

# Fast n' Delight Cafe

## Phase 1

## Team 2

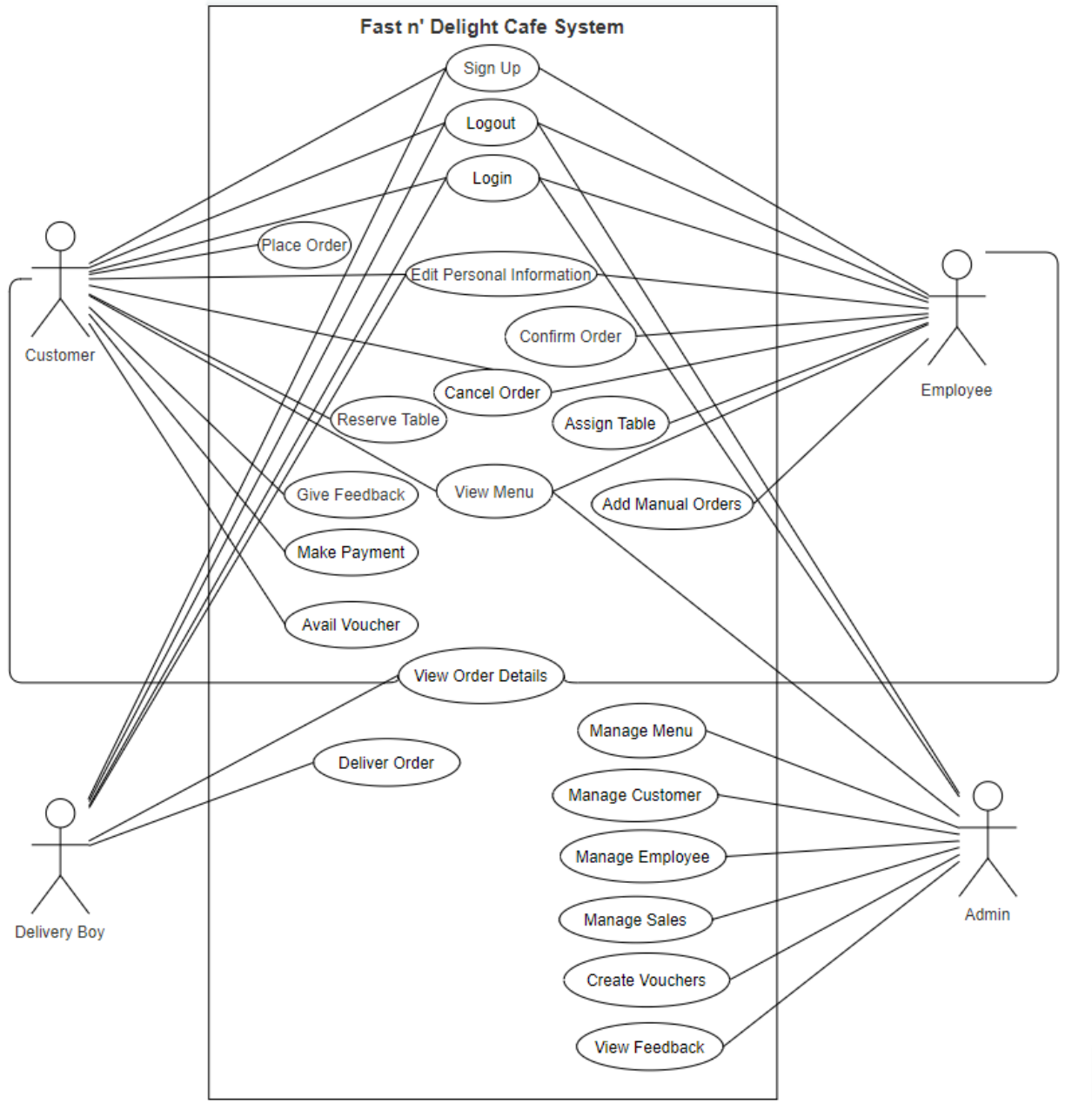
Member Name	Member Roll #	Primary Responsibility
Usman Faisal	20L-1385	ACD, UC-1,2,3,4,5,6,13,16,17
Fizza Sajjad	20L-1389	ACD, UC-7,8,9
Noor Fatima	20L-0993	ACD, UC-10,11,12
Abdur Rafay	20L-1391	UC-14
Fiza Hassan	20L-1279	UC-15,18
Hassan Murtaza	20K-1641	UC-19,20
Hira Touqeer	20L-1375	UC-21,22

# Table of Contents

<b>Use Case Diagram</b>	<b>2</b>
<b>Analysis Level Class Diagram</b>	<b>3</b>
<b>Use Cases</b>	<b>4</b>
Use Case 1	4
Use Case 2	6
Use Case 3	7
Use Case 4	8
Use Case 5	9
Use Case 6	10
Use Case 7	12
Use Case 8	13
Use Case 9	15
Use Case 10	16
Use Case 11	17
Use Case 12	18
Use Case 13	19
Use Case 14	20
Use Case 15	22
Use Case 16	23
Use Case 17	25
Use Case 18	26
Use Case 19	27
Use Case 20	29
Use Case 21	30
Use Case 22	31

# 1. Use Case Diagram

\*Every use case includes that the actors must have logged into the system.



## **2. Analysis Level Class Diagram**

[Link for Class Diagram](#)

### 3. Use Cases

#### Use Case 1

Identifier		UC-1
Name		Sign Up
Summary		The user can sign up using their personal information
Priority		High
Actors		Customer, Employee, Delivery Boy
Pre-condition(s)		None
Post-condition(s)		User accounts are created in database
Typical Course of Action		
S#	Actor Action	System Response
1	Selects Sign Up option	
2		Displays various field boxes such as (First Name, Last Name, Phone Number, Email ID, Password)
3	Enters their information	
4		Displays message "Account successfully created"
Alternate Course of Action (Text field left empty)		
S#	Actor Action	System Response

<b>4</b>		Displays an error "Cannot sign up as the field was left empty"
Go to 2		

### Use Case 2

Identifier	UC-2	
Name	Login	
Summary	The user can login using their email address and password.	
Priority	High	
Actors	Customer, Employee, Admin, Delivery Boy	
Pre-condition(s)	The user must have an account	
Post-condition(s)	The user is logged in	
Typical Course of Action		
S#	Actor Action	System Response
1	Selects login option	
2		Prompts for email and password
3	Enters their login information	
4		Displays a message “Logged in successfully”

<b>5</b>		Navigate to homepage
<b>Alternate Course of Action (Invalid Credentials Entered)</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>4</b>		Displays an error "Invalid email or password entered"
Go to 2		

---

### Use Case 3

Identifier	UC-3	
Name	Logout	
Summary	The user can logout from the website	
Priority	High	
Actors	Customer, Employee, Admin, Delivery Boy	
Pre-condition(s)	The user must be logged in	
Post-condition(s)	The user is logged out	
Typical Course of Action		
S#	Actor Action	System Response
<div>Visual Paradigm</div> <div><div>feedbackID</div><div>description</div><div>rating</div><div>viewFeedback</div></div>		

<b>1</b>	<b>Include UC-3</b>	
<b>2</b>	Selects logout option	
<b>3</b>		Displays a message "Successfully Logged out"
<b>4</b>		Navigate to homepage

### Use Case 4

Identifier	UC-4	
Name	Edit Personal Information	
Summary	The customer and employee can edit their personal details (such as name, password, phone number, etc)	
Priority	High	
Actors	Customer, Employee, Delivery Boy	
Pre-condition(s)	Both customer and employee must be logged in to their account.	
Post-condition(s)	Personal information is saved.	
Typical Course of Action		
S#	Actor Action	System Response
1	Include UC-2	
2	Selects edit personal information	



<b>3</b>		Shows a list of options to edit
<b>4</b>	Clicks the desired option	
<b>5</b>		Displays a message "Account updated successfully"

---

### Use Case 5

Identifier	UC-5	
Name	Place Order	
Summary	The customer places an order by selecting from the menu.	
Priority	High	
Actors	Customer	
Pre-condition(s)	1. The customer must be logged in to the account.	
Post-condition(s)	Order is placed.	
Typical Course of Action - Takeaway Order		
S#	Actor Action	System Response
1	Include UC-2, UC-11	
2	Selects items from the menu	
3		Displays a message “Items added to cart” & shows a cart popup

4	Clicks cart button	
5		Displays order details summary & total order bill
6		Prompts for order type {Takeaway, dine-in, delivery}
7	Selects takeaway option	
8	Include UC-13	
9		Displays a message "Order is placed successfully, please wait for confirmation"
Alternate Course of Action (Delivery Order)		
S#	Actor Action	System Response
7	Selects Delivery Option	
8		Prompts to select a location
9	Selects desired delivery location	
Go to 8		
Alternate Course of Action (Dine-in Order)		
S#	Actor Action	System Response
7	Selects dine-in option	
8	Include UC-7	
Go to 8		

---

**Use Case 6**

Identifier		UC-6	
Name		Confirm Order	
Summary		The employee confirms the order which will notify the customer	
Priority		High	
Actors		Employee	
Pre-condition(s)		1. The Employee should be logged in to their account 2. The Customer should have placed their order	
Post-condition(s)		The order is confirmed and placed	
Typical Course of Action			
S#	Actor Action		System Response
1	Include UC-2		
2			Displays customer order details and a button saying "Confirm Order" or "Decline Order"
3	Selects confirm order		
4			Displays a message "Order is placed"
5			View Order Details Summary

<b>Alternate Course of Action (Order is Confirmed with Cash Payment)</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>4</b>		Displays a message "Order is placed"
<b>5</b>		View Order Details Summary
<b>6</b>		Prompts for payment is done via cash
<b>7</b>	Confirms the payment is done	
<b>Alternate Course of Action (Order is not Confirmed)</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>3</b>	Selects decline Order	
<b>4</b>		Displays a message "Order is declined"

---

### Use Case 7

<b>Identifier</b>	UC-7
<b>Name</b>	Reserve Table
<b>Summary</b>	The customer reserves a table by telling the time and number of people
<b>Priority</b>	High
<b>Actors</b>	Customer

<b>Pre-condition(s)</b>		The Customer should have logged in to their account
<b>Post-condition(s)</b>		The Customer table reservation request is accepted
<b>Typical Course of Action</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>1</b>	<b>Include UC-2</b>	
<b>4</b>		Displays option to reserve a table
<b>5</b>	Add details of reservation by giving time and number of people	
<b>6</b>		Displays a message "Reservation is Confirmed" and show reservation details
<b>7</b>		Show reservation details
<b>Alternate Course of Action (No table is available)</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>6</b>		Shows an error "No reservation spots available in the cafe"
<b>7</b>		Prompts to cancel the order or change it to takeaway
<b>8</b>	<b>Go to UC-5 (S# 6)</b>	

---

### Use Case 8

<b>Identifier</b>	UC-8
-------------------	------

<b>Name</b>		Assign Table
<b>Summary</b>		The employee assigns a table to the customer
<b>Priority</b>		Medium
<b>Actors</b>		Employee
<b>Pre-condition(s)</b>		1. The Employee should have logged in to their account 2. The Customer should have placed a dine-in order
<b>Post-condition(s)</b>		A table is reserved for the customer
<b>Typical Course of Action</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>1</b>	<b>Include UC-2</b>	
<b>2</b>		Prompts for accepting customer reservation request
<b>3</b>	Selects accept	
<b>4</b>		Displays message “Reservation request is accepted”
<b>5</b>		Show reservation details
<b>Alternate Course of Action (No table is available)</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>3</b>	Selects decline	
<b>4</b>		Displays a message “Reservation request is declined”

<b>5</b>		Show updated order type status from the customer
----------	--	--

---

### Use Case 9

Identifier		UC-9	
Name		Cancel Order	
Summary		Employee can cancel the order on request	
Priority		Medium	
Actors		Employee	
Pre-condition(s)		The Employee should have logged into their account	
Post-condition(s)		The order is canceled	
Typical Course of Action			
S#	Actor Action		System Response
1	Include UC-2		
2		Prompts a message request customer wants to cancel the order	
3	Selects cancel order		
4		Checks time limit of 10 minutes	
5		Displays a message "Order has been successfully Canceled"	
Alternate Course of Action(Time Limit Exceeds)			

S#	Actor Action	System Response
4		Display a message "Order can't be canceled as the time limit has been exceeded"

---

### Use Case 10

<b>Identifier</b>	UC-10
<b>Name</b>	Give Feedback
<b>Summary</b>	The Customer provides a feedback based on his/her experience
<b>Priority</b>	Low
<b>Actors</b>	Customer
<b>Pre-condition(s)</b>	<ol style="list-style-type: none"> <li>1. The customer should have logged in to their account</li> <li>2. The Customer should have placed an order</li> </ol>
<b>Post-condition(s)</b>	The feedback is submitted to be reviewed by the admin.

#### Typical Course of Action

S#	Actor Action	System Response
1	<b>Include UC-2</b>	
2	Clicks on the give feedback icon	
3		Displays the Feedback Box



<b>4</b>	Types in the feedback and clicks on submit button	
<b>5</b>		Displays the "Feedback is successfully submitted." message.
<b>Alternate Course of Action (Feedback form is empty)</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>4</b>	Submits empty feedback	
<b>5</b>		Displays an error "Empty Feedback cannot be submitted"
<b>6</b>	Go to 3	

---

### Use Case 11

<b>Identifier</b>		UC-11
<b>Name</b>		View Menu
<b>Summary</b>		The user can view the currently available menu
<b>Priority</b>		High
<b>Actors</b>		Customer, Employee, Admin
<b>Pre-condition(s)</b>		The user must be logged in to the account.
<b>Post-condition(s)</b>		None
<b>Typical Course of Action</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>

<b>1</b>	<b>Include UC-2</b>	
<b>2</b>	Clicks a Menu icon	
<b>3</b>		Displays different categories of food
<b>4</b>	Can browse and click through the categories	
<b>5</b>		Shows items with pictures and details

---

### Use Case 12

Identifier	UC-12	
Name	Add Manual Orders	
Summary	The employee can add manual orders to system which are not placed on the site	
Priority	High	
Actors	Employee	
Pre-condition(s)	The employee must be logged in to the account.	
Post-condition(s)	Manual order is added to database	
Typical Course of Action		
S#	Actor Action	System Response
1	Include UC-2	
2	Clicks add a new order	

<b>3</b>		Prompts for selecting order items
<b>4</b>	Add order items using their ID	
<b>5</b>		Prompts for order type {Takeaway, Dine-in}
<b>6</b>	Selects the desired option	
<b>7</b>		Displays a total bill amount and order details summary

---

### Use Case 13

Identifier	UC-13	
Name	Make Payment	
Summary	The Customer pays for the order	
Priority	High	
Actors	Customer	
Pre-condition(s)	1. The customer must have placed an order 2. The customer must be logged in to the account.	
Post-condition(s)	The order is confirmed	
Typical Course of Action (Online Payment)		
S#	Actor Action	System Response
1	Include UC-2, UC-5	

<b>2</b>		Prompts for payment method {Cash on delivery or Online payment}
<b>3</b>	Selects Online Payment	
<b>4</b>		Prompts to "Enter Credit Card Details"
<b>5</b>	Enters card details	
<b>6</b>		Displays a message "Payment is successful & Order is confirmed"
<b>Alternate Course of Action (Cash Payment)</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>3</b>	Selects Payment by Cash	
<b>4</b>		Displays a message "Order is confirmed"
<b>Alternate Course of Action (Invalid Card Details)</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>6</b>		Displays an error "Unable to verify the payment method, try again"
Go to 4		

---

### Use Case 14

<b>Identifier</b>	UC-14
-------------------	-------

<b>Name</b>		Avail Vouchers
<b>Summary</b>		The Customer is able to avail a voucher and get a discount on purchased items.
<b>Priority</b>		Medium
<b>Actors</b>		Customer
<b>Pre-condition(s)</b>		1. The customer must have logged into their account 2. The customer is placing an order
<b>Post-condition(s)</b>		The Customer gets a discount on total bill
<b>Typical Course of Action</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>1</b>	<b>Include UC-2</b>	
<b>2</b>		Prompts a type box field for applying voucher if any
<b>3</b>	Inputs a voucher code and clicks ok	
<b>4</b>		Displays a message "Voucher applied successfully"
<b>5</b>		Shows discounted total bill amount
<b>Alternate Course of Action (Invalid Voucher Code)</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>4</b>		Displays a message "Voucher cannot be applied as it is invalid."
Go to 2		

Alternate Course of Action (Voucher Code is Skipped)		
S#	Actor Action	System Response
3	Selects skip option	
4		Shows total bill amount

---

### Use Case 15

Identifier	UC-15	
Name	View Order Details	
Summary	User can view the order details	
Priority	High	
Actors	Customer, Employee, Delivery Boy	
Pre-condition(s)	The user should have logged into their account	
Post-condition(s)	The order has been placed by the customer	
Typical Course of Action		
S#	Actor Action	System Response
1	Include UC-2	

<b>2</b>	Enter Order ID	
<b>3</b>		Display various options
<b>4</b>	Select view order details	
<b>5</b>		Displays all the details of order
<b>Alternate Course of Action(Order ID is Invalid)</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
		Display Message "Order ID is not Correct".
	Go to 2	

---

### Use Case 16

<b>Identifier</b>	UC-16
<b>Name</b>	Deliver Order
<b>Summary</b>	The delivery boy delivers the order of the customer
<b>Priority</b>	High
<b>Actors</b>	Delivery Boy
<b>Pre-condition(s)</b>	Delivery order is placed by the customer
<b>Post-condition(s)</b>	Order is delivered to customer
<b>Typical Course of Action</b>	

S#	Actor Action	System Response
1	Confirms the order pick-up	
2		View order details and location of the customer
3	Confirms arrival at the location	
5		Displays a message "Order delivery is confirmed by the customer"
6	Confirms order is delivered	
7		Marks the order as delivered
<b>Alternate Course of Action(Customer is not giving any response)</b>		
S#	Actor Action	System Response
5		Displays a message "Wait for customer's confirmation till 15 minutes."
6		Prompts to cancel the order after the wait
7	Selects cancel order option	
		Displays a message "Order has been canceled"
<b>Alternate Course of Action(Customer confirms the order delivery late)</b>		
S#	Actor Action	System Response
5		Displays a message "Wait for customer's confirmation till 15 minutes."
6	Go to 5	



Alternate Course of Action(Delivery boy is at the wrong location)		
S#	Actor Action	System Response
5		Displays a message "the customer is unable to confirm the delivery of their order because you are at a wrong location"
6	Goto 2	

---

### Use Case 17

Identifier	UC-17	
Name	Manage Menu	
Summary	Admin can update items in menu	
Priority	High	
Actors	Admin	
Pre-condition(s)	The Admin should have logged into their account	
Post-condition(s)	The menu is updated	
Typical Course of Action		
S#	Actor Action	System Response
1	Include UC-2	
2	Selects Edit Menu	

<b>3</b>		Shows the categories of menu
<b>4</b>	Selects the desired category	
<b>5</b>		Shows the list of items in the category
<b>6</b>	Selects the desired item	
<b>7</b>		Displays a set of options such as "Available today, Not available temporarily, Remove Item Permanently"
<b>8</b>	Selects the desired option	
<b>9</b>		Displays a message "Menu updated successfully"
<b>Alternate Course of Action (Choosing the same option as current state of item)</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>8</b>	Selects the same option	Displays a message "the item is already set to current status"
Go to 7		

---

### Use Case 18

<b>Identifier</b>	UC-18
<b>Name</b>	Manage Customer
<b>Summary</b>	Admin can Manage the customer details.

Priority	High	
Actors	Admin	
Pre-condition(s)	The Admin should have logged into their account	
Post-condition(s)	The customer records are updated	
Typical Course of Action		
S#	Actor Action	System Response
1	Include UC-2	
2	Selects Edit Customers	
3		Prompts to "Enter Customer ID"
4	Enters Customers ID	
5		Displays a set of options such as "Add Customer, Delete Customer or Edit Customer"
6	Selects the Desired Option	
7		Displays a message "Customer record updated successfully"
Alternate Course of Action (Customer ID is Invalid)		
S#	Actor Action	System Response
5		Displays an error "Customer record does not exist"
Go to 3		

---

**Use Case 19**

<b>Identifier</b>		UC-19
<b>Name</b>		Manage Employee
<b>Summary</b>		Admin can Manage the employee details.
<b>Priority</b>		High
<b>Actors</b>		Admin
<b>Pre-condition(s)</b>		The Admin should have logged into their account
<b>Post-condition(s)</b>		The customer records are updated
<b>Typical Course of Action</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>1</b>	<b>Include UC-2</b>	
<b>2</b>	Selects Edit Employee	
<b>3</b>		Prompts to “Enter Employee ID”
<b>4</b>	Enters Employee ID	
<b>5</b>		Displays a set of options such as “Add Employee, Delete Employee or Edit Employee”
<b>6</b>	Selects the Desired Option	

<b>7</b>		Displays a message "Employee record updated successfully"
<b>Alternate Course of Action (Employee ID is Invalid)</b>		
<b>S#</b>	<b>Actor Action</b>	<b>System Response</b>
<b>5</b>		Displays an error "Employee record does not exist"
Go to 3		

---

### Use Case 20

Identifier	UC-20	
Name	Manage Sales	
Summary	The admin manages orders sales, generate report etc	
Priority	High	
Actors	Admin	
Pre-condition(s)	The admin should have logged in to his account	
Post-condition(s)	The report of sales is generated/updated	
Typical Course of Action		
S#	Actor Action	System Response
1	Include UC-2	
2	Selects manage sales	

<b>3</b>		Displays various operations such as "Generate Sales Report, Get Profit, Calculate Statistics"
<b>4</b>	Selects desired option	
<b>5</b>		Displays the result of selected operation on screen

---

### Use Case 21

Identifier	UC-21	
Name	Create Vouchers	
Summary	The Admin can create vouchers for the frequent customers	
Priority	Medium	
Actors	Admin	
Pre-condition(s)	The Admin should have logged into their account	
Post-condition(s)	Vouchers are generated and sent to customers via mail	
Typical Course of Action		
S#	Actor Action	System Response
1	Include UC-2	
2	Selects create a voucher	
3		Shows the list of frequent customers that are eligible for vouchers
4	Selects all customers in the list	

<b>5</b>		Displays a message "Voucher has been successfully delivered to the customer through mail"
----------	--	---

---

### Use Case 22

Identifier	UC-22	
Name	View Feedback	
Summary	The Admin is able to view all the feedback received.	
Priority	Medium	
Actors	Admin	
Pre-condition(s)	The Admin must be logged in to the account.	
Post-condition(s)	None	
Typical Course of Action		
S#	Actor Action	System Response
1	Include UC-2	
2	Selects view customer feedback option	
3		Displays the list of recent feedbacks from customer with checkmark