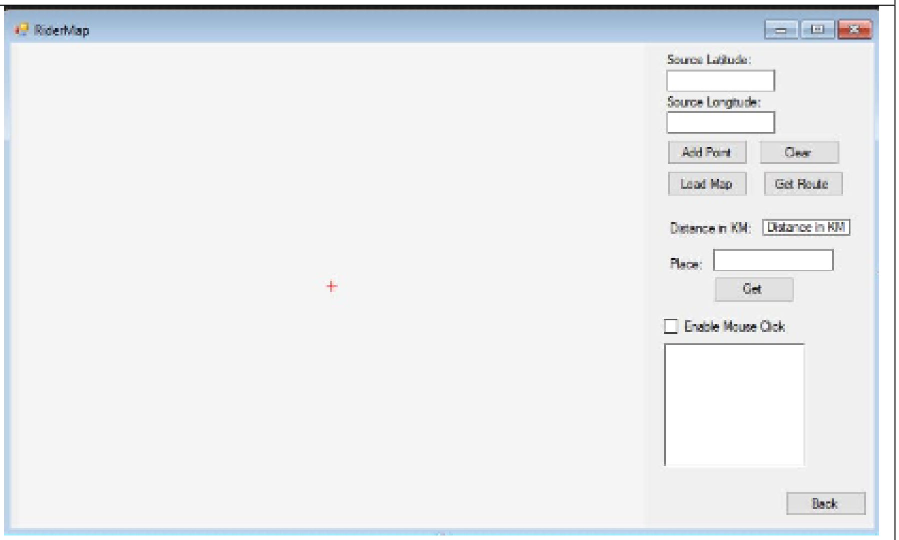
Interface ID	I25
Name	Rider's Map
Linked Use Case	U35
UI Screen (JustInMind)	 Location of Customers Map electronic Garhi Shahu Packages Mall road Next Back
UI Screen	 RiderMap Source Latitude: Source Longitude: Add Point Clear Load Map Get Route Distance in KM: Distance in KM Place: Get □ Enable Mouse Click Back
Validators	1: Map doesn't work. 2: Location is not shown.

Table 65: I26

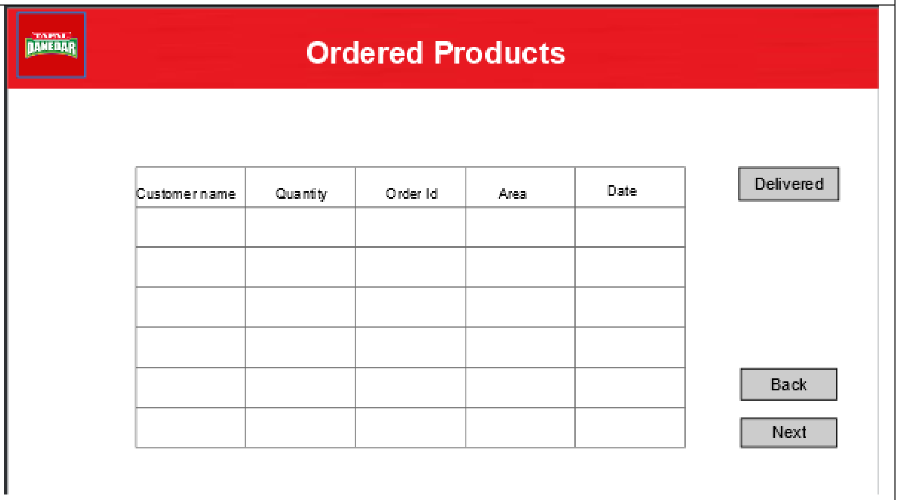
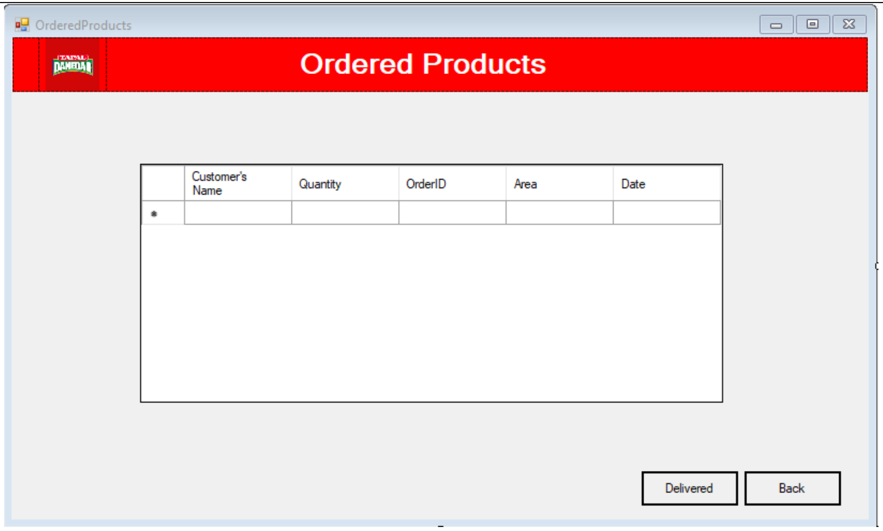
Interface ID	I26
Name	Rider Order Product
Linked Use Case	U34
UI Screen (JustInMind)	 Ordered Products Customer name | Quantity | Order Id | Area | Date
UI Screen	 OrderedProducts Ordered Products Customer's Name | Quantity | OrderID | Area | Date
Validators	1: Products doesn't show in the table. 2: File loading error.

Table 66: I27

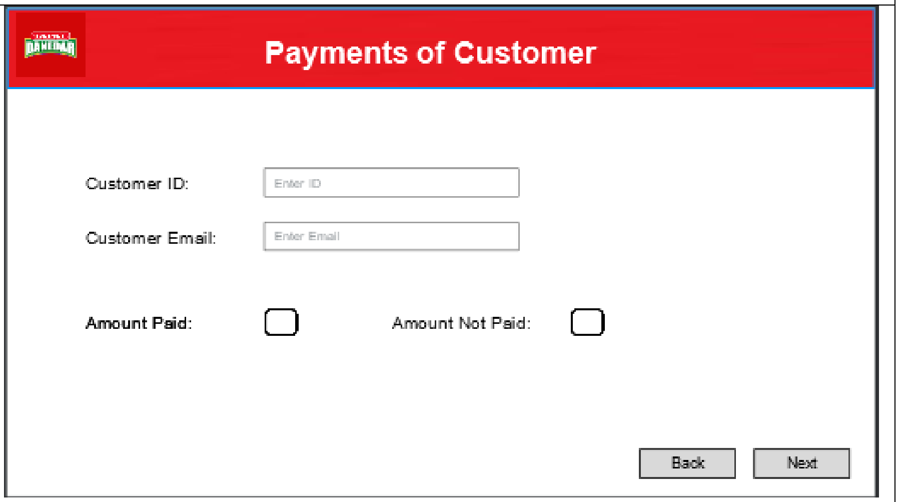
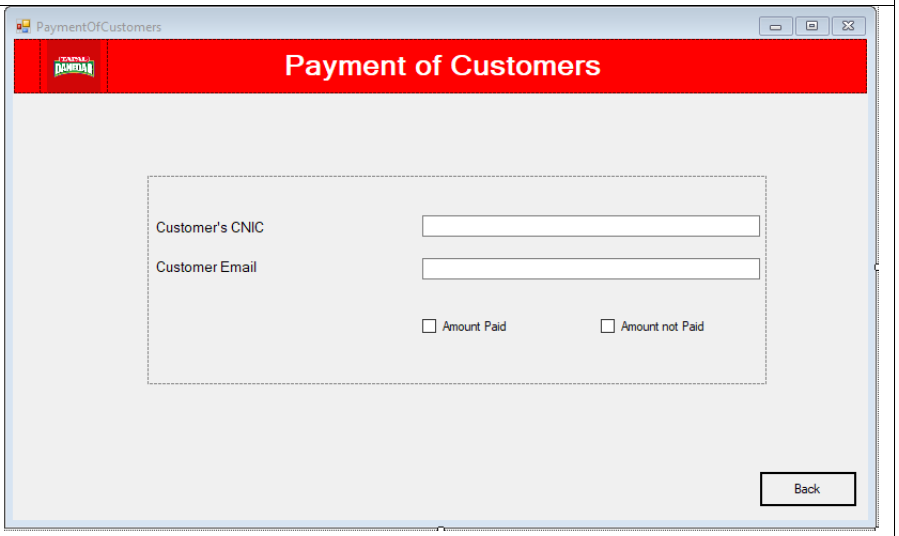
Interface ID	I27
Name	Rider Customer Payment
Linked Use Case	U36
UI Screen (JustInMind)	 Payments of Customer Customer ID: Enter ID Customer Email: Enter Email Amount Paid: □ Amount Not Paid: □ Back Next
UI Screen	 PaymentOfCustomers Payment of Customers Customer's CNIC Customer Email □ Amount Paid □ Amount not Paid Back
Validators	1: Id not found. 2: Check box validation. 3: Email not found.

Table 67: I28

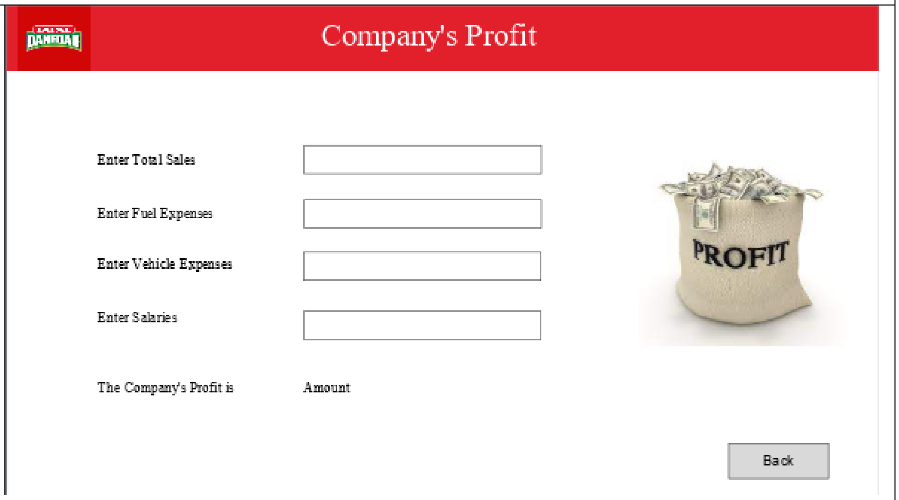
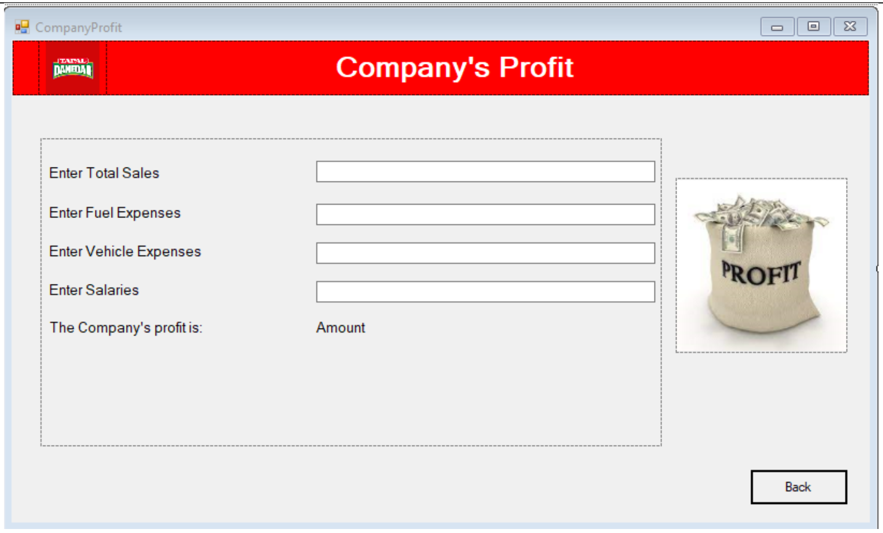
Interface ID	I28
Name	Company's Profit
Linked Use Case	U29
UI Screen (JustInMind)	 Company's Profit Enter Total Sales Enter Fuel Expenses Enter Vehicle Expenses Enter Salaries The Company's Profit is Amount PROFIT Back
UI Screen	 CompanyProfit Company's Profit Enter Total Sales Enter Fuel Expenses Enter Vehicle Expenses Enter Salaries The Company's profit is: Amount PROFIT Back
Validators	1: Calculation error

Table 68: I29

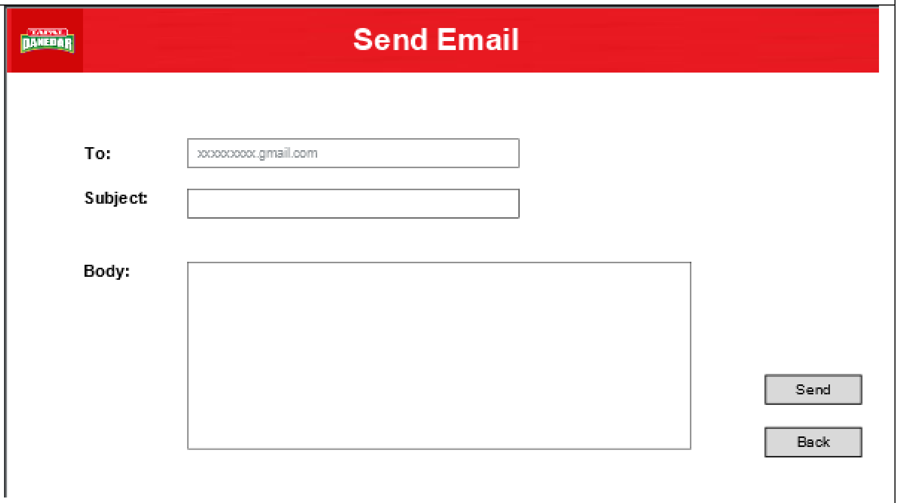
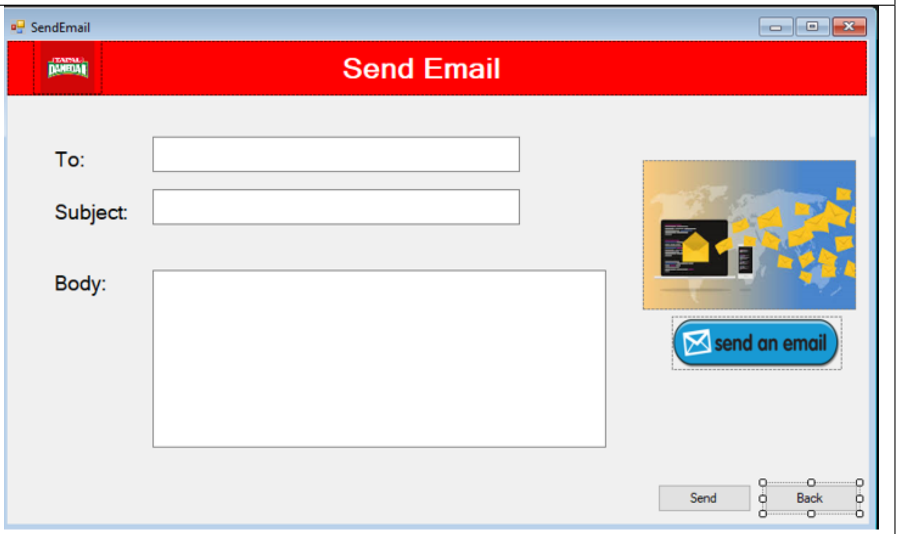
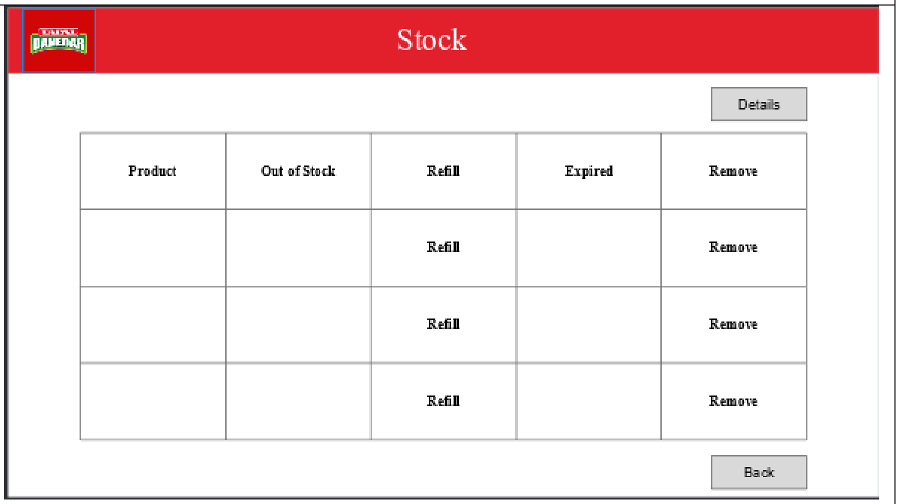
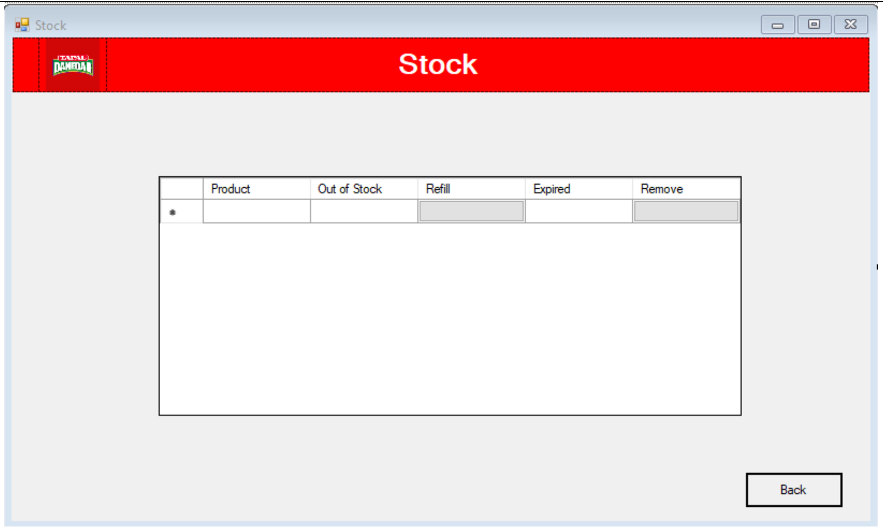
Interface ID	I29
Name	Send Email
Linked Use Case	U32
UI Screen (JustInMind)	 Send Email To: xxxxxxxxx.gmail.com Subject: Body: Send Back
UI Screen	 SendEmail Send Email To: Subject: Body: send an email Send Back
Validators	1: Email not sent

Table 69: I30

<table>
<tr><td>Interface ID</td><td>I30</td></tr>
<tr><td>Name</td><td>Stock</td></tr>
<tr><td>Linked Use Case</td><td></td></tr>
<tr><td>UI Screen (JustInMind)</td><td>


</td></tr>
<tr><td>UI Screen</td><td>


</td></tr>
<tr><td>Validators</td><td>1: Product not found.
2: Cannot Refill.
3: Cannot Remove.</td></tr>
</table>

Table 70: I31

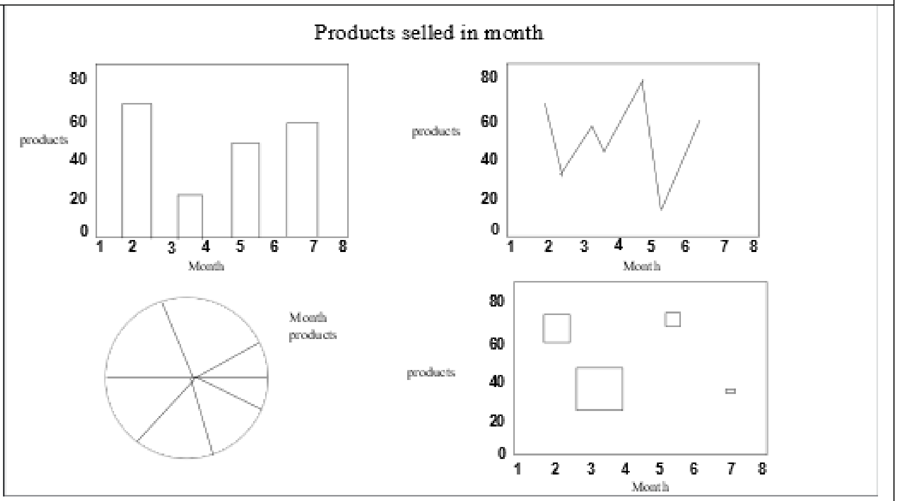
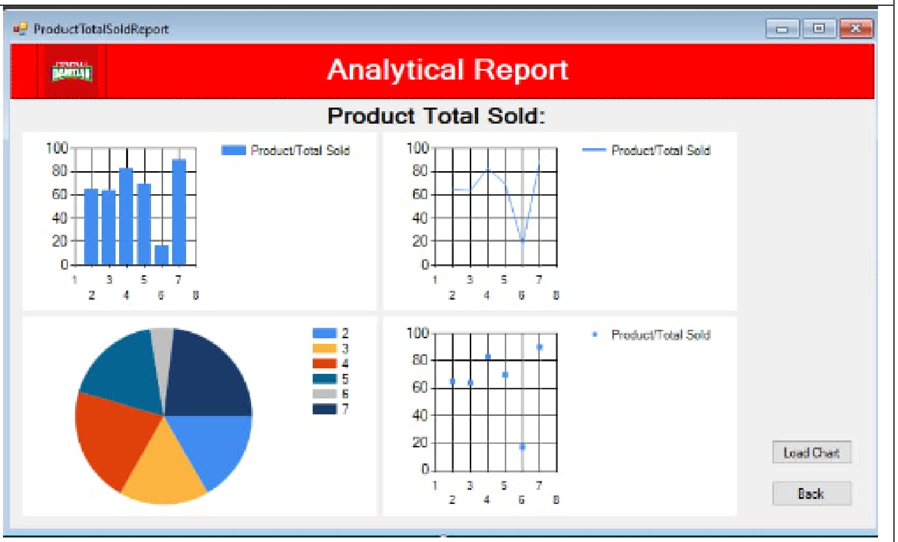
Interface ID	I31
Name	Graph
Linked Use Case	U33
UI Screen (JustInMind)	
UI Screen	 ProductTotalSoldReport Analytical Report Product Total Sold: Product/Total Sold 100 80 60 40 20 0 1 2 3 4 5 6 7 8 2 3 4 5 6 7 Load Chart Back
Validators	1: Data lost. 2: Data doesn't show in the Graph.

Table 71: I32

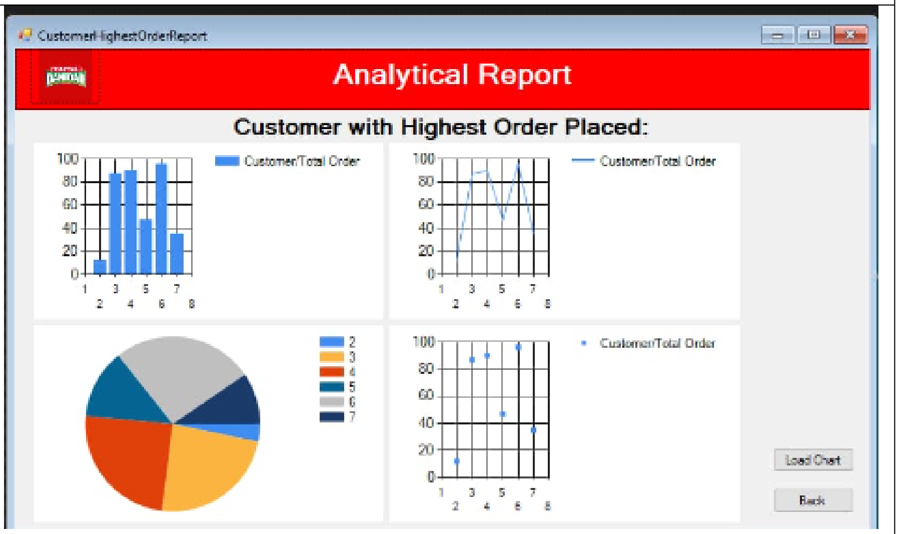
Interface ID	I32
Name	Graph
Linked Use Case	U33
UI Screen	 CustomerHighestOrderReport Analytical Report Customer with Highest Order Placed: Customer/Total Order 100 80 60 40 20 0 1 2 3 4 5 6 7 8 2 3 4 5 6 7 Load Chart Back
Validators	1: Data lost. 2: Data doesn't show in the Graph.

Table 72: I33

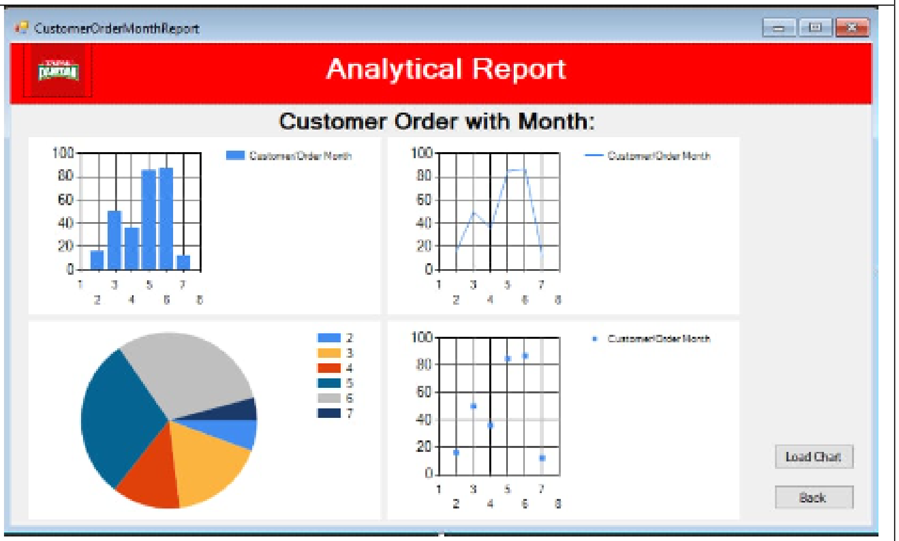
Interface ID	I33
Name	Graph
Linked Use Case	U33
UI Screen	 CustomerOrderMonthReport Analytical Report Customer Order with Month: CustomerOrder Month 100 80 60 40 20 0 1 2 3 4 5 6 7 8 2 3 4 5 6 7 Load Chart Back
Validators	1: Data lost. 2: Data doesn't show in the Graph.

Table 73: I34

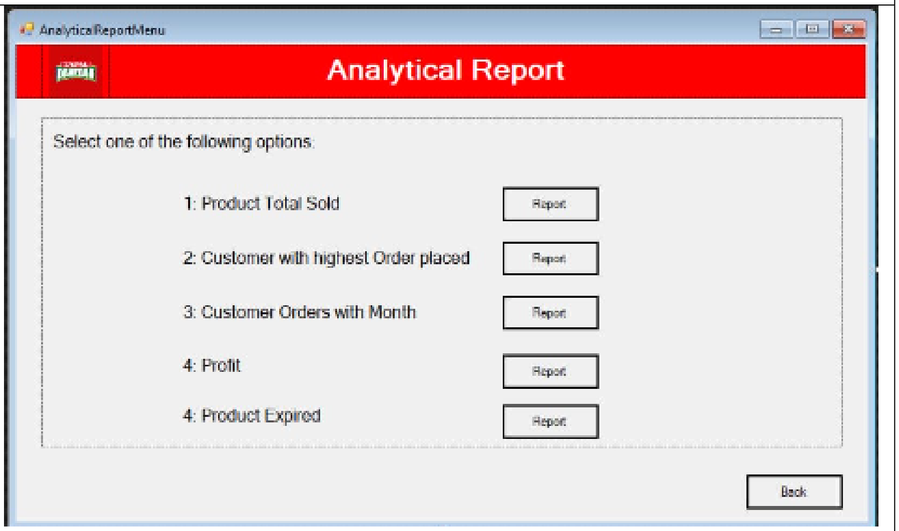
Interface ID	I34
Name	Graph
Linked Use Case	U33
UI Screen	 AnalyticalReportMenu Analytical Report Select one of the following options: 1: Product Total Sold 2: Customer with highest Order placed 3: Customer Orders with Month 4: Profit 4: Product Expired
Validators	1: Data lost. 2: Data doesn't show in the Graph.

Table 74: I35

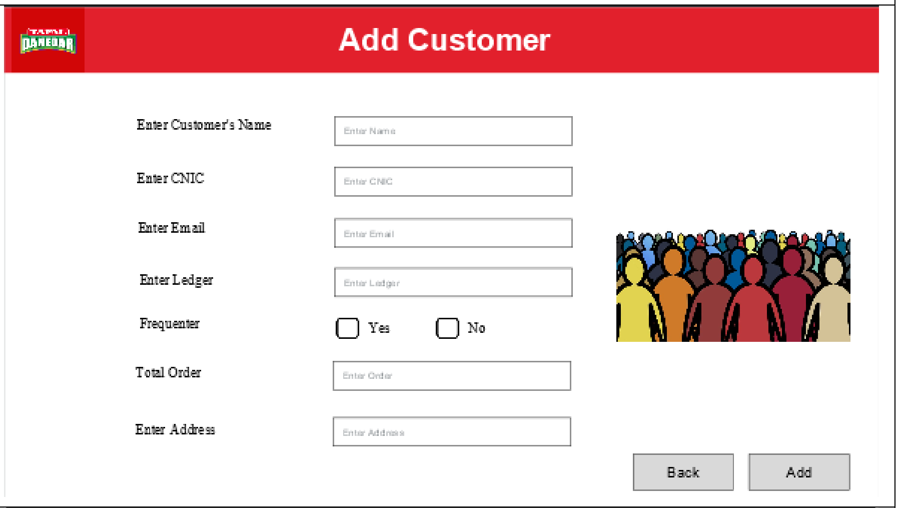
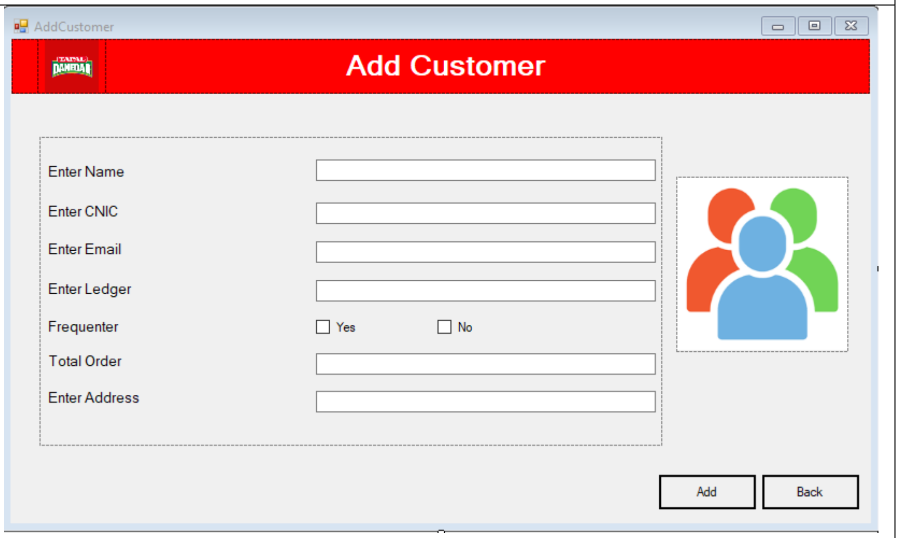
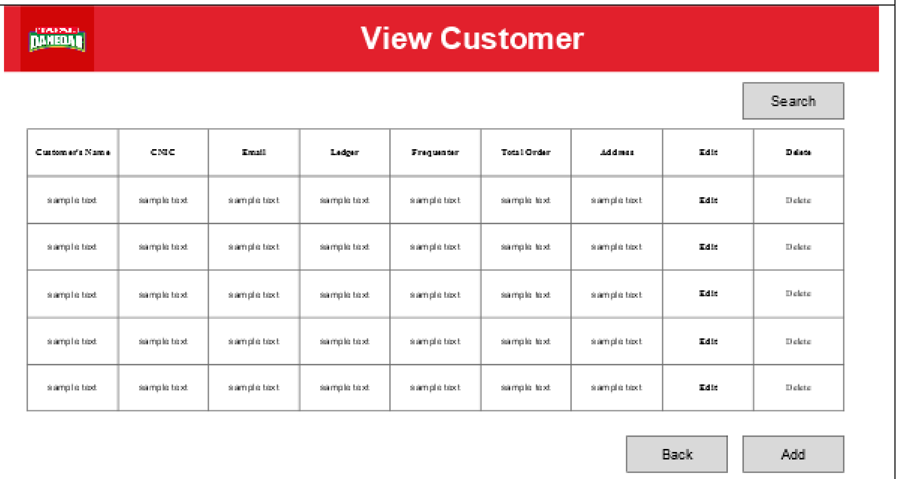
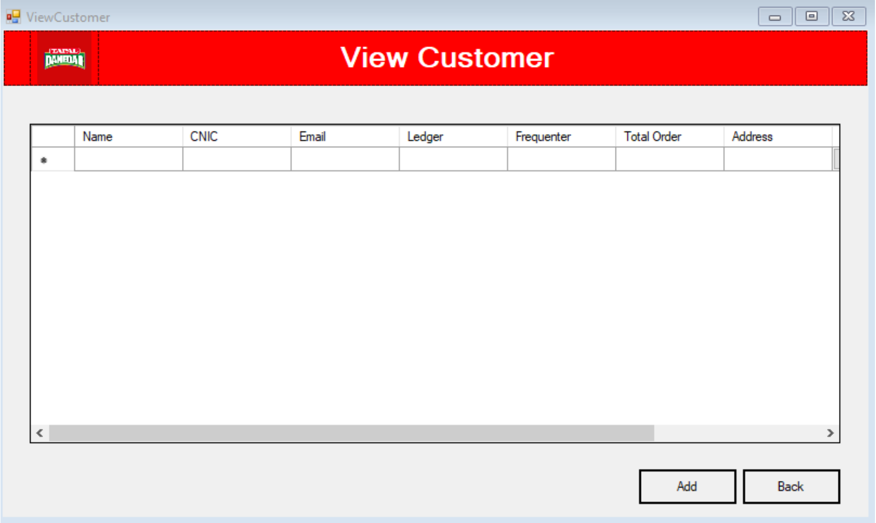
Interface ID	I35
Name	Add Customer
Linked Use Case	U09
UI Screen (JustInMind)	 Add Customer Enter Customer's Name Enter Name Enter CNIC Enter CNIC Enter Email Enter Email Enter Ledger Enter Ledger Frequenter □ Yes □ No Total Order Enter Order Enter Address Enter Address Back Add
UI Screen	 AddCustomer Add Customer Enter Name Enter CNIC Enter Email Enter Ledger Frequenter □ Yes □ No Total Order Enter Address Add Back
Validators	1: Customer cannot be added 2: Add button does not works.

Table 75: I36

<table>
<tr><td>Interface ID</td><td>I36</td></tr>
<tr><td>Name</td><td>View Customer</td></tr>
<tr><td>Linked Use Case</td><td>U11, U13</td></tr>
<tr><td>UI Screen (JustInMind)</td><td>

<table>
<tr><th>Customer's Name</th><th>CNIC</th><th>Email</th><th>Ledger</th><th>Frequenter</th><th>Total Order</th><th>Address</th><th>Edit</th><th>Delete</th></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>Edit</td><td>Delete</td></tr>
</table></td></tr>
<tr><td>UI Screen</td><td>

<table>
<tr><th></th><th>Name</th><th>CNIC</th><th>Email</th><th>Ledger</th><th>Frequenter</th><th>Total Order</th><th>Address</th></tr>
<tr><td>*</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td>Validators</td><td>1: data doesn’t shows in the table.</td></tr>
</table>

Table 76: I37

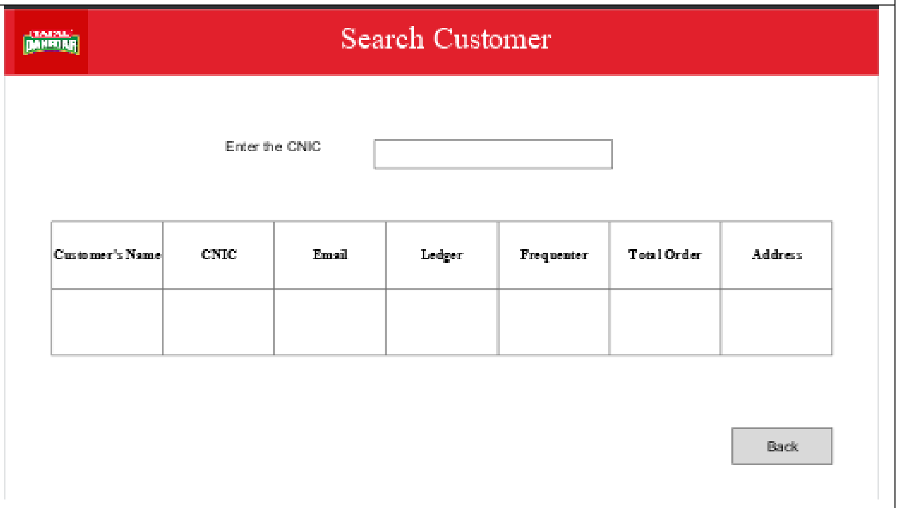
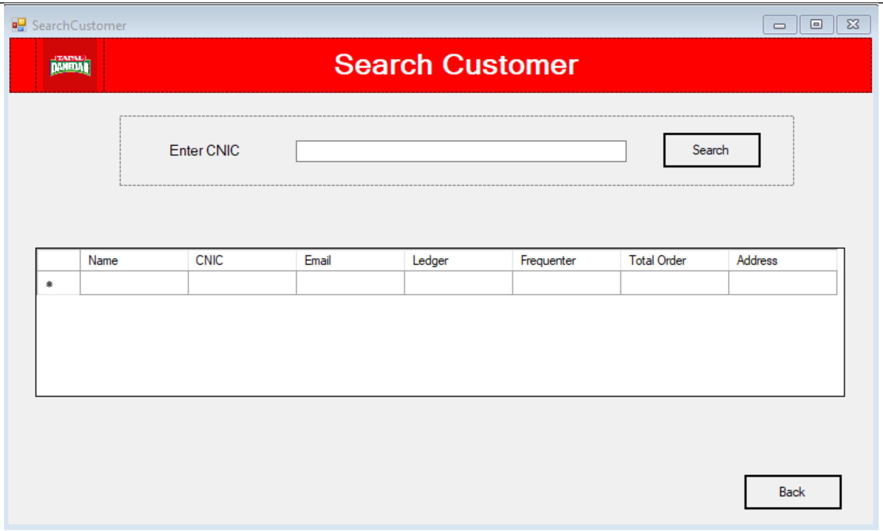
Interface ID	I37
Name	Search Customer
Linked Use Case	U12
UI Screen (JustInMind)	 Search Customer Enter the CNIC Customer's Name CNIC Email Ledger Frequenter Total Order Address Back
UI Screen	 SearchCustomer Search Customer Enter CNIC Search Name CNIC Email Ledger Frequenter Total Order Address Back
Validators	1: Invalid CNIC. 2. Searched customer is not shown.

Table 77: I38

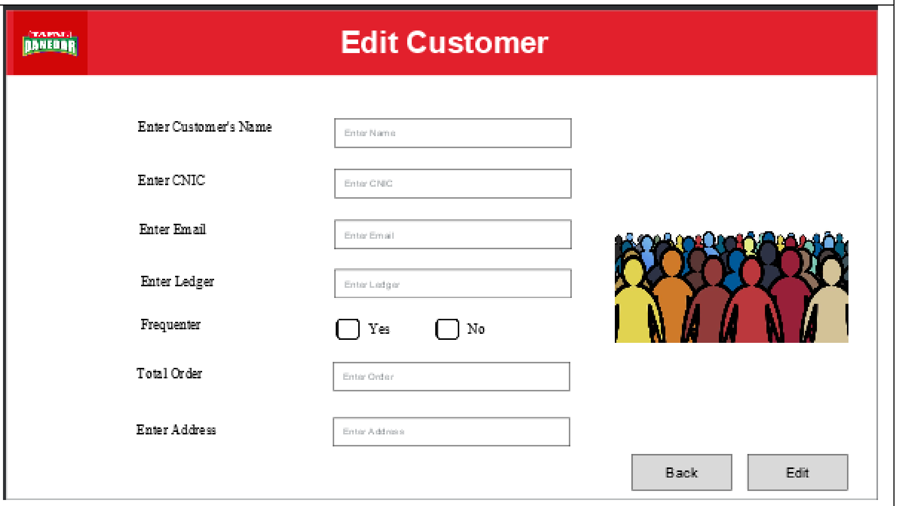
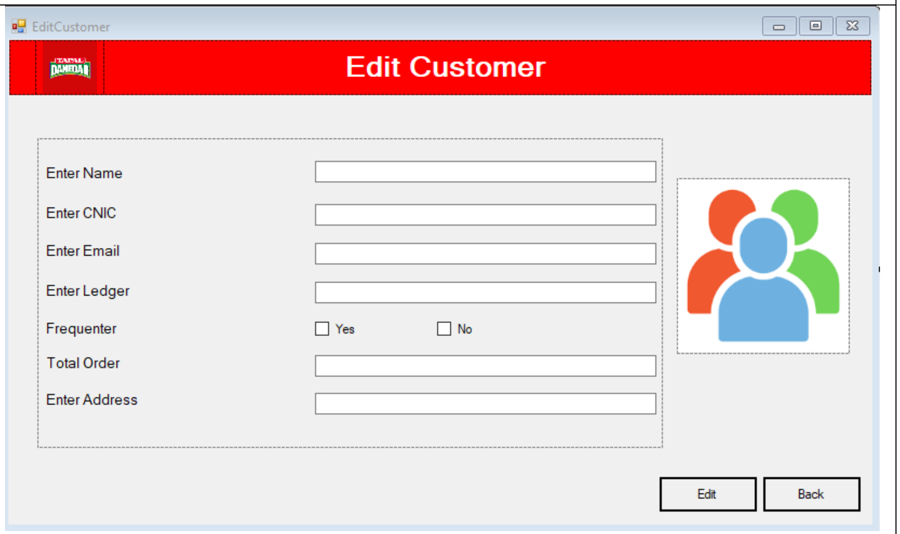
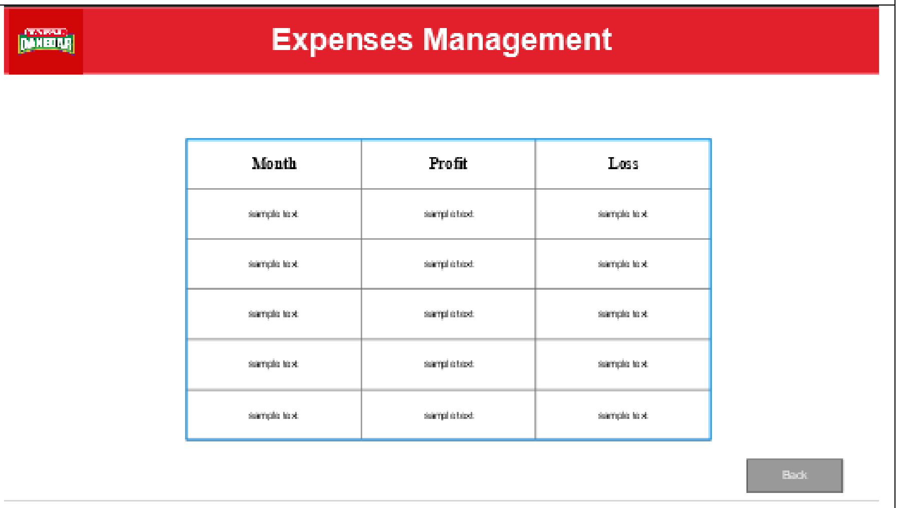
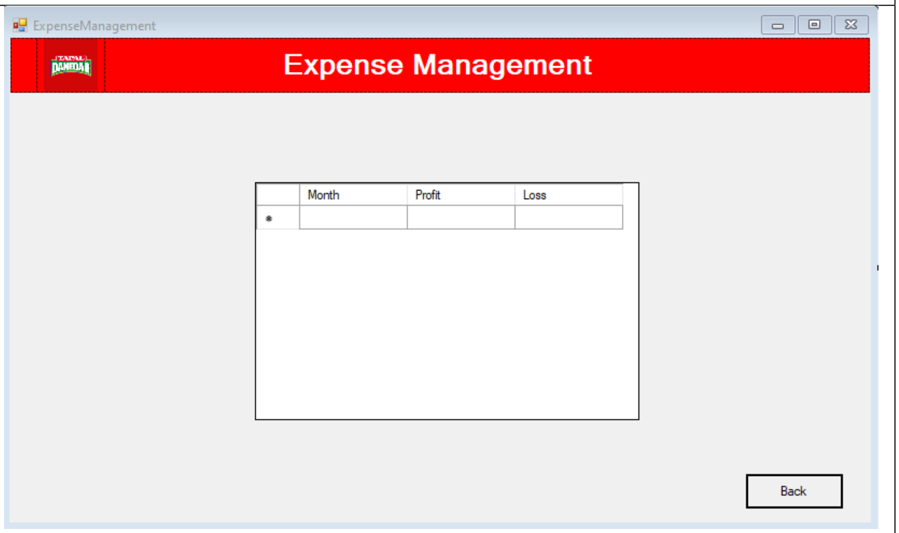
Interface ID	I38
Name	Edit Customer
Linked Use Case	U10
UI Screen (JustInMind)	 Edit Customer Enter Customer's Name: Enter Name Enter CNIC: Enter CNIC Enter Email: Enter Email Enter Ledger: Enter Ledger Frequenter: □ Yes □ No Total Order: Enter Order Enter Address: Enter Address
UI Screen	 EditCustomer Edit Customer Enter Name Enter CNIC Enter Email Enter Ledger Frequenter □ Yes □ No Total Order Enter Address
Validators	1: Customer cannot be edited. 2. Edit button does not change the already present customer.

Table 78: I39

<table>
<tr><td>Interface ID</td><td>I39</td></tr>
<tr><td>Name</td><td>Expenses Management</td></tr>
<tr><td>Linked Use Case</td><td>U29</td></tr>
<tr><td>UI Screen (JustInMind)</td><td>
Expenses Management
<table>
<tr><th>Month</th><th>Profit</th><th>Loss</th></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td></tr>
</table>
</td></tr>
<tr><td>UI Screen</td><td>

Expense Management
<table>
<tr><th></th><th>Month</th><th>Profit</th><th>Loss</th></tr>
<tr><td>*</td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Validators</td><td>1: Data is not shown in the table.</td></tr>
</table>

Table 79: I40

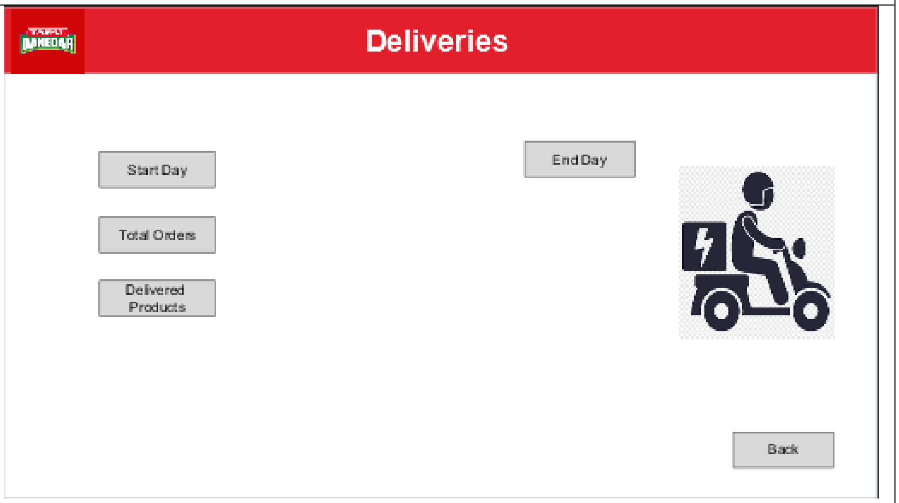
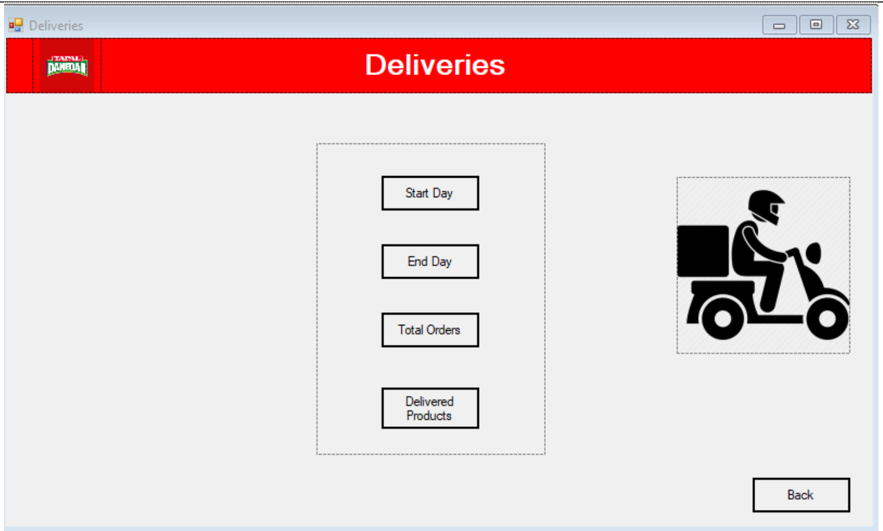
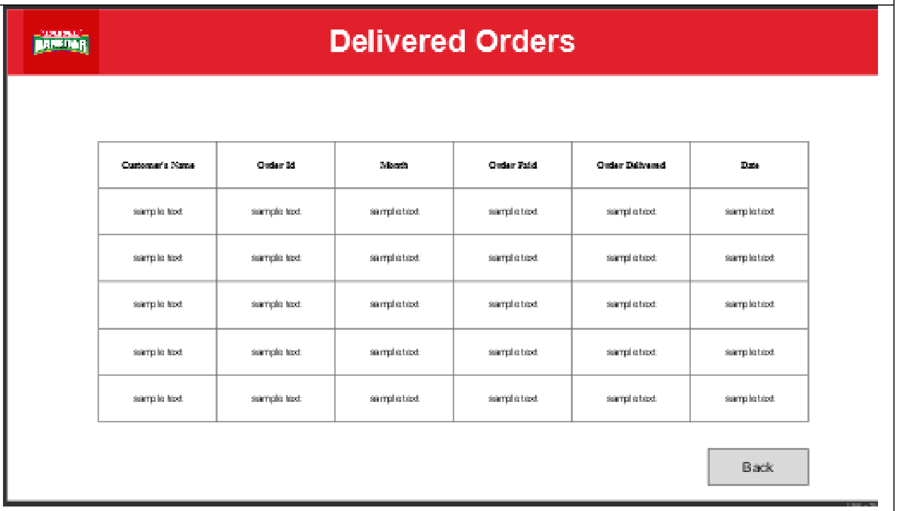
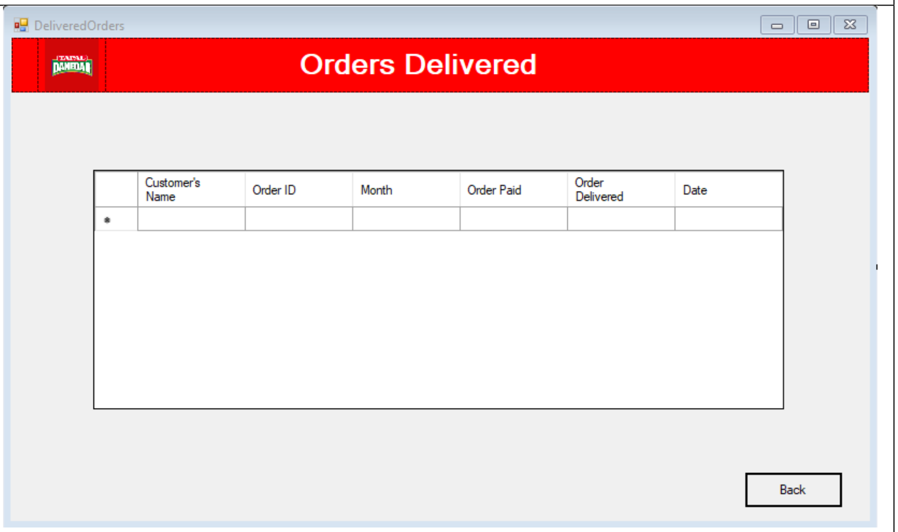
Interface ID	I40
Name	Deliveries
Linked Use Case	U34
UI Screen (JustInMind)	 Deliveries Start Day End Day Total Orders Delivered Products Back
UI Screen	 Deliveries Deliveries Start Day End Day Total Orders Delivered Products Back
Validators	1: Map is not shown by clicking at the start day button.

Table 80: I41

<table>
<tr><td>Interface ID</td><td>I41</td></tr>
<tr><td>Name</td><td>Delivered Orders</td></tr>
<tr><td>Linked Use Case</td><td>U34</td></tr>
<tr><td>UI Screen (JustInMind)</td><td>
Delivered Orders
<table>
<tr><th>Customer's Name</th><th>Order Id</th><th>Month</th><th>Order Paid</th><th>Order Delivered</th><th>Date</th></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td></tr>
<tr><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td><td>sample text</td></tr>
</table>
</td></tr>
<tr><td>UI Screen</td><td>
DeliveredOrders
Orders Delivered
<table>
<tr><th></th><th>Customer's Name</th><th>Order ID</th><th>Month</th><th>Order Paid</th><th>Order Delivered</th><th>Date</th></tr>
<tr><td>*</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Validators</td><td>1: Data is not shown in the table.</td></tr>
</table>

7 Data Structure:

Table 81: Data Structures

Use Case Id	Data Structures Used	Justification for the usage of data structure
U01,U02,U03,U04,U05 U06,U07,U08,U09,U10 U11,U12,U13,U19,U20 U21,U22,U23,U24,U25 U26,U27,U28,U29,U30 U31,U32,U33,U34,U36	Doubly Link List	The data of newly added data are stored in the doubly Linked-List as insertion and deletion takes O(1) , for searching it takes O(n) time.
U14,U15,U16,U17,U18	Queue	The data of newly added orders and already added are stored in the Queue as insertion in O(1)and deletion, searching takes O(n), Also it will follow FIFO policy.

8 Exceptions:

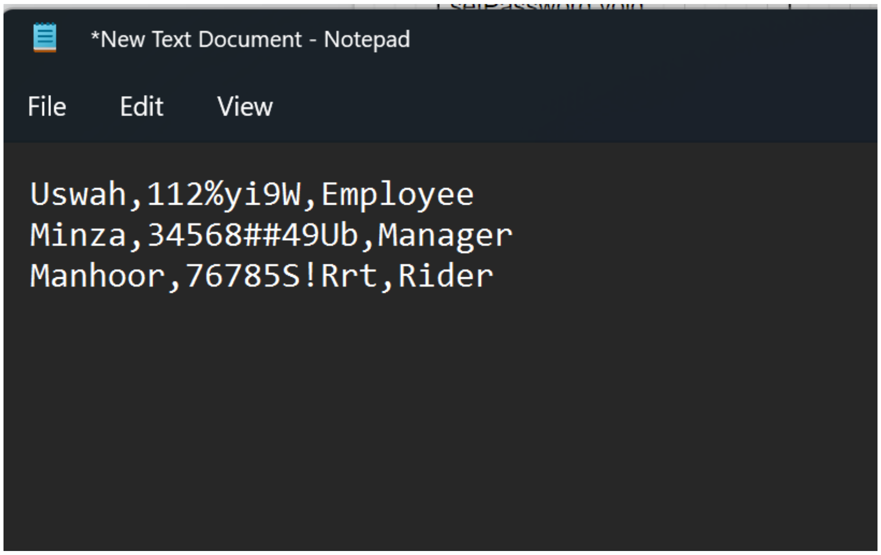
Table 82: Exceptions

Type of Exception	Why this exception will occur	Use Case Id in which exception could be occurred	How you will handle the exception
Invalid User	The Username/Password enter doesn't match the data or the password doesn't match.	U01	Displaying which of both username or password is wrong. If the user the user doesn't remember the password he/she can go to forgot password form.
Repetition	The Id already assigned is being using again.	U03,U04,U05,U09, U10,U14,U15,U19, U20,U24,U25	The data will checked if it already exists, the user can be given a chance to change the Id. There should unique ID in every case.
Unavailability	The order is placed for the product which is not available or maybe there is no rider	U07,U12,U14,U17, U22,U27	The employee has the option to refill the stock and hire all riders
Host Error	The email sent other than the gmail.	U32	The email should be only gmail there will be restriction
Absence	The address entered by the rider is not available on the map.	U35	We will allow user to select from the map or pin it from the map. The map might assign the nearest location.

9 Data Storage:

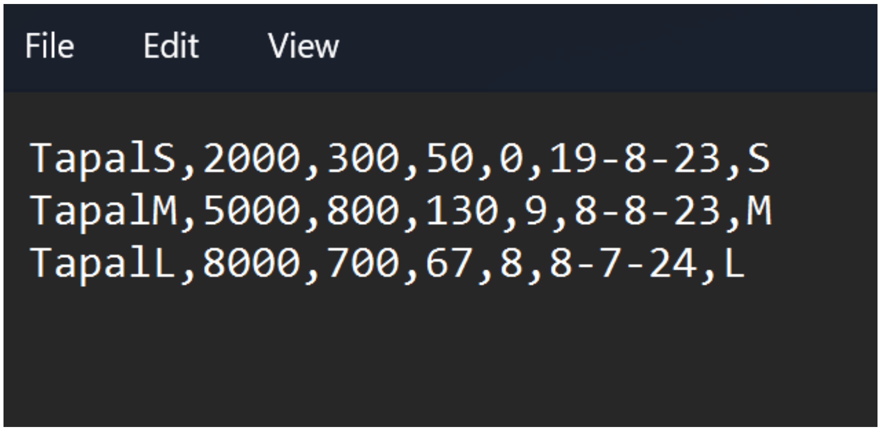
The Data is stored in text file, our employee, order, customer, rider, product that will be stored in different files. User data is saved in a file which contains userName,userPassword and userRole in the end.

Product.txt will have ProductName,ProductStock,ProductThreshold,ProductPrice,ProductTotalSelled,Proc and ProductSize. Customer.txt will have CusName,CusEmail,CusLedger,isFrequenter,CusTotalOrder,CusAc and CusCinic as shown below. Employee.txt will store EmpName, EmpId, EmpPhone , EmpSalary,EmpCinic and isSalaryPaid as shown below. Rider.txt will store EmpName, EmpId, EmpPhone , EmpSalary,EmpCinic ,RiderArea,RiderVechileId and RiderTotalDelivery. Order.txt will have OrderId, mapped CustomerBL names as OrderCustomer, OrderProduct,OrderMonth,OrderP OrderDelivered, mapped OrderRider and OrderDeliveryDate. The data will be loaded from the file using stream reader, stored in the respected lists. If any changes are made the list is rewritten.



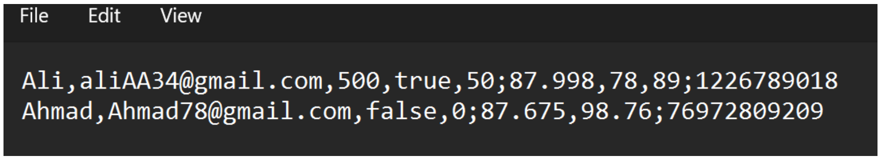
```
Uswah,112%yi9W,Employee
Minza,34568##49Ub,Manager
Manhoor,76785S!Rrt,Rider
```

Figure 1: User



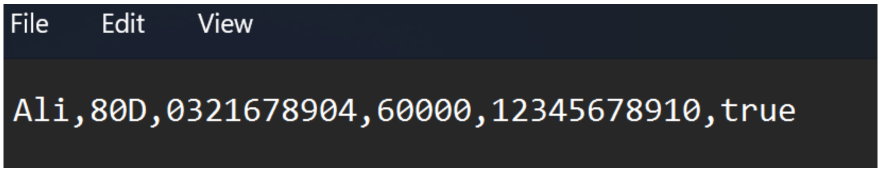
```
TapalS,2000,300,50,0,19-8-23,S
TapalM,5000,800,130,9,8-8-23,M
TapalL,8000,700,67,8,8-7-24,L
```

Figure 2: product



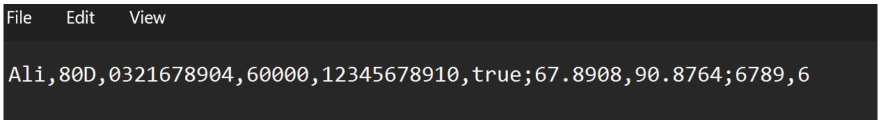
```
Ali,aliAA34@gmail.com,500,true,50;87.998,78,89;1226789018
Ahmad,Ahmad78@gmail.com,false,0;87.675,98.76;76972809209
```

Figure 3: Customer



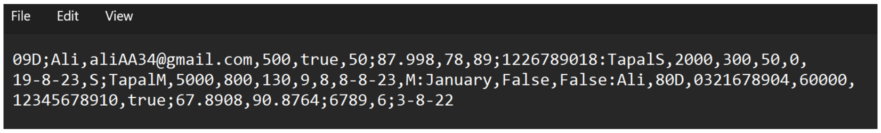
```
Ali,80D,0321678904,60000,12345678910,true
```

Figure 4: Employee



```
Ali,80D,0321678904,60000,12345678910,true;67.8908,90.8764;6789,6
```

Figure 5: Rider



```
09D;Ali,aliAA34@gmail.com,500,true,50;87.998,78,89;1226789018:TapalS,2000,300,50,0,
19-8-23,S;TapalM,5000,800,130,9,8,8-8-23,M:January,False,False:Ali,80D,0321678904,60000,
12345678910,true;67.8908,90.8764;6789,6;3-8-22
```

Figure 6: Order

10 Email Sending:

The Email will be send to customers for “total amounts of order”.
Sample:
Subject: ˜Your Order has been placed

Hello Customer,
Thank you for your order. We appreciate your business and will be thrilled to send you [Products Ordered] as soon as possible. An email with tracking information will be sent to you once your order has shipped. Your total bill is 70,000.
Thanks again, and we look forward to seeing you soon.
UMM Distribution Company

11 Object Oriented Features:

Composition:

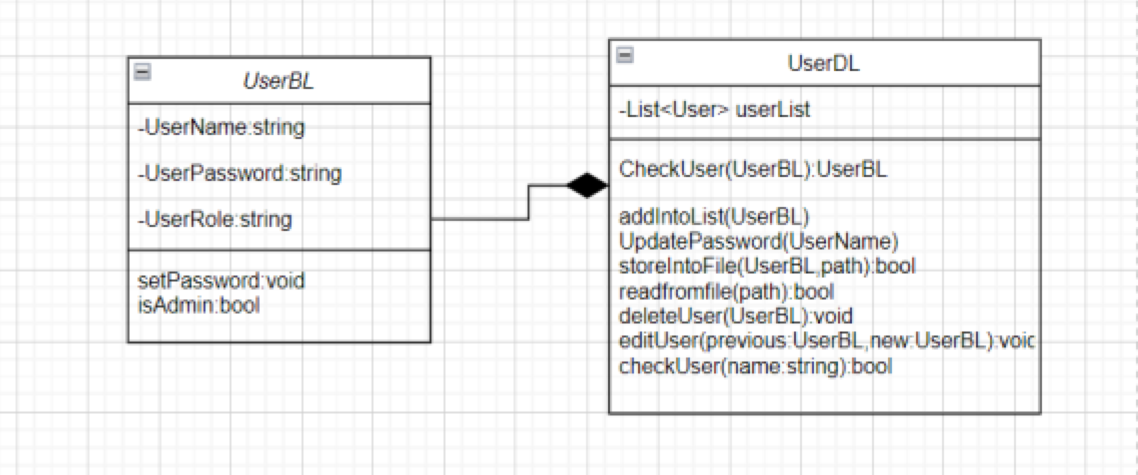


Figure 9.1

LikeWise UserBL and UserDL there are several examples are there such as ProductDL , RiderDL,

EmployeeDL and CustomerDL acting similar to UserDL handling data logics for ProductBL, RiderBL
EmployeeBL and CustomerBL respectively.

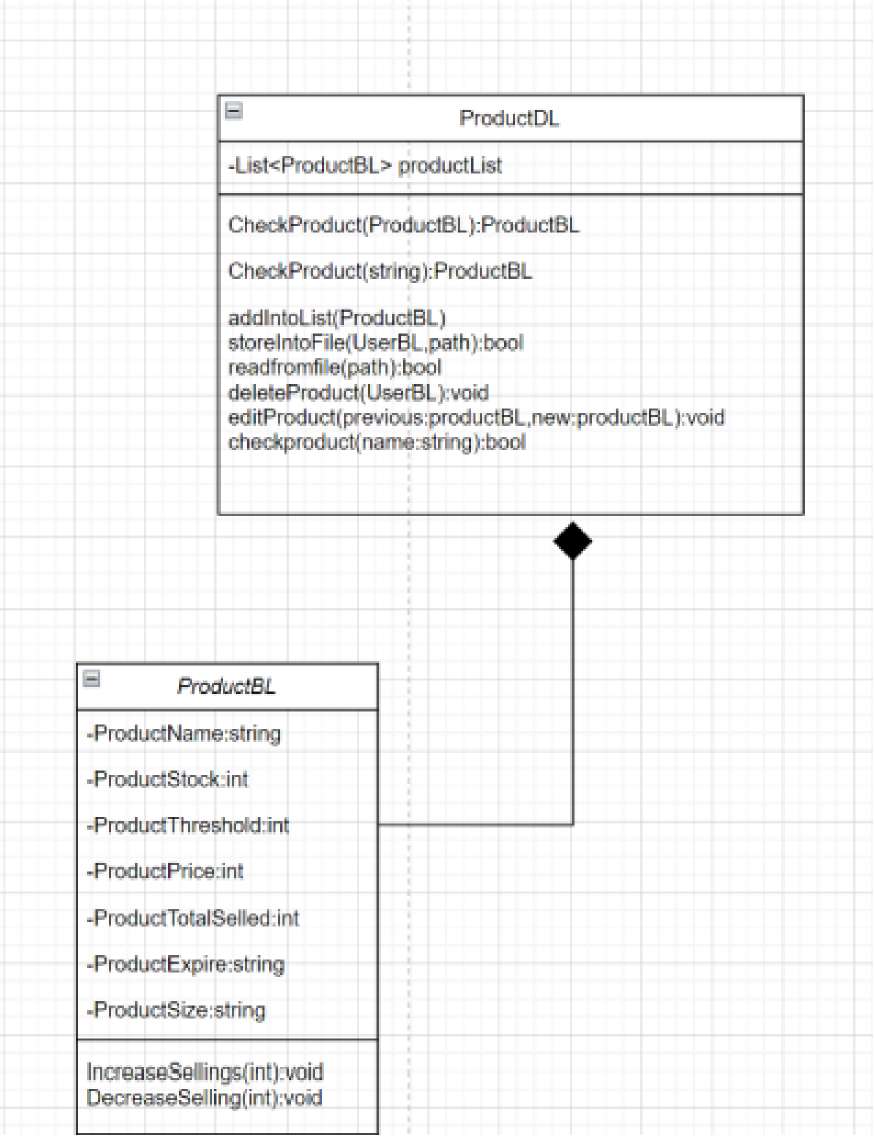

ProductDL
-List<ProductBL> productList
CheckProduct(ProductBL):ProductBL
CheckProduct(string):ProductBL
addIntoList(ProductBL)
storeIntoFile(UserBL,path):bool
readfromfile(path):bool
deleteProduct(UserBL):void
editProduct(previous:productBL,new:productBL):void
checkproduct(name:string):bool
ProductBL
-ProductName:string
-ProductStock:int
-ProductThreshold:int
-ProductPrice:int
-ProductTotalSelled:int
-ProductExpire:string
-ProductSize:string
IncreaseSellings(int):void
DecreaseSelling(int):void

Figure 9.2

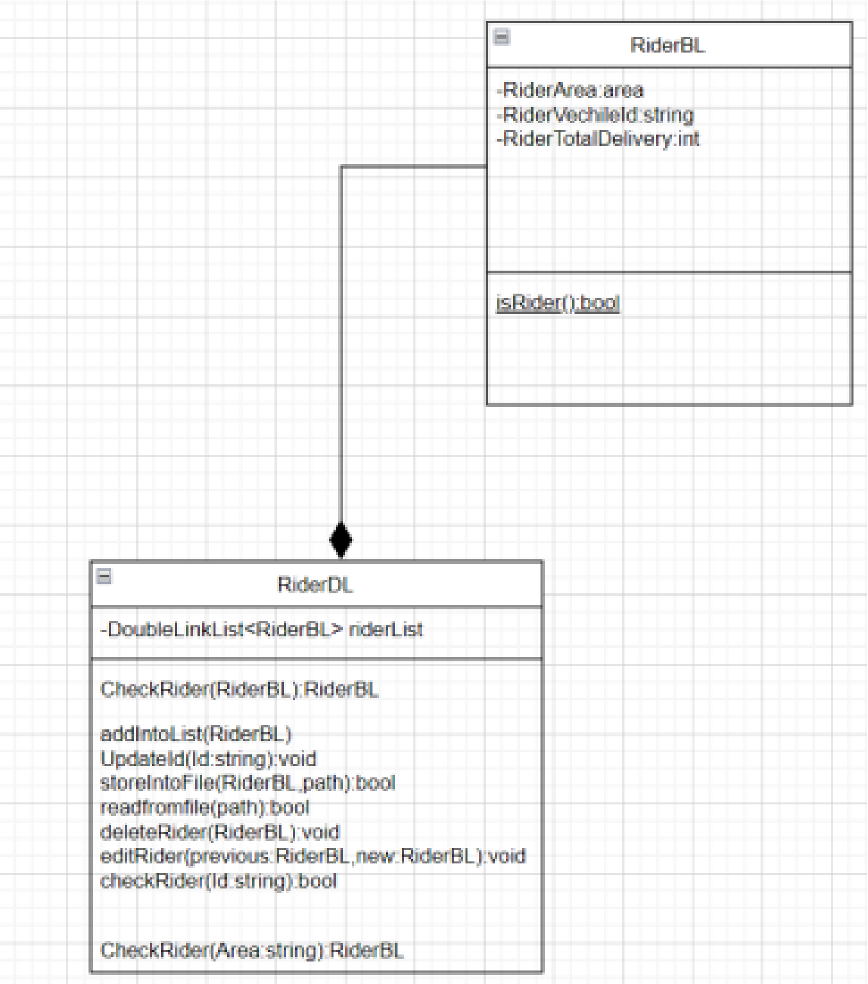

RiderBL
-RiderArea:area
-RiderVechileId:string
-RiderTotalDelivery:int
isRider():bool
RiderDL
-DoubleLinkList<RiderBL> riderList
CheckRider(RiderBL):RiderBL
addIntoList(RiderBL)
UpdateId(Id:string):void
storeIntoFile(RiderBL,path):bool
readfromfile(path):bool
deleteRider(RiderBL):void
editRider(previous:RiderBL,new:RiderBL):void
checkRider(Id:string):bool
CheckRider(Area:string):RiderBL

Figure 9.3

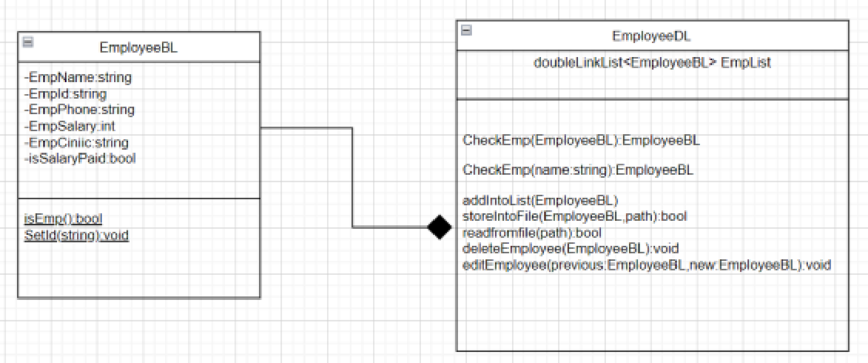

EmployeeBL
-EmpName:string
-EmpId:string
-EmpPhone:string
-EmpSalary:int
-EmpCiniic:string
-isSalaryPaid:bool
isEmp():bool
SetId(string):void
EmployeeDL
doubleLinkList<EmployeeBL> EmpList
CheckEmp(EmployeeBL):EmployeeBL
CheckEmp(name:string):EmployeeBL
addIntoList(EmployeeBL)
storeIntoFile(EmployeeBL,path):bool
readfromfile(path):bool
deleteEmployee(EmployeeBL):void
editEmployee(previous:EmployeeBL,new:EmployeeBL):void

Figure 9.4

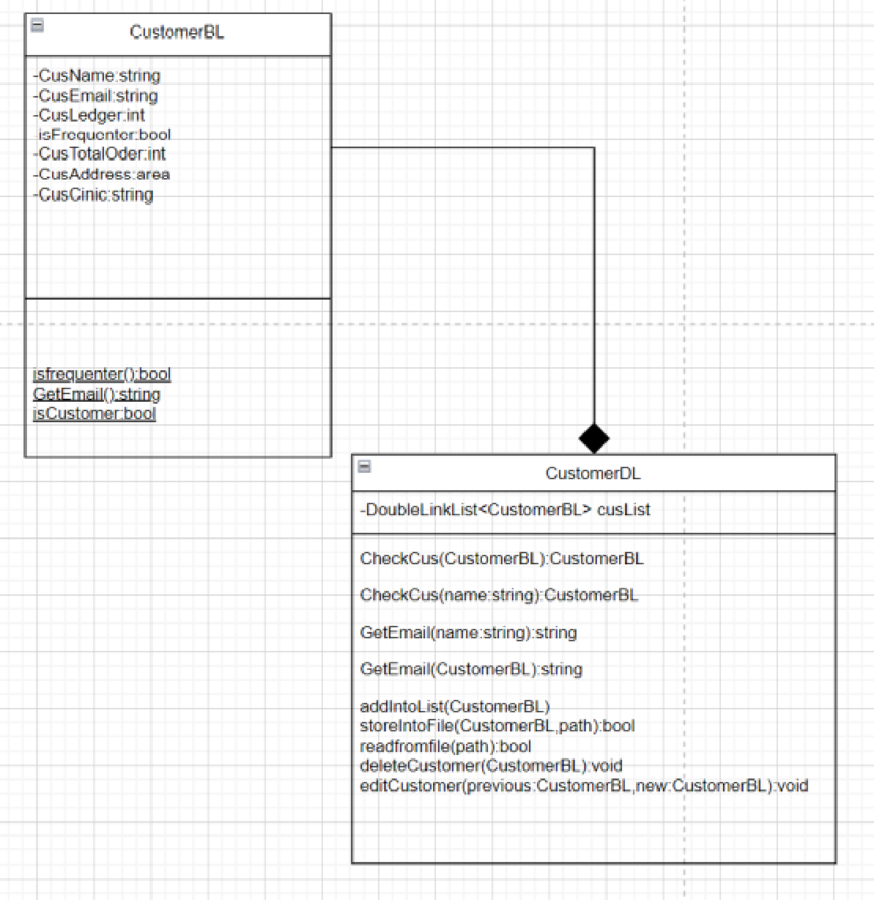

CustomerBL
-CusName:string
-CusEmail:string
-CusLedger:int
isFrequenter:bool
-CusTotalOder:int
-CusAddress:area
-CusCinic:string
isfrequenter():bool
GetEmail():string
isCustomer:bool
CustomerDL
-DoubleLinkList<CustomerBL> cusList
CheckCus(CustomerBL):CustomerBL
CheckCus(name:string):CustomerBL
GetEmail(name:string):string
GetEmail(CustomerBL):string
addIntoList(CustomerBL)
storeIntoFile(CustomerBL,path):bool
readfromfile(path):bool
deleteCustomer(CustomerBL):void
editCustomer(previous:CustomerBL,new:CustomerBL):void

Figure 9.5

Likewise, explained before OrderBL is composited with OrderDL.

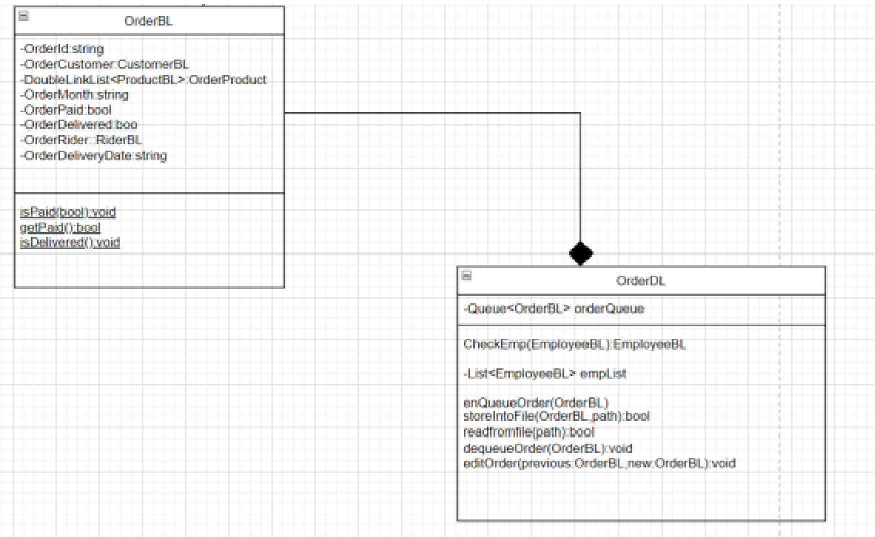

OrderBL
-OrderId:string
-OrderCustomer:CustomerBL
-DoubleLinkList<ProductBL>:OrderProduct
-OrderMonth:string
-OrderPaid:bool
-OrderDelivered:boo
-OrderRider: RiderBL
-OrderDeliveryDate:string
isPaid(bool):void
getPaid():bool
isDelivered():void
OrderDL
-Queue<OrderBL> orderQueue
CheckEmp(EmployeeBL):EmployeeBL
-List<EmployeeBL> empList
enQueueOrder(OrderBL)
storeIntoFile(OrderBL,path):bool
readfromfile(path):bool
dequeueOrder(OrderBL):void
editOrder(previous:OrderBL,new:OrderBL):void

Figure 9.6

12 Inheritance:

The rider is also an employee but it has some extra features so the RiderBL, class having information
about Rider inherits the EmployeeBL.

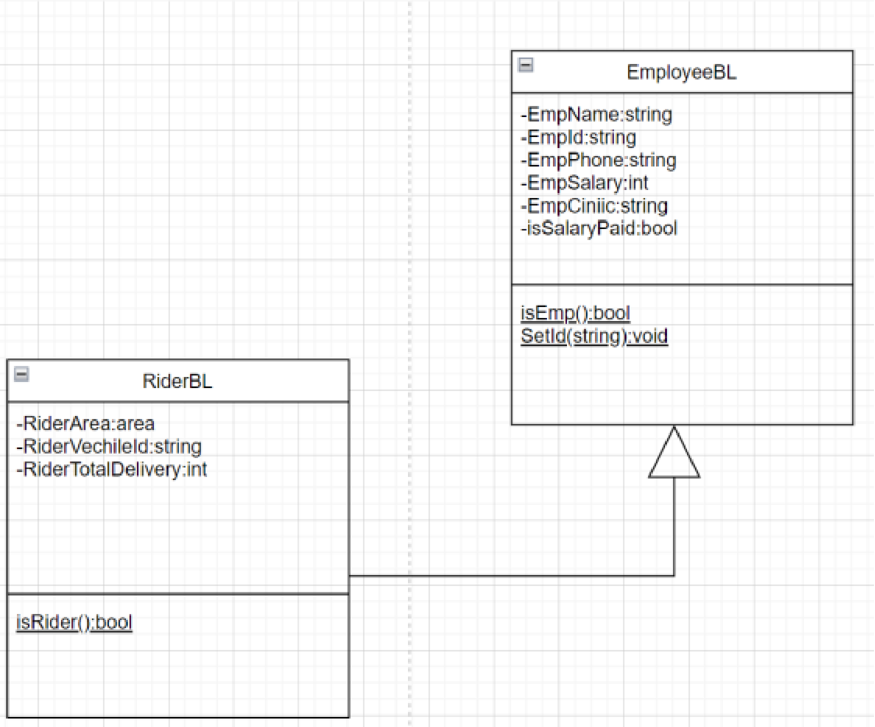


Figure 10.1

13 Polymorphism:

The class having function CheckEmpty which can check the Employee Existence function by the object
or through its name both the function have same return type.

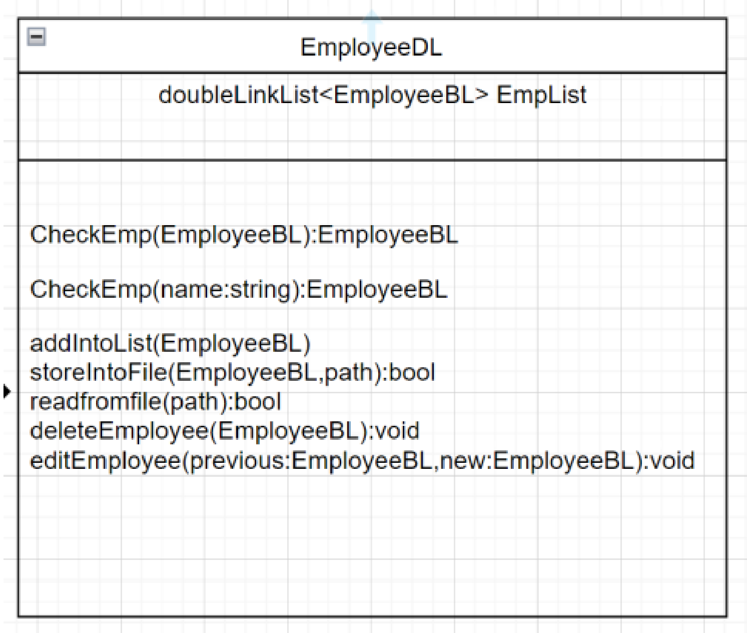

EmployeeDL
doubleLinkList<EmployeeBL> EmpList
CheckEmp(EmployeeBL):EmployeeBL
CheckEmp(name:string):EmployeeBL
addIntoList(EmployeeBL)
storeIntoFile(EmployeeBL,path):bool
readfromfile(path):bool
deleteEmployee(EmployeeBL):void
editEmployee(previous:EmployeeBL,new:EmployeeBL):void

Figure 11.1

Similarly the same case is in ProductDL searching product by its name or object through function
CheckProduct.

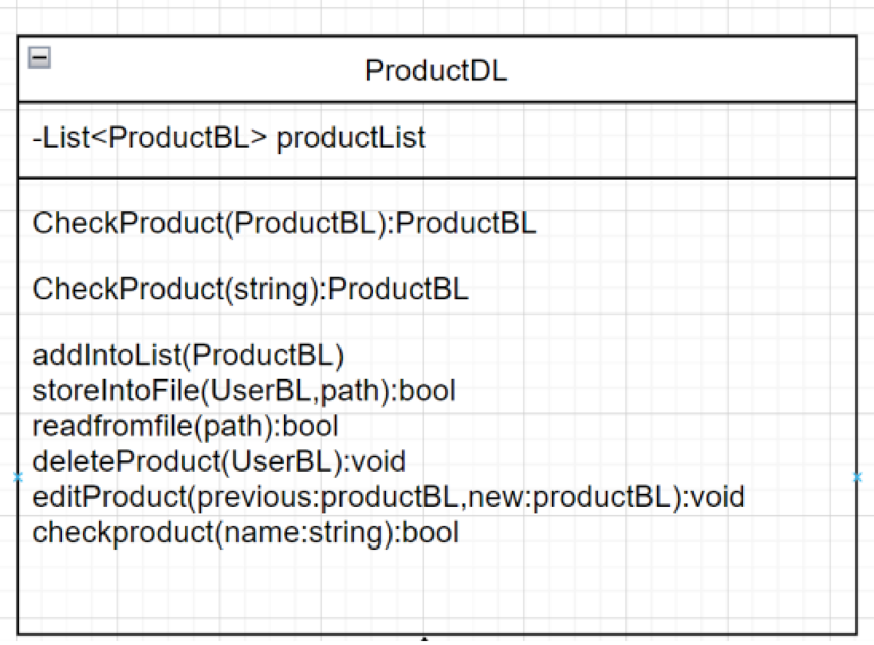

ProductDL
-List<ProductBL> productList
CheckProduct(ProductBL):ProductBL
CheckProduct(string):ProductBL
addIntoList(ProductBL)
storeIntoFile(UserBL,path):bool
readfromfile(path):bool
deleteProduct(UserBL):void
editProduct(previous:productBL,new:productBL):void
checkproduct(name:string):bool

Figure 11.2

The CustomerDL have the similar searching option but also GetEmail which can be used in the forms
to access the email, it can be access by the name of customer which is string, also through CustomerBL

object.

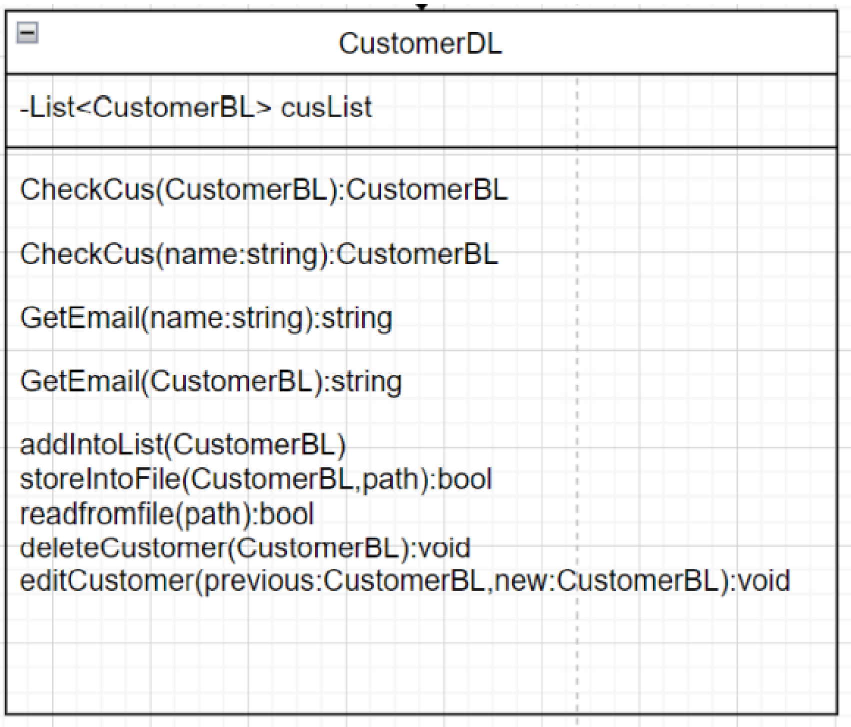

CustomerDL
-List<CustomerBL> cusList
CheckCus(CustomerBL):CustomerBL
CheckCus(name:string):CustomerBL
GetEmail(name:string):string
GetEmail(CustomerBL):string
addIntoList(CustomerBL)
storeIntoFile(CustomerBL,path):bool
readfromfile(path):bool
deleteCustomer(CustomerBL):void
editCustomer(previous:CustomerBL,new:CustomerBL):void

Figure 11.3

14 Detailed Object Oriented Design:

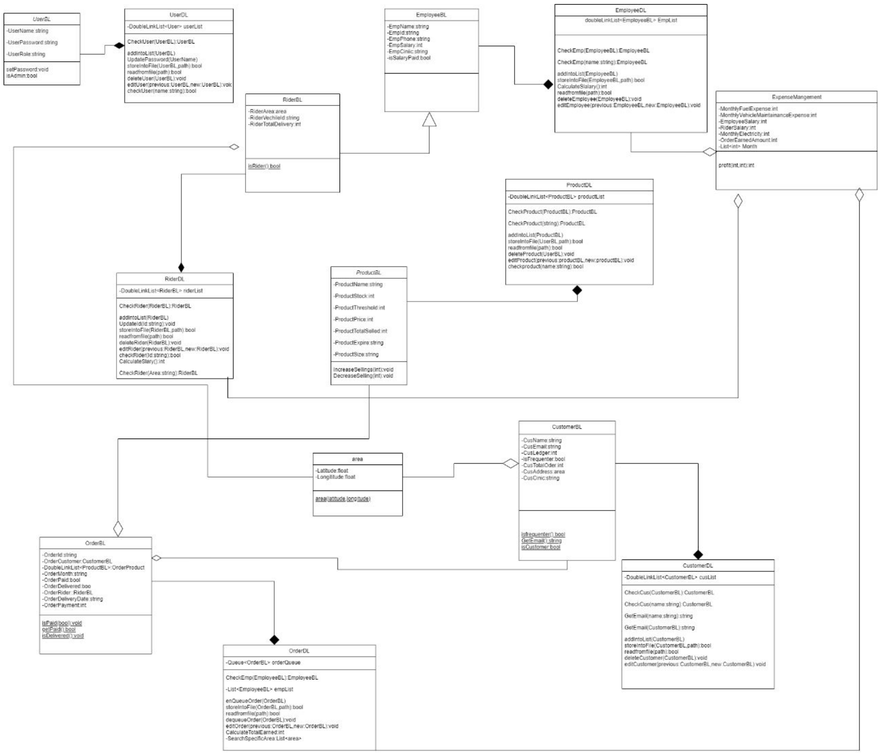

UserBL
-UserName:string
-UserPassword:string
-UserRole:string
setPassword:void
isAdmin:bool
UserDL
-DoubleLinkList<User> userList
CheckUser(UserBL):UserBL
addIntoList(UserBL)
UpdatePassword(UserName)
storeIntoFile(UserBL,path):bool
readfromfile(path):bool
deleteUser(UserBL):void
editUser(previous:UserBL,new:UserBL):void
checkUser(name:string):bool
EmployeeBL
-EmpName:string
-EmpId:string
-EmpPhone:string
-EmpSalary:int
-EmpCiniic:string
-isSalaryPaid:bool
EmployeeDL
doubleLinkList<EmployeeBL> EmpList
CheckEmp(EmployeeBL):EmployeeBL
CheckEmp(name:string):EmployeeBL
addIntoList(EmployeeBL)
storeIntoFile(EmployeeBL,path):bool
CalculateSlalary():int
readfromfile(path):bool
deleteEmployee(EmployeeBL):void
editEmployee(previous:EmployeeBL,new:EmployeeBL):void
RiderBL
-RiderArea:area
-RiderVechileId:string
-RiderTotalDelivery:int
isRider():bool
ExpenseMangement
-MonthlyFuelExpense:int
-MonthlyVehicleMaintainanceExpense:int
-EmployeeSalary:int
-RiderSalary:int
-MonthlyElectricity:int
-OrderEarnedAmount:int
-List<int> Month
profit(int,int):int
ProductDL
-DoubleLinkList<ProductBL> productList
CheckProduct(ProductBL):ProductBL
CheckProduct(string):ProductBL
addIntoList(ProductBL)
storeIntoFile(UserBL,path):bool
readfromfile(path):bool
deleteProduct(UserBL):void
editProduct(previous:productBL,new:productBL):void
checkproduct(name:string):bool
RiderDL
-DoubleLinkList<RiderBL> riderList
CheckRider(RiderBL):RiderBL
addIntoList(RiderBL)
UpdateId(Id:string):void
storeIntoFile(RiderBL,path):bool
readfromfile(path):bool
deleteRider(RiderBL):void
editRider(previous:RiderBL,new:RiderBL):void
checkRider(Id:string):bool
CalculateSlary():int
CheckRider(Area:string):RiderBL
ProductBL
-ProductName:string
-ProductStock:int
-ProductThreshold:int
-ProductPrice:int
-ProductTotalSelled:int
-ProductExpire:string
-ProductSize:string
IncreaseSellings(int):void
DecreaseSelling(int):void
CustomerBL
-CusName:string
-CusEmail:string
-CusLedger:int
-isFrequenter:bool
-CusTotalOder:int
-CusAddress:area
-CusCinic:string
isfrequenter():bool
GetEmail():string
isCustomer:bool
area
-Latitude:float
-Longitude:float
area(latitude,longitude)
OrderBL
-OrderId:string
-OrderCustomer:CustomerBL
-DoubleLinkList<ProductBL>:OrderProduct
-OrderMonth:string
-OrderPaid:bool
-OrderDelivered:boo
-OrderRider::RiderBL
-OrderDeliveryDate:string
-OrderPayment:int
isPaid(bool):void
getPaid():bool
isDelivered():void
CustomerDL
-DoubleLinkList<CustomerBL> cusList
CheckCus(CustomerBL):CustomerBL
CheckCus(name:string):CustomerBL
GetEmail(name:string):string
GetEmail(CustomerBL):string
addIntoList(CustomerBL)
storeIntoFile(CustomerBL,path):bool
readfromfile(path):bool
deleteCustomer(CustomerBL):void
editCustomer(previous:CustomerBL,new:CustomerBL):void
OrderDL
-Queue<OrderBL> orderQueue
CheckEmp(EmployeeBL):EmployeeBL
-List<EmployeeBL> empList
enQueueOrder(OrderBL)
storeIntoFile(OrderBL,path):bool
readfromfile(path):bool
dequeueOrder(OrderBL):void
editOrder(previous:OrderBL,new:OrderBL):void
CalculateTotalEarned:int
-SearchSpecificArea:List<area>

Figure 12.1

15 OOP Status Table

Table 83: OOP Status Table

Class name	Status	Reason
UserBL	same	
UserDL	same	
RiderBL	same	
EmployeeBL	updated	isEmp and selled function are not needed.
EmployeeDL	same	
ExpenseManagement	same	
RiderDL	same	
ProductBL	same	
ProductDL	same	
Area	same	
OrderBL	same	
OrderDL	same	
CustomerBL	same	
CustomerDL	same	

16 Analytical Reports:

We will show the manager the success of selled products i.e; which product sold highest, also the profit earned within each month, the riders progress showing number of deliveries done by the rider. This all data will be maintained side by side while any change is made, the data will be shown in the graphs Line chart, bar chart and pie chart the user can choose which type of graph he wants.

The data can be sorted providing number of sales highest from which frequenter, allows the company to make strong relations with that retailer and where there are less sales to identify the reason and improve it. The data works in best interest of the workers.

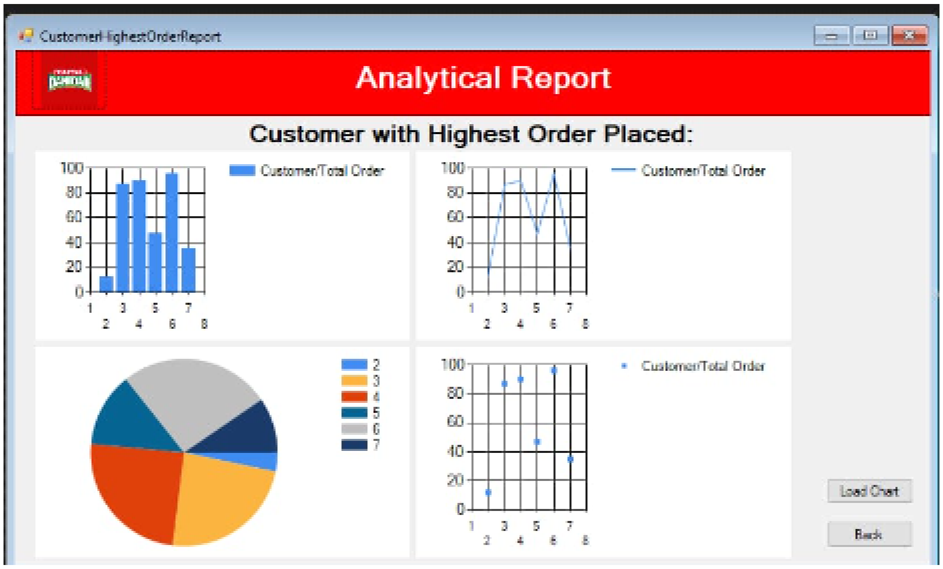


Figure 17.1

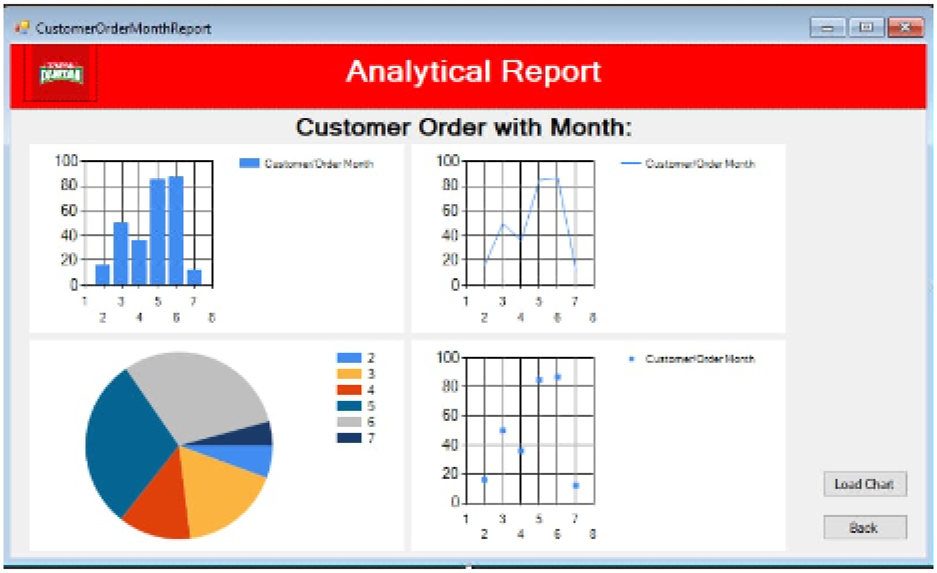
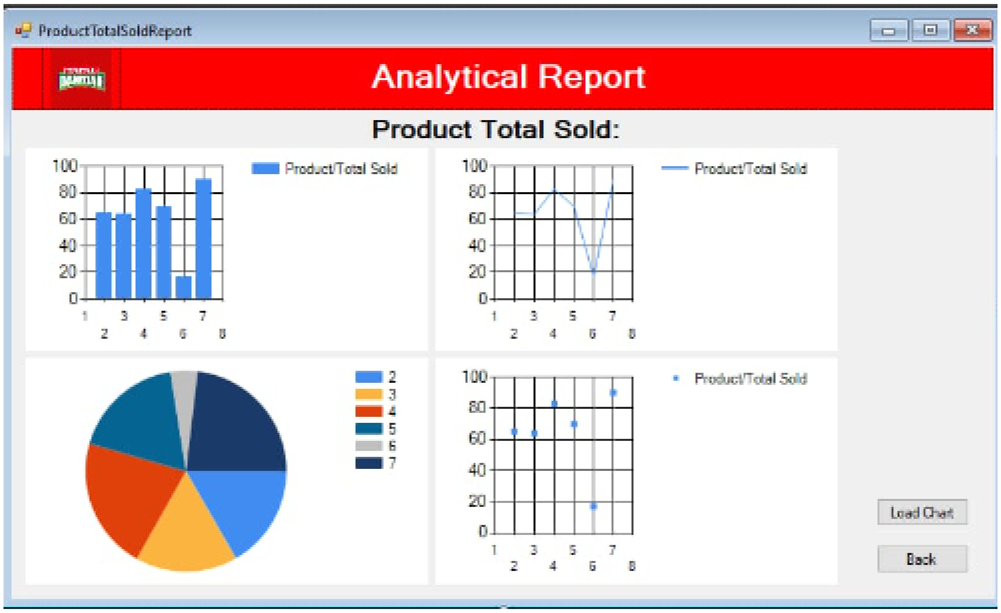


Figure 17.2



ProductTotalSoldReport

Analytical Report

Product Total Sold:

17 New Project Plan

Table 84: Project Plan

Use Case Id	Use Case Name	Member Name	Estimated Date	Completion Date
U01	SignIn page	Mahnoor Hassan	7-12-2022	14-12-2022
U02	Create Account	Mahnoor Hassan	7-12-2022	14-12-2022
U03	Reset password	Mahnoor Hassan	7-12-2022	14-12-2022
U04	Add Rider	Uswa Arif	7-12-2022	14-12-2022
U05	Edit Rider	Uswa Arif	7-12-2022	14-12-2022
U06	Remove Rider	Uswa Arif	7-12-2022	14-12-2022
U07	Search Rider	Uswa Arif	7-12-2022	14-12-2022
U08	View Rider	Uswa Arif	7-12-2022	14-12-2022
U09	Add Customer	Mutaiba Mohsin	7-12-2022	14-12-2022
U10	Edit customer	Mutaiba Mohsin	7-12-2022	14-12-2022
U11	Remove Customer	Mutaiba Mohsin	7-12-2022	14-12-2022
U12	Search Customer	Mutaiba Mohsin	7-12-2022	14-12-2022
U13	View Customer	Mutaiba Mohsin	7-12-2022	14-12-2022
U19	Add Employee	Mahnoor Hassan	8-12-2022	15-12-2022
U20	Edit Employee	Mahnoor Hassan	8-12-2022	15-12-2022
U21	Remove Employee	Mahnoor Hassan	8-12-2022	15-12-2022
U22	Search Employee	Mahnoor Hassan	8-12-2022	15-12-2022
U23	View Employee	Mahnoor Hassan	8-12-2022	15-12-2022
U14	Add order	Mutaiba Mohsin	8-12-2022	15-12-2022
U15	Edit Order	Mutaiba Mohsin	8-12-2022	15-12-2022
U16	Remove Order	Mutaiba Mohsin	8-12-2022	15-12-2022
U17	Search Order	Mutaiba Mohsin	8-12-2022	15-12-2022
U18	View Order	Mutaiba Mohsin	8-12-2022	15-12-2022
U24	Add Product	Mutaiba Mohsin	9-12-2022	16-12-2022
U25	Edit Product	Mutaiba Mohsin	9-12-2022	16-12-2022
U26	Remove Product	Mutaiba Mohsin	9-12-2022	16-12-2022
U27	Search Product	Mutaiba Mohsin	9-12-2022	16-12-2022
U28	View Product	Mutaiba Mohsin	9-12-2022	16-12-2022
U29	Manage Expenses	Uswa Arif	11-12-2022	18-12-2022
U30	Give Salary to riders	Uswa Arif	12-12-2022	19-12-2022
U31	Give Salary to employees	Uswa Arif	12-12-2022	19-12-2022
U32	Send Email	Uswa Arif	14-12-2022	11-12-2022
U33	Make Graph	Mutaiba Mohsin	14-12-2022	20-12-2022
U34	Deliver Ordered Products	Uswa Arif	15-12-2022	21-12-2022
U35	Show location of customers to deliver Products	Uswa Arif	17-12-2022	21-12-2022
U36	Update payment of customers	Mahnoor Hassan	13-12-2022	21-12-2022
U37	Designing UI	Mahnoor Hassan	11-12-2022	12-12-2022
	Report	Mahnoor Hassan	19-12-2022	22-12-2022