



MINISTRY OF

EDUCATION AND TRAINING

# DISCOVERY

The platform supports the deliveries of near-expired foods

Project Code	SE1609G1
Document Code	SE1609G1 - v1.0
Group Member	Nguyen Van Tu Cuong - SE160875 Lam Minh Phuc - SE161448 Tran Hoang Nguyen - SE160170 Vu Dai Minh Quang - SE
Supervisor	Mr. Lai The Hung

Ho Chi Minh City, Oct 26th 2022

# TABLE OF CONTENT

<b>MINISTRY OF EDUCATION AND TRAINING</b>	<b>1</b>
<b>TABLE OF CONTENT</b>	<b>2</b>
<b>1. INTRODUCTION</b>	<b>4</b>
1.1. Purpose	5
1.2. Scope	5
1.3. Definitions, Acronyms, and Abbreviations	5
1.4. References	6
1.5. Overview	6
<b>2. OVERALL DESCRIPTION</b>	<b>7</b>
2.1. Product Overview	7
2.1.1. Guest Requirements	
2.1.2. Customer Requirements	7
2.1.3 Distributor Requirements	7
2.1.4 Admin Requirements	7
2.2. Business Rules	7
<b>3. FUNCTIONAL REQUIREMENT</b>	<b>9</b>
3.1. Use Case Explanation	9
3.2. Use Case Diagram	9
3.3. Use Case List	10

3.4. Use Case Detail	11
UC-01: Sign In	11
UC-02: Sign Up	13
UC-03: Sign Out	15
UC-04: Forget Password	16
UC-05: View Latest Near-expired Foods List	18
UC-06: View Near-expired Food Detail	19
UC-07: Search Near-Expired Foods	20
UC-08: Add Food to Cart	21
UC-09: View Cart	23
UC-10: Remove Food from Cart	23
UC-11: Check Out	25
UC-12: View Check Out History	26
UC-13: Set Default Payment Method	28
UC-14: View Delivery List	29
UC-15: View deliver Detail	30
UC-16: Contact via Built-in Chat System	31
UC-17: Contact via Zalo	32
UC-18: View Profile	33
UC-19: Edit Profile	34
UC-20: View Accounts Detail	35
UC-21: Create Account	37
UC-22: Modify Accounts	38
UC-23: Remove Account	39
UC-24: Remove Sale Tasks	41
UC-25: Modify Sale Tasks	42
UC-26: Submit New Sale Request	44
UC-27: View Own Sale Task	45
UC-28: Create new Sale Request	47
UC-29: Modify Own Sale Tasks	48
UC-30: Remove Own Sale Tasks	49
UC-31: View Delivery	51

UC-32: Create Delivery	52
UC-33: Modify Delivery	53
UC-34: Remove Delivery	56
<b>5. Database</b>	<b>58</b>
5.1. ERD	58
5.2. Relational Schema	58

# 1. INTRODUCTION

## 1.1. Purpose

This document represents the Software Requirements Specification (SRS) of **Dissovery**. Its purpose is to describe the scope, both the functional and non-functional software requirements, as well as the design constraints of the whole logic sub-component. Furthermore, this document shows how the system's interfaces are designed in detail.

## 1.2. Scope

**Dissovery** is the mobile platform that supports customers who are looking for usable foods with better prices and distributors who are having trouble dealing with foods close to date. **Dissovery** also creates connections between customers and distributors, stores delivery information and makes payment.

More specifically, Customers can look for the kind of food they want by providing some keywords, or categories. After that, they can add them to their carts, then check out. A connection and a delivery involving distributors are created, which provide customers information. Distributors can also push their food to the application, to approach more and more customers. We have a use-case model that states the correlation relationship between the user and the system, functional and non-functional requirements, an ERD and a relational schema.

## 1.3. Definitions, Acronyms, and Abbreviations


Acronym	Definition	Vietnamese
Guest	A person using the application without signing in.	Khách mời

Customer	A person who signed in as a customer, the purpose is to buy food.	Khách hàng
Distributor	A person who signed in as a distributor, the purpose is to sell food.	Nhà phân phối
Administrator	A person who signed in as a distributor, the purpose is to manage food requests and organise accounts.	Quản trị viên

## 1.4. References

### Refer User Case Detail

Grap-Work Capstone Project

 Grabwork-Final-Report\_Full (1).docx

### Refer Template

Approved Food

<https://www.approvedfood.co.uk/>

## 1.5. Overview

The next chapter, the Overall Description, of this document provides an overview of the product's functionality. It describes the informal requirements and is used to establish the context for the specification in the next chapter.

The third chapter, the Functional Requirements section, of this document is written to describe the functionality of the product in technical terms including its use-cases and details.

The fourth chapter, the Non-functional Requirements section, of this document is written to describe in detail the non-functionality of the product, defining how a system should look like.

The fifth chapter, the Database section, of this document is written to describe in detail the database (specifically, ERD and Relation Schema) of the product, defining how data are stored and organised.

The last chapter, is the Supporting Information section, of this document provides the supporting information that makes the SRS easier to use. It includes:

- Table of contents
- Index
- Appendices and be considered part of the requirements or not.

## 2. OVERALL DESCRIPTION

### 2.1. Product Overview

#### 2.1.1. Guest Requirements

Guests are people who have not been authenticated to the system. They only have limited access the following functions:

- Sign Up, Sign In, Forget Password
- Search and View Near-expired Foods

#### 2.1.2. Customer Requirements

Members are people who have signed in to the job board website with the “Customer” role. They can use all Guest’s functions, plus these following ones:

- View/Edit Profile
- Add Goods to Cart
- Check Out
- Get Deliveries
- Connect to Distributors

#### 2.1.3 Distributor Requirements

Members are people who have logged in to the job board website with the “Distributor” role. They can use all Customer’s functions, plus these following ones:

- Create New Sale Request
- View/Modify/Remove Own Sale Task
- Get Business Statistics

#### 2.1.4 Admin Requirements

Members are people who have logged in to the admin dashboard website with “Admin” role. They can use all Guest’s functions, plus these following ones:

- Organise Accounts
- Submit New Sale Requests
- Add/Modify/Remove Sale Task

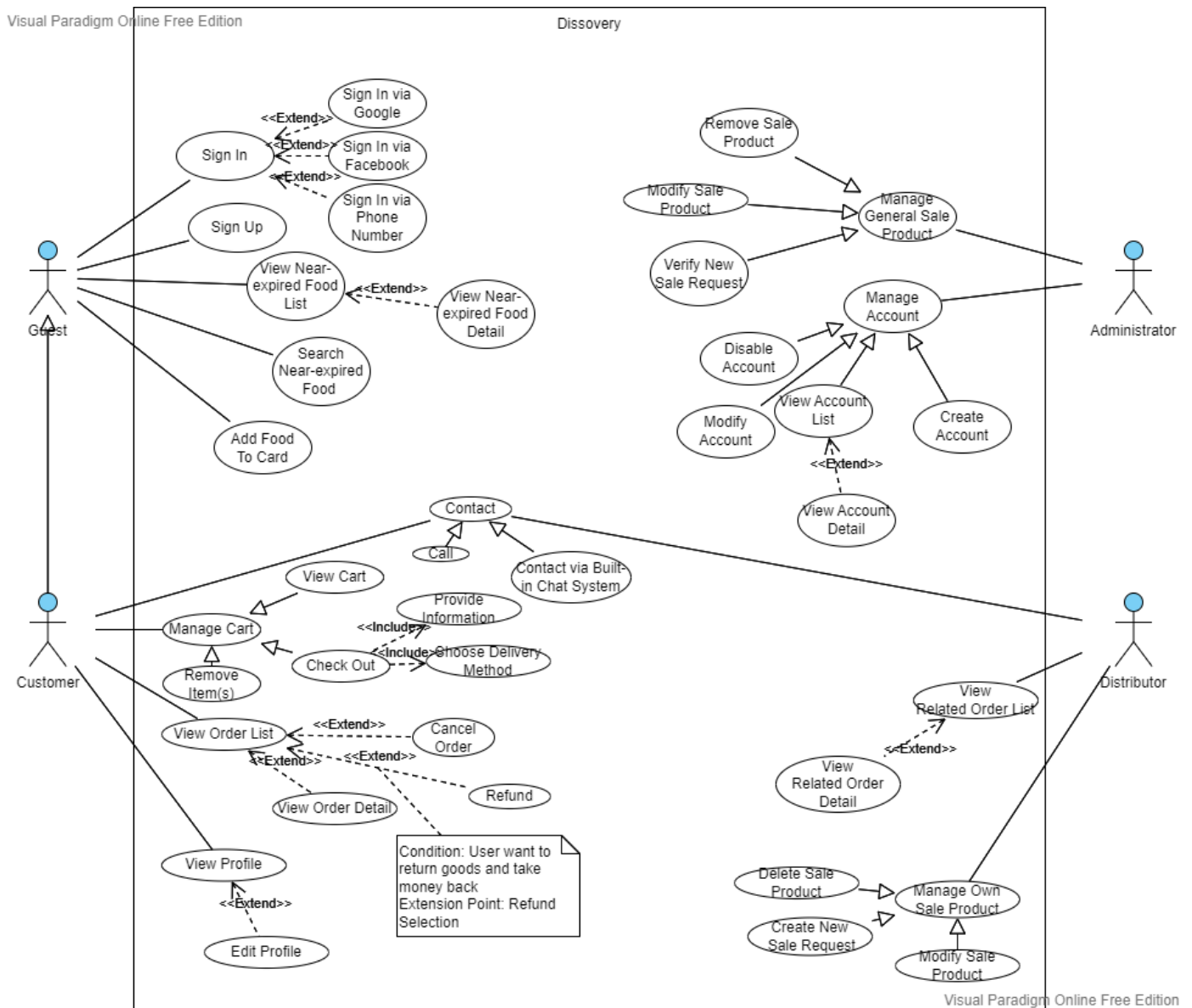


## 2.2. Business Rules

ID	Rule Definition
BR-01	The notification sent to the user must be a push notification, it can show on the device's screen. If the device is turned off, the message should appear again when the user opens the device.
BR-02	Built-in Chat System on the website should be real-time messages.
BR-03	Some job tools such as resume export, tax calculation can be used even there's no internet connection
BR-04	Distributors must upload the business licence to get verified. The time that waits for administrator verifying is under 2 weeks
BR-05	Both customers and distributors must complete the basic profile before doing any action.
BR-06	
BR-07	
BR-08	
BR-09	
BR-10	

### **3. FUNCTIONAL REQUIREMENT**

### 3.1. Use Case Diagram



Full Resolution:

<https://drive.google.com/file/d/1tLNOeJTP8Bpc4F2xnCK1uep1oEuu0BNI/view?usp=sharing>

## 3.2. Use Case List

ID	UseCase	Primary Actor	Secondary Actor
UC-01	Sign In	Guest	
UC-02	Sign Up	Guest	
UC-03	View Near-Expired Food list	Guest	
UC-04	View Near-expired Food Detail	Guest	
UC-05	Search near-Expired Foods	Guest	
UC-06	Add Food To Cart	Guest	
UC-07	View Cart	Customer	
UC-08	Remove Item(s)	Customer	
UC-09	Check Out	Customer	
UC-10	View Order List	Customer	
UC-11	View Order Detail	Customer	
UC-12	Cancel Order	Customer	
UC-13	Refund	Customer	
UC-14	Contact via Built-in Chat System	Customer	
UC-15	Calling	Customer	
UC-16	View Profile	Customer	
UC-17	Edit Profile	Customer	
UC-18	View Accounts List	Admin	
UC-19	View Accounts Detail	Admin	
UC-20	Create Accounts	Admin	

UC-21	Modify Accounts	Admin	
UC-22	Disable Account	Admin	
UC-23	Remove Sale Product	Admin	
UC-24	Modify Sale Product	Admin	
UC-25	Verify New Sale Request	Admin	
UC-26	View own Sale Request	Distributor	
UC-27	Create new Sale Product	Distributor	
UC-28	Modify own Sale Product	Distributor	
UC-29	Delete own Sale Product	Distributor	
UC-30	View Related Order List	Distributor	
UC-31	View Related Order Detail	Distributor	

### 3.3. Use Case Detail

#### UC-01: Sign In

<b>ID and Name:</b>	<b>UC-01: Sign In</b>		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Guest	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Usually
Trigger:	The actor wants to sign in to the Dissovery mobile app.		
Description:	This feature allows actor to login the system		
Preconditions:	Actor has a Facebook Account or a Google Account or a phone number that has registered before		

Post-conditions:	Success: Redirect to home page Fail: The system shows error message corresponding to error code		
Normal Flow:			
	Step	Actor Action	System Response
	1	In sign in page, Actor inputs email and passwords	
	2	Actor clicks “sign in”	System redirects to the home page
Alternative Flow:	1:		
	Step	Actor Action	System Response
	1	In sign in page, Actor clicks on the Facebook icon	System redirects to the Facebook interface and shows a confirmation message
	2	Actor clicks “yes” button	System redirects to the home page
	2:		
	Step	Actor Action	System Response
	1	In sign in page, Actor clicks on the Google icon	System shows a list of google accounts of users
	2	Actor clicks the account they want to use to sign in	System redirects to the home page
	3:		
	Step	Actor Action	System Response
	1	In sign in page, Actor clicks on the Phone icon	System shows a form contains: - Inputting phone number text box

			<ul style="list-style-type: none"> <li>- Send OTP button</li> <li>- Inputting OTP text box</li> <li>- "Login" button</li> </ul>
	2	Actor inputs phone number	
	3	The actor clicks "Send OTP" button	System validates the phone number and invokes 3rd service to send OTP to the actor's device.
	4	Actor inputs OTP number	
	5	Actor clicks "SignUp"	System checks OTP. If correct, create new customer account and redirect to home page
Exceptions:			
	No	Cause	System Response
	1	The email or password is not correct format	System informs error message to remind user inputs correct format
	2	The email or password is not correct	System informs "email or password is incorrect"
	3	Incorrect OTP	System informs "OTP is incorrect"
	4	OTP number is expired	System shows error message: "Expired OTP"
Business Rules:	<ul style="list-style-type: none"> <li>- Password is not empty and at least 8 characters</li> <li>- If an actor provides incorrect username &amp; password 5 times, they will have to wait at least 5 mins until their next try.</li> </ul>		
Assumptions:	<ul style="list-style-type: none"> <li>- Actor's internet connection is available</li> <li>- Actor has signed up by phone number before or has Facebook</li> </ul>		

	Account or has Google Account
--	-------------------------------

## UC-02: Sign Up

ID and Name:	UC-02: Sign Up		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Guest	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Usually
Trigger:	The actor wants to register an account in Dissovery App		
Description:	This feature allows actor to create new account to become customer in the system		
Preconditions:	Actor has an email that has been authorised before (Google, Yahoo, Microsoft, ... ).		
Post-conditions:	Success: Redirect to home page Fail: The system shows error message corresponding to error code		
Normal Flow:			
	Step	Actor Action	System Response
	1	In the sign in page, actor clicks “Sign Up”	System shows a form with these fields: - Email: a text input, a string input must follow the email regex (XXX@YYY.ZZ). - Password/Confirm password: two strings must be identical, not empty and at least 8 characters - First name/Last name: two string are not empty - Phone number: a string of numbers has 9-11 characters.
2	Actor inputs phone number, email,password,		



		confirm password, first name, last name																
	3	The actor clicks “Next”	System sends to that registered email an activation code, then shows a new form with a new field - activation code: A string of numbers that has 6 characters.															
	4	Actor inputs activation																
	5	Actor clicks “Sign Up”	System checks the activation code. If correct, create new customer account and redirect to home page															
Alternative Flow:	N/A																	
Exceptions:	<table><tr><th>No</th><th>Cause</th><th>System Response</th></tr><tr><td>1</td><td>At least one input field doesn’t match the constraint</td><td>System shows an error message corresponding to the fail constraint.</td></tr><tr><td>2</td><td>Email is not existed</td><td>System detects that the registered email does not exist, then shows a warning error.</td></tr><tr><td>3</td><td>activation code is expired</td><td>System shows error message: “Expired activation code”</td></tr><tr><td>4</td><td>Activation code is not correct</td><td>System shows error message: “Incorrect activation code”</td></tr></table>			No	Cause	System Response	1	At least one input field doesn’t match the constraint	System shows an error message corresponding to the fail constraint.	2	Email is not existed	System detects that the registered email does not exist, then shows a warning error.	3	activation code is expired	System shows error message: “Expired activation code”	4	Activation code is not correct	System shows error message: “Incorrect activation code”
No	Cause	System Response																
1	At least one input field doesn’t match the constraint	System shows an error message corresponding to the fail constraint.																
2	Email is not existed	System detects that the registered email does not exist, then shows a warning error.																
3	activation code is expired	System shows error message: “Expired activation code”																
4	Activation code is not correct	System shows error message: “Incorrect activation code”																
Business Rules:	<ul style="list-style-type: none"><li>- Activation code consists of 6 positive numbers and is valid in 10 minutes.</li><li>- If actors input wrong activation code 5 times, their</li></ul>																	

	email/phone-number will be locked in 10 minutes
Assumptions:	<ul style="list-style-type: none"> <li>- Actor does not block Discovery from sending activation code</li> <li>- The activation code 3rd service is still working well</li> <li>- Actor's internet connection is available</li> </ul>

### UC-03: View Near-expired Foods List

UC-03: View Near-expired Foods List			
ID and Name:	UC-03: View Near-expired Foods List		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Guest	Secondary Actors:	N/A
Priority:	Normal	Frequency of Use:	Often
Trigger:	The actor wants to view the list of food.		
Description:	This feature allows actors to view food sorted by latest.		
Preconditions:	N/A		
Post-conditions:	Success: The system shows lists of food categorised by latest date. Fail: System informs “No data” .		
Normal Flow:			
	Step	Actor Action	System Response
	1	In home page, actor clicks on “Home” button.	System shows the list of food.
Alternative Flow:	N/A		
Exceptions:			
	No	Cause	System Response
	1	Result is empty.	System shows message “No data” .

Business Rules:	<ul style="list-style-type: none"> <li>- The food must have a record in the storage and still be available.</li> <li>- All information in the list must be clear and up-to-date.</li> <li>- Foods that have been expired will be hidden.</li> </ul>
Assumptions:	Actor's internet connection is available

#### UC-04: View Near-expired Food Detail

ID and Name:	UC-04: View Near-expired Food Detail		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Guess	Secondary Actors:	N/A
Priority:	Normal	Frequency of Use:	Often
Trigger:	The actor wants to view food in detail after choosing that food.		
Description:	This feature allows the actor to view food in detail.		
Preconditions:	N/A		
Post-conditions:	Success: The System shows the details of food such as name, price, manufacturer, ...		
Normal Flow:			
	Step	Actor Action	System Response
	1	After the actor clicks on the picture of food or name button.	Redirect to the food detail information page. The website shows food detail information such as: <ul style="list-style-type: none"><li>- Name</li><li>- Price</li><li>- Picture</li><li>- Description</li><li>- Available quantity</li><li>- Manufacturer</li></ul>

	<table><tr><td></td><td></td><td>- ... And below page shows “add to cart” button</td></tr></table>			- ... And below page shows “add to cart” button
		- ... And below page shows “add to cart” button		
Alternative Flow:	N/A			
Exceptions:	N/A			
Business Rules:	All foods in the list must be available and up-to-date.			
Assumptions:	Actor’s internet connection is available			

#### UC-05: Search Near-Expired Foods

ID and Name:	UC-05: Search Near-Expired Foods		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Guess	Secondary Actors:	N/A
Priority:	Normal	Frequency of Use:	Often
Trigger:	The actor wants to search for food.		
Description:	This feature allows actor to search for food that details contain the keyword.		
Preconditions:	N/A		
Post-conditions:	Success: The system shows a list of food in search page that details contain the keyword Fail: The system shows message “No data”		
Normal Flow:			
	Step	Actor Action	System Response
	1	In the homepage, the actor types the keyword into the search textbox.	

	<table><tr><td>2</td><td>Actor clicks on the “search” button or search icon.</td><td>System show the list of food that details contain the keyword.</td></tr></table>	2	Actor clicks on the “search” button or search icon.	System show the list of food that details contain the keyword.			
2	Actor clicks on the “search” button or search icon.	System show the list of food that details contain the keyword.					
Alternative Flow:	N/A						
Exceptions:	<table><tr><td>No</td><td>Cause</td><td>System Response</td></tr><tr><td>1</td><td>Result is empty.</td><td>System shows message “No data”.</td></tr></table>	No	Cause	System Response	1	Result is empty.	System shows message “No data”.
No	Cause	System Response					
1	Result is empty.	System shows message “No data”.					
Business Rules:	- All food must have a record in the storage and still be available.						
Assumptions:	Actor’s internet connection is available						

#### UC-06: Add Food to the Cart

<b>ID and Name:</b>	<b>UC-06: Add Food to the Cart</b>		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Customer	Secondary Actors:	N/A
Priority:	Medium	Frequency of Use:	Often
Trigger:	The actor wants to add food to their cart.		
Description:	This feature allows the actor to add any food to their cart.		
Preconditions:	Actors must sign in as “Customer”. Actors should be viewing details of a product.		
Post-conditions:	Success: System saves the product to the cart and return to home page. Fail: System shows message “The amount of this food in stock is not enough.”		
Normal Flow:			

	<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>						
	1	In home page, actor clicks on a food cart.	System redirects to the food detail page						
	2	In the food detail page, actor choose number of products(if countable, or by kg)	System checks the number of that product in the stock and displays the increase or decrease number corresponding to the field.						
	3	In the food’s detail page, actor clicks on “Add to Cart”	System saves food to the cart.						
Alternative Flow:	N/A								
Exceptions:	<table><tr><td>No</td><td>Cause</td><td>System Response</td></tr><tr><td>1</td><td>If the quantity actor wants to buy higher than quantity in the store/database.</td><td>System shows message “The amount of this food in stock is not enough.”</td></tr></table>			No	Cause	System Response	1	If the quantity actor wants to buy higher than quantity in the store/database.	System shows message “The amount of this food in stock is not enough.”
No	Cause	System Response							
1	If the quantity actor wants to buy higher than quantity in the store/database.	System shows message “The amount of this food in stock is not enough.”							
Business Rules:	- The food must be available in stock.								
Assumptions:	Actor’s internet connection is available								

#### UC-07: View the Cart

<b>ID and Name:</b>	<b>UC-07: View the Cart</b>		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Customer	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Usually
Trigger:	The actor wants to view their cart.		

Description:	This feature allows actors to view what food they have chosen to buy by their name, price, number,...								
Preconditions:	The actor must login to their account as “Customer” role								
Post-conditions:	Success: System shows the cart with some food cards Fail: The system shows the message “You haven’t picked anything!”								
Normal Flow:	<table><tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>In home page, Actor clicks “Cart” section</td><td>System redirects to the cart section and shows some food cart that has been picked and a “checkout” button.</td></tr></table>			Step	Actor Action	System Response	1	In home page, Actor clicks “Cart” section	System redirects to the cart section and shows some food cart that has been picked and a “checkout” button.
Step	Actor Action	System Response							
1	In home page, Actor clicks “Cart” section	System redirects to the cart section and shows some food cart that has been picked and a “checkout” button.							
Alternative Flow:	N/A								
Exceptions:	<table><tr><th>No</th><th>Cause</th><th>System Response</th></tr><tr><td>1</td><td>There is no food that has been picked</td><td>System informs the message “You haven’t picked anything!”</td></tr></table>			No	Cause	System Response	1	There is no food that has been picked	System informs the message “You haven’t picked anything!”
No	Cause	System Response							
1	There is no food that has been picked	System informs the message “You haven’t picked anything!”							
Business Rules:	System must show total price of the cart								
Assumptions:	Actor’s internet connection is available								

#### UC-08: Remove Item(s)

ID and Name:	UC-10: Remove Item(s)		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Customer	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Often
Trigger:	The actor wants to remove food from Cart		

Description:	This feature allows actor to remove the food that they have chosen to buy from cart														
Preconditions:	Actor has signed in as “Customer” and has picked some food														
Post-conditions:	Success: The Food is removed from the cart.														
Normal Flow:	<table><tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>In home page, Actor clicks “Cart” section</td><td>System redirects to the cart section and shows some food cart that has been picked and a “checkout” button.</td></tr><tr><td>2</td><td>Actor swipes the food cart that he/she wants to remove</td><td>An “X” button appears.</td></tr><tr><td>3</td><td>Actor clicks the “X” button</td><td>The food card is removed from the cart.</td></tr></table>			Step	Actor Action	System Response	1	In home page, Actor clicks “Cart” section	System redirects to the cart section and shows some food cart that has been picked and a “checkout” button.	2	Actor swipes the food cart that he/she wants to remove	An “X” button appears.	3	Actor clicks the “X” button	The food card is removed from the cart.
Step	Actor Action	System Response													
1	In home page, Actor clicks “Cart” section	System redirects to the cart section and shows some food cart that has been picked and a “checkout” button.													
2	Actor swipes the food cart that he/she wants to remove	An “X” button appears.													
3	Actor clicks the “X” button	The food card is removed from the cart.													
Alternative Flow:	1: <table><tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>In home page, Actor clicks “Cart” section</td><td>System redirects to the cart section and shows some food cart that has been picked and a “check out” button.</td></tr><tr><td>2</td><td>Actor clicks the trash icon button</td><td>System shows the message “Are you sure to remove all food in the cart? Y/N”</td></tr><tr><td>3</td><td>Actor clicks “Yes”</td><td>System removes all food carts in the cart.</td></tr></table>			Step	Actor Action	System Response	1	In home page, Actor clicks “Cart” section	System redirects to the cart section and shows some food cart that has been picked and a “check out” button.	2	Actor clicks the trash icon button	System shows the message “Are you sure to remove all food in the cart? Y/N”	3	Actor clicks “Yes”	System removes all food carts in the cart.
Step	Actor Action	System Response													
1	In home page, Actor clicks “Cart” section	System redirects to the cart section and shows some food cart that has been picked and a “check out” button.													
2	Actor clicks the trash icon button	System shows the message “Are you sure to remove all food in the cart? Y/N”													
3	Actor clicks “Yes”	System removes all food carts in the cart.													
Exceptions:	N/A														



Business Rules:	N/A
Assumptions:	Actor's internet connection is available

#### UC-09: Check Out

ID and Name:	UC-09: Check Out		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Customer	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Often
Trigger:	The actor wants to check out.		
Description:	This feature allows actor to make order for the food they have chosen in carts		
Preconditions:	<ul style="list-style-type: none"><li>- The actor must login to their account as “Customer” role</li><li>- The actor has set payment and a place to deliver the food.</li></ul>		
Post-conditions:	Success: System shows message “Purchase successfully!” and save the order to history		
Normal Flow:			
	Step	Actor Action	System Response
	1	In home page, Actor clicks “Cart” section	System redirects to the cart section and shows some food cart that has been picked and a “checkout” button.
	2	Actor clicks “Checkout” card	System shows two options of receiving method “Directly”, “Delivery”.
	3	Actor clicks “Directly” button	System shows a QR code, message: “purchase successfully!” and saves the order to history
Alternative Flow:	1:		

	<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>
	1	In home page, Actor clicks "Cart" section	System redirects to the cart section and shows some food cart that has been picked and a "checkout" button.
	2	Actor clicks "Checkout" card	System shows two options of receiving method "Directly", "Delivery".
	3	Actor clicks "Delivery" button	System shows a form with these fields: <ul style="list-style-type: none"> <li>- Address</li> <li>- Phone number</li> </ul>
	4	Actor inputs the form	
	5	Actor clicks "Confirm" button	System valids the information then shows message: "Purchase successfully!" and save the order to history
Exceptions:			
	<b>No</b>	<b>Cause</b>	<b>System Response</b>
	1	Phone number is in incorrect format or unavailable.	System informs the message "There is no such phone number."
Business Rules:	<ul style="list-style-type: none"> <li>- QR code will expired after 1 hour</li> <li>- Address can be found by google map</li> <li>- Phone number must existed</li> </ul>		
Assumptions:	Actor's internet connection is available		

#### UC-10: View Order List

<b>ID and Name:</b>	<b>UC-10: View Order List</b>
---------------------	-------------------------------

Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Customer	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Often
Trigger:	The actor wants to view order that has been checked out		
Description:	This feature allows the actor to view the order that has been made in the past sorted by date from nearest		
Preconditions:	- The actor must login to their account as “Customer” role		
Post-conditions:	Success: System shows a list of created order Fail: The system shows message “No data”		
Normal Flow:			
	Step	Actor Action	System Response
	1	In home page, actor clicks on profile section	System redirects to profile page containing view history action.
	2	Actor clicks “View Order” button	System redirects history page showing a list of created order sorted by date from nearest
Alternative Flow:	N/A		
Exceptions:			
	No	Cause	System Response
	1	The actor has never checked out	System informs “No data”
Business Rules:	N/A		
Assumptions:	Actor’s internet connection is available		

### UC-11: View Order Detail

ID and Name:	UC-11: View Order detail		
Created By:	QuangVDM	Date Created:	11/10/2022
Primary Actor:	Customer	Secondary Actors:	N/A
Priority:	Normal	Frequency of Use:	Often
Trigger:	The actor wants to view details of an order.		
Description:	This feature allows actors to set a default payment method, which supports all later check out.		
Preconditions:	<ul style="list-style-type: none"><li>- Guests must login and use the function as “Customer”.</li><li>- The actor has checked out at least 1 card before.</li></ul>		
Post-conditions:	Success: System shows details of an order card.		
Normal Flow:			
	Step	Actor Action	System Response
	1	In home page, actor clicks on profile section	System redirects to profile page containing view history action.
	2	Actor clicks “View Order” button	System redirects history page showing a list of created order sorted by date from nearest
	3	Actor clicks an order card	System redirects to detail page showing: <ul style="list-style-type: none"><li>- Food card(name, number, price,...)</li><li>- Total Price</li><li>- Phone number</li></ul>

	<table><tr><td></td><td></td><td><ul style="list-style-type: none"><li>- Address</li><li>- Date created</li><li>- Processing status(paid, Pending, Cancelled)</li></ul></td></tr></table>			<ul style="list-style-type: none"><li>- Address</li><li>- Date created</li><li>- Processing status(paid, Pending, Cancelled)</li></ul>
		<ul style="list-style-type: none"><li>- Address</li><li>- Date created</li><li>- Processing status(paid, Pending, Cancelled)</li></ul>		
Alternative Flow:	N/A			
Exceptions:	N/A			
Business Rules:	<ul style="list-style-type: none"><li>- Actor’s payment method must be valid.</li><li>- After synchronisation, actor has to wait 1 month after next alteration.</li></ul>			
Assumptions:	Actor’s internet connection is available			

#### UC-12: Cancel Order

<b>ID and Name:</b>	<b>UC-12: Cancel Order</b>		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Customer	Secondary Actors:	N/A
Priority:	Medium	Frequency of Use:	Often
Trigger:	The actor wants to cancel an order		
Description:	This feature allows actor to cancel an order that has been checked out but not paid.		
Preconditions:	<ul style="list-style-type: none"> <li>- The actor login to Discovery System as Customer</li> <li>- The actor has checked out an order but hasn't paid nor received the products.</li> <li>- The order has "pending" status.</li> </ul>		
Post-conditions:	Success: The system displays shows message "Your delivery has been cancelled successfully" Fail: System shows message "The message has not been cancelled"		
Normal Flow:			

	Step	Actor Action	System Response
	1	In home page, actor clicks on profile section	- System redirects to profile page containing view history action.
	2	Actor clicks "View Order" button	- System redirects history page showing a list of created order sorted by date from nearest
	3	Actor clicks on "Cancel" button of an order card	System shows message: "Would you like to cancel this order? Y/N"
	4	Actor clicks "Yes"	System shows status of the order which is now "Cancelled"
Alternative Flow:			
	Step	Actor Action	System Response
	1	In home page, actor clicks on profile section	- System redirects to profile page containing view history action.
	2	Actor clicks "View Order" button	- System redirects history page showing a list of created order sorted by date from nearest
	3	Actor clicks an order card	System redirects to detail page showing: <ul style="list-style-type: none"> <li>- Food card(name, number, price,...)</li> <li>- Total Price</li> <li>- Pay method</li> <li>- Address</li> <li>- Date created</li> </ul>

			<ul style="list-style-type: none"><li>- Processing status(paid, Unpaid, Cancel)</li></ul>						
	4	Actor clicks on “Cancel” button of an order card	<ul style="list-style-type: none"><li>- System shows a message: “Would you like to cancel this order? Y/N”</li></ul>						
	5	Actor clicks “Yes”	System shows status of the order which is now “Cancelled”						
Exceptions:	<table><tr><th>No</th><th>Cause</th><th>System Response</th></tr><tr><td>1</td><td>Actor clicks “No”</td><td>System shows a message: “The message has not been cancelled”</td></tr></table>			No	Cause	System Response	1	Actor clicks “No”	System shows a message: “The message has not been cancelled”
No	Cause	System Response							
1	Actor clicks “No”	System shows a message: “The message has not been cancelled”							
Business Rules:	<ul style="list-style-type: none"><li>- The order hasn’t been paid.</li><li>- If the order has been delivered, the estimated shipped date must be before or the same with the day in the present.</li></ul>								
Assumptions:	Actor’s internet connection is available								

### UC-13: Refund

<b>ID and Name:</b>	<b>UC-13: Refund</b>		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Customer	Secondary Actors:	N/A
Priority:	Medium	Frequency of Use:	Often
Trigger:	The actor wants to refund a paid order		
Description:	This feature allows actor to return the product to the supermarket and receive back money		

Preconditions:	<ul style="list-style-type: none"><li>- The actor login to Dissovery System as Customer</li><li>- The actor has checked out, paid for an order and received the product.</li><li>- The order has “paid” status.</li></ul>																	
Post-conditions:	Success: The system displays shows message “Your delivery has been cancelled successfully” Fail: System shows message “The message has not been cancelled”																	
Normal Flow:	<table><tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>In home page, actor clicks on profile section</td><td>System redirects to profile page containing view history action.</td></tr><tr><td>2</td><td>Actor clicks “View Order” button</td><td>System redirects history page showing a list of created order sorted by date from nearest</td></tr><tr><td>3</td><td>Actor clicks on “Refund” button of an order card</td><td>System shows a message: “Would you like a refund for this order? Y/N”</td></tr><tr><td>4</td><td>Actor clicks “Yes”</td><td>System shows a QR code and a message: “This qr code will expire after 1 hour. Please show this to the shopkeeper. You can only refund if products are intact!”</td></tr></table>			Step	Actor Action	System Response	1	In home page, actor clicks on profile section	System redirects to profile page containing view history action.	2	Actor clicks “View Order” button	System redirects history page showing a list of created order sorted by date from nearest	3	Actor clicks on “Refund” button of an order card	System shows a message: “Would you like a refund for this order? Y/N”	4	Actor clicks “Yes”	System shows a QR code and a message: “This qr code will expire after 1 hour. Please show this to the shopkeeper. You can only refund if products are intact!”
Step	Actor Action	System Response																
1	In home page, actor clicks on profile section	System redirects to profile page containing view history action.																
2	Actor clicks “View Order” button	System redirects history page showing a list of created order sorted by date from nearest																
3	Actor clicks on “Refund” button of an order card	System shows a message: “Would you like a refund for this order? Y/N”																
4	Actor clicks “Yes”	System shows a QR code and a message: “This qr code will expire after 1 hour. Please show this to the shopkeeper. You can only refund if products are intact!”																
Alternative Flow:	<table><tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>In home page, actor clicks on profile section</td><td><ul style="list-style-type: none"><li>- System redirects to profile page containing view history action.</li></ul></td></tr><tr><td>2</td><td>Actor clicks “View Order”</td><td><ul style="list-style-type: none"><li>- System redirects history</li></ul></td></tr></table>			Step	Actor Action	System Response	1	In home page, actor clicks on profile section	<ul style="list-style-type: none"><li>- System redirects to profile page containing view history action.</li></ul>	2	Actor clicks “View Order”	<ul style="list-style-type: none"><li>- System redirects history</li></ul>						
Step	Actor Action	System Response																
1	In home page, actor clicks on profile section	<ul style="list-style-type: none"><li>- System redirects to profile page containing view history action.</li></ul>																
2	Actor clicks “View Order”	<ul style="list-style-type: none"><li>- System redirects history</li></ul>																



		button	page showing a list of created order sorted by date from nearest						
	3	Actor clicks an order card	System redirects to detail page showing: <ul style="list-style-type: none"><li>- Food card(name, number, price,...)</li><li>- Total Price</li><li>- Pay method</li><li>- Address</li><li>- Date created</li><li>- Processing status(paid, Unpaid, Cancel)</li></ul>						
	4	Actor clicks on “Refund” button of an order card	<ul style="list-style-type: none"><li>- System shows message: “Would you like to refund for this order? Y/N”</li></ul>						
	5	Actor clicks “Yes”	System shows a QR code and a message: “This qr code will expire after 1 hour. Please show this to the shopkeeper. You can only refund if products are intact!”						
Exceptions:	<table><tr><th>No</th><th>Cause</th><th>System Response</th></tr><tr><td>1</td><td>Actor clicks “No”</td><td>System shows a message:“The order has not been cancelled”</td></tr></table>			No	Cause	System Response	1	Actor clicks “No”	System shows a message:“The order has not been cancelled”
No	Cause	System Response							
1	Actor clicks “No”	System shows a message:“The order has not been cancelled”							
Business Rules:	<ul style="list-style-type: none"><li>- The order has been paid within 24 hours.</li><li>- QR code will expired after 1 hour.</li><li>- The product must be intact.</li></ul>								
Assumptions:	Actor’s internet connection is available								

#### UC-14: Contact via Built-in Chat System

ID and Name:	UC-14: Contact via Built-in Chat System		
Created By:	PhucLM	Date Created:	11/10/2022
Primary Actor:	Customer	Secondary Actors:	Distributor
Priority:	Normal	Frequency of Use:	Often
Trigger:	The actor wants to contact on App.		
Description:	This feature allows actor to access the chat box in order to understand more about the product and feedback		
Preconditions:	N/A		
Post-conditions:	Success: system shows a successful message Fail: System shows error message		
Normal Flow:			
	Step	Actor Action	System Response
	1	Actor clicks icon textbox	System shows a list of consultants.
	2	Actor choose a consultants to chat	System shows a chatbox between actor and consultants
	3	Actor types a message and click icon send or “ENTER”	System redirect a message into chatbox [Exception 1]
Alternative Flow:	1:		
	Step	Actor Action	System Response

	<table><tr><td>1</td><td>In chatbox, actor clicks icon call</td><td>System connect actor with consultants by a call</td></tr></table>			1	In chatbox, actor clicks icon call	System connect actor with consultants by a call			
1	In chatbox, actor clicks icon call	System connect actor with consultants by a call							
Exceptions:	<table><tr><td>No</td><td>Cause</td><td>System Response</td></tr><tr><td>1</td><td>Loss of network connection</td><td>System shows a message with an icon “!”</td></tr></table>			No	Cause	System Response	1	Loss of network connection	System shows a message with an icon “!”
No	Cause	System Response							
1	Loss of network connection	System shows a message with an icon “!”							
Business Rules:	- Actor can also give feedback via this way of contact.								
Assumptions:	Actor’s internet connection is available								

#### UC-15: Calling

UC-17: Calling						
ID and Name:	UC-17: Calling					
Created By:	PhucLM	Date Created:	11/10/2022			
Primary Actor:	Customer	Secondary Actors:	Distributor			
Priority:	Normal	Frequency of Use:	Often			
Trigger:	The actor wants to contact the distributor.					
Description:	This feature allows actor to call distributor and have a conversation					
Preconditions:	The actor must login as “Customer” The actor must have a phone number.					
Post-conditions:	Success: System shows a phone icon and voice of distributor can be hear. Fail: System sends sound message corresponding to the call.					
Normal Flow:	<table><tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr></table>			Step	Actor Action	System Response
Step	Actor Action	System Response				

	1	Actor clicks icon textbox	System shows a list of consultants.
	2	Actor choose a consultants to call	System shows a chatbox between actor and consultants
	3	Actor clicks on phone icon	System shows the interface of the call and connects actors with consultants by a call. [Exception 1]
Alternative Flow:	1:		
	Step	Actor Action	System Response
	1	In chatbox, actor clicks icon call [Exception 2]	System connect actor with consultants by a call
	2	In the chat box, actor clicks the icon "..."	System shows "Media & Files "of the conversation between actor and consultants
Exceptions:	N/A		
Business Rules:	N/A		
Assumptions:	Actor's internet connection is available		

#### UC-16: View Profile

<b>ID and Name:</b>	<b>UC-16: View Profile</b>		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Customer	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Often

Trigger:	The actor wants to view their profile.		
Description:	This feature allows the actor to view a detailed profile and some support services such as name, view payment card, view history button,...		
Preconditions:	The actor must login to their account as “Customer” role		
Post-conditions:	Success: System shows the profile page		
Normal Flow:			
	Step	Actor Action	System Response
	1	In home page, actor clicks profile section	System redirects to the profile page showing: <ul style="list-style-type: none"><li>- Name</li><li>- Phone Number(if signed in by phone number)</li><li>- View payment card</li><li>- View history button</li><li>- Contact</li><li>- Address</li></ul>
Alternative Flow:	N/A		
Exceptions:	N/A		
Business Rules:	N/A		
Assumptions:	Actor’s internet connection is available		

#### UC-17: Edit Profile

<b>ID and Name:</b>	<b>UC-17: Edit Profile</b>		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Customer	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Often

Trigger:	The actor wants to edit their profile		
Description:	This feature allows actor to change their name, address and/or date of birth		
Preconditions:	The actor must login to their account as “Customer” role		
Post-conditions:	Success: System shows the changed information		
Normal Flow:			
	Step	Actor Action	System Response
	1	In home page, actor clicks profile section	System redirects to the profile page showing: <ul style="list-style-type: none"><li>- Name</li><li>- Phone Number(if signed in by phone number)</li><li>- View payment card</li><li>- View history button</li><li>- Contact</li><li>- Address</li><li>- Date of Birth</li></ul>
	2	Actor clicks personal information card	System shows a form containing textbox for: <ul style="list-style-type: none"><li>- Name</li><li>- Address</li><li>- Date of Birth</li></ul>
	3	Actor inputs the form and clicks “save” button	System shows a new information profile page.
Alternative Flow:	N/A		
Exceptions:	N/A		
Business Rules:	The date of birth textbox must be imputed by choosing a day from the calendar.		
Assumptions:	Actor’s internet connection is available		

### UC-18: View Accounts List

ID and Name:	UC-18: View Accounts List		
Created By:	QuangVDM	Date Created:	11/10/2022
Primary Actor:	Admin	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Often
Trigger:	The actor wants to view accounts in the system		
Description:	This feature allows actor to view list of accounts in the system		
Preconditions:	The actor must login to their account as “admin” role Current screen is the “Homepage” screen.		
Post-conditions:	Success: System returned list of accounts. Fail: System shows message error.		
Normal Flow:			
	Step	Actor Action	System Response
	1	In the home page, click on the “Manage Account” button in Account manage tasks on the navigation bar.	System redirect to “Item Repository” screen, each row item displays basic information like: <ul style="list-style-type: none"><li>- Name: text</li><li>- Email: text</li><li>- ID: text</li><li>- Gender: text</li><li>- Birthday: text</li><li>- Avatar: image</li></ul>
Alternative Flow:	N/A		
Exceptions:	N/A		
Business Rules:	N/A		

Assumptions:	Actor's internet connection is available
--------------	--

### UC-19: View Accounts Detail

ID and Name:	UC-19: View Accounts Detail		
Created By:	PhucLM	Date Created:	11/10/2022
Primary Actor:	Admin	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Often
Trigger:	The actor wants to view detailed information about Accounts.		
Description:	This feature allows actor to view details such as Name, Phone, Address, ...		
Preconditions:	The account publicises their profile.		
Post-conditions:	Success: Show detail information of accounts Fail: Show not found page if the account does not exist or the account is not public.		
Normal Flow:			
	Step	Actor Action	System Response
	1	Click on Account detail’s card in page “Manage Account”.  [Alternative 1]	Redirect to enterprise detail information page showing enterprise detail information such as: <ul style="list-style-type: none"><li>- Name</li><li>- Phone</li><li>- Address</li><li>- Date of Birth</li><li>- Contact information (if any)</li></ul>
Alternative Flow:	1:		



	Step	Actor Action	System Response						
	1	In browser search box, actors enter a links of account detail	System redirects to candidate detail information page. The website shows candidate detail information such as: <ul style="list-style-type: none"><li>- Name</li><li>- Job title</li><li>- Address</li><li>- Educations</li><li>- Work experiences</li><li>- Awards</li></ul> <b>[Exception 1]</b>						
Exceptions:	<table><tr><th>No</th><th>Cause</th><th>System Response</th></tr><tr><td>1</td><td>Result is empty</td><td>System informs “Rất tiếc chúng tôi không tìm thấy kết quả theo yêu cầu của bạn.”.</td></tr></table>			No	Cause	System Response	1	Result is empty	System informs “Rất tiếc chúng tôi không tìm thấy kết quả theo yêu cầu của bạn.”.
No	Cause	System Response							
1	Result is empty	System informs “Rất tiếc chúng tôi không tìm thấy kết quả theo yêu cầu của bạn.”.							
Business Rules:	<ul style="list-style-type: none"><li>- Accounts display for actor must allow publish status and status of account must be active.</li><li>- If an actor doesn’t choose age options in a field of filters, the system will respond to all values of these fields and the field age is from 18 to 60 to the actor.</li></ul>								
Assumptions:	Actor’s internet connection is available								

#### UC-20: Create Account

<b>ID and Name:</b>	<b>UC-20: Create Account</b>		
Created By:	PhucLM	Date Created:	11/10/2022
Primary Actor:	Admin	Secondary Actors:	N/A

Priority:	High	Frequency of Use:	Often
Trigger:	The actor wants to create a new Account.		
Description:	This feature allows actor to create a new account as public.		
Preconditions:	<ul style="list-style-type: none"><li>- The actor has logged in by an Admin account.</li><li>- Current screen is "Account List" screen</li></ul>		
Post-conditions:	Success: System created account Fail: System shows message error		
Normal Flow:			
	Step	Actor Action	System Response
	1	The actor clickson "Create Account" button on "Account List" screen	System shows pop up showing all field for the admin to fill, includes: <ul style="list-style-type: none"><li>- Name: text</li><li>- Email: text</li><li>- Phone: text</li><li>- Gender: text</li><li>- Birthday: text</li><li>- Avatar: image</li><li>- Role: text</li></ul>
	2	Admin fills information, then clicks on the "Create" button. (Alternative A1)	System valid item's information inputted by admin and creates account.
Alternative Flow:			
	Step	Actor Action	System Response
	1	User clicks on the "Cancel" button.	System closes the pop up.

Exceptions:			
	No	Cause	System Response
	1	Existing invalid input	System informs a specified error message in a popup dialog.
Business Rules:	Name cannot be null		
Assumptions:	Actor's internet connection is available		

### UC-21: Modify Accounts

ID and Name:	UC-21: Modify Accounts		
Created By:	PhucLM	Date Created:	11/10/2022
Primary Actor:	Admin	Secondary Actors:	N/A
Priority:	Normal	Frequency of Use:	Often
Trigger:	The actor wants to modify detailed information of accounts.		
Description:	This feature allows Admin to update accounts.		
Preconditions:	<ul style="list-style-type: none"> <li>- User has logged in as Admin successfully.</li> <li>- Current screen is the "Account List" screen.</li> </ul>		
Post-conditions:	Success: System shows a dialog with a successful message when the account has been modified successfully in the database. Fail: System shows a dialog with specified error message.		
Normal Flow:			
	Step	Actor Action	System Response
	1	Click on Account detail's card in page "Manage Account".  <b>[Alternative 1]</b>	System shows a form with these fields: <ul style="list-style-type: none"> <li>- Name: text</li> <li>- Email: text</li> <li>- Phone: text</li> </ul>

			<ul style="list-style-type: none"> <li>- Gender: text</li> <li>- Birthday: text</li> <li>- Avatar: image</li> </ul>
	2	Admin fills information, then clicks on the "Update" button. (Alternative A1)	System modifies the fields and redirects to the home screen interface.
Alternative Flow:			
	Step	Actor Action	System Response
	1	User clicks on the "Cancel" button. [Alternative 1]	System closes the pop up.
Exceptions:			
	No	Cause	System Response
	1	Existing invalid input.	System informs a specified error message in a popup dialog.
Business Rules:	Account just only can be updated by the Admin that created it.		
Assumptions:	Actor's internet connection is available		

## UC-22: Disable Accounts

<b>ID and Name:</b>	<b>UC-22: Disable Account</b>		
Created By:	QuangVDM	Date Created:	11/10/2022
Primary Actor:	Admin	Secondary Actors:	N/A

Priority:	High	Frequency of Use:	Often
Trigger:	The actor wants to view detailed information about Accounts.		
Description:	This feature allows actors to view details such as Name, Phone, Address, ...		
Preconditions:	The account publicises their profile.		
Post-conditions:	Success: Show detail information of accounts  Fail: Show not found page if the account does not exist or the account is not public.		
Normal Flow:			
	Step	Actor Action	System Response
	1	Click on Account detail's card.	Redirect to enterprise detail information page showing enterprise detail information such as: <ul style="list-style-type: none"><li>• Name</li><li>• Phone</li><li>• Address</li><li>• Date of Birth</li><li>• Contact information (if any)</li></ul> And 2 Dot Selection with values "Activate" and "Disable".

	2	Actor clicks on the “Activate” button.  [Alternative 1]	System redirects to the manage account page and set user’s account as Active account.
Alternative Flow:	1:		
	Step	Actor Action	System Response
	1	Actor clicks on the “Disable” button.	System redirects to the manage account page and set user’s account as Disable account.
Business Rules:	<ul style="list-style-type: none"> <li>Accounts displayed for actors must allow publish status and status of account must be active.</li> <li>If an actor doesn’t choose age options in a field of filters, the system will respond to all values of these fields and the field age is from 18 to 60 to the actor.</li> </ul>		
Assumptions:	Actor’s internet connection is available		

### UC-23: Remove Sale Product

ID and Name:	UC-23: Remove Sale Product		
Created By:	PhucLM	Date Created:	11/10/2022
Primary Actor:	Admin	Secondary Actors:	N/A

Priority:	Normal	Frequency of Use:	Often
Trigger:	The actor wants to remove the sales task .		
Description:	This feature allows actor to remove a sale task created by Distributor.		
Preconditions:	Actor login into an App with an account has an admin role.		
Post-conditions:	Success: System displays a message “Remove sale task success!”.		
Normal Flow:			
	Step	Actor Action	System Response
	1	In the Sale tasks list page, actor clicks the button in the Action column at the row that the actor wants to delete.	System shows options :  - Delete
	2	Actor clicks “Remove”	System shows a message is “Remove sale task success!” and remove this food.
Alternative Flow:			
	Step	Actor Action	System Response
	1	In sale tasks list page, actor inputs keyword to search textbox	System shows a list of sale tasks containing the keyword
	1	In sale tasks list, actor clicks the button in the Action column at the row that the actor wants to delete.	System shows options :  - Delete
	2	Actor clicks “Remove”	System shows a message is “Remove sale task success!” and remove this food.

Exceptions:	N/A
Business Rules:	If Actor has already submitted for a sale task , he cannot submit that sale task again
Assumptions:	Actor's internet connection is available

#### UC-24: Modify Sale Product

ID and Name:	UC-24: Modify Sale Product		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Admin	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Often
Trigger:	The actor wants to modify sale task		
Description:	This feature allows actor to modify the sale tasks created by Distributor		
Preconditions:	The actor must login to their account as an “Admin” role.		
Post-conditions:	Success: System shows message: “The sale tasks have been updated” and shows the new information. Fail: System shows a dialog with specified error message.		
Normal Flow:			
	Step	Actor Action	System Response
	1	In home page, actor clicks “Sale Task” section	System shows a list of sale tasks sorted by date from nearest.
	2	Actor clicks on a sale task card	System show details of the sale task in these fields: <ul style="list-style-type: none"><li>- Name of food</li><li>- Price of food</li><li>- Number in stock</li></ul>



			<ul style="list-style-type: none"> <li>- Description</li> <li>- Image</li> <li>- Produced and expired date</li> </ul>
	3	Actor clicks on a field	System shows keyboard to modify the field
	4	Actor inputs new information	
	5	Actor clicks "Save" button	System saves the new information and shows message: "The sale tasks has been updated"
Alternative Flow:			
	Step	Actor Action	System Response
	1	In home page, actor clicks "Sale Task" section	System shows a list of sale tasks sorted by date from nearest.
	2	Actor inputs keyword into search textbox	System shows a list of sale tasks containing the keyword
	3	Actor clicks on a sale task card	System show details of the sale task in these fields: <ul style="list-style-type: none"> <li>- Name of food</li> <li>- Price of food</li> <li>- Number in stock</li> <li>- Description</li> <li>- Image</li> <li>- Produced and expired date</li> </ul>
	4	Actor clicks on a field	System shows keyboard to modify the field
	5	Actor inputs new information	
	6	Actor clicks "Save"	System saves the new information

	button	and shows message: "The sale tasks has been updated"						
Exceptions:	<table> <tr> <th>No</th><th>Cause</th><th>System Response</th></tr> <tr> <td>1</td><td>There is no sale task containing the keyword</td><td>System informs "No data"</td></tr> </table>		No	Cause	System Response	1	There is no sale task containing the keyword	System informs "No data"
No	Cause	System Response						
1	There is no sale task containing the keyword	System informs "No data"						
Business Rules:	<ul style="list-style-type: none"> <li>- Except for Description, other fields can't be blank.</li> <li>- Price must be higher than 0.</li> <li>- Number in stock must be higher or equal 0. If Number is equal to 0, the customer can only view the food.</li> <li>- If the food is expired, the sale task can only be viewed or removed</li> </ul>							
Assumptions:	Actor's internet connection is available							

### UC-25: Verify New Sale Request

<b>ID and Name:</b>	<b>UC-25: Verify New Sale Request</b>		
Created By:	PhucLM	Date Created:	11/10/2022
Primary Actor:	Admin	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Often
Trigger:	The actor wants to submit for a food offer so that he can buy it		
Description:	This feature allows actor to submit for a food offer to start process		
Preconditions:	<ul style="list-style-type: none"> <li>- Actors login into website with Admin roles.</li> <li>- Actor has updated food offer</li> </ul>		
Post-conditions:	Success: · System inform message "Submit successfully".		

	Fail: System informs error message based on the error code.		
Normal Flow:			
	Step	Actor Action	System Response
	1	In the food detail page, actor clicks to submit for a food offer.	System shows a confirmed modal.
	2	Actor clicks to "Confirm" in the confirm modal.	System shows a successful message. <b>[Exception 1]</b>
Alternative Flow:	N/A		
Exceptions:			
	No	Cause	System Response
	1	The food offer is full	System shows error "The list food offer is full "
Business Rules:	If Actor has already submitted for a food offer , he cannot submit that food offer again		
Assumptions:	Actor's internet connection is available		

#### UC-26: View Own Sale Product

<b>ID and Name:</b>	<b>UC-27: View Own Sale Product</b>		
Created By:	PhucLM	Date Created:	11/10/2022
Primary Actor:	Distributor	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Usually
Trigger:	The actor wants to view their own sale product		

Description:	This feature allows actor to view details of their own sale task such as: name of food, price, quantity,...											
Preconditions:	The actor login to Dissovery System as Distributor											
Post-conditions:	Success: The system displays details of Sale Product Fail: System inform “No Data”											
Normal Flow:	<table><tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>In the home page, Actor clicks on a sale task card</td><td>System show details of the sale task in these fields:<ul style="list-style-type: none"><li>- Image</li><li>- Product Name</li><li>- Description</li><li>- Price of food</li><li>- Quantity</li><li>- Button Remove &amp; Modify</li></ul></td></tr></table>			Step	Actor Action	System Response	1	In the home page, Actor clicks on a sale task card	System show details of the sale task in these fields: <ul style="list-style-type: none"><li>- Image</li><li>- Product Name</li><li>- Description</li><li>- Price of food</li><li>- Quantity</li><li>- Button Remove &amp; Modify</li></ul>			
Step	Actor Action	System Response										
1	In the home page, Actor clicks on a sale task card	System show details of the sale task in these fields: <ul style="list-style-type: none"><li>- Image</li><li>- Product Name</li><li>- Description</li><li>- Price of food</li><li>- Quantity</li><li>- Button Remove &amp; Modify</li></ul>										
Alternative Flow:	<table><tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>In home page, actor inputs keyword to search textbox</td><td>System shows a list of sale tasks containing the keyword</td></tr><tr><td>2</td><td>Actor clicks on a sale task card</td><td>System show details of the sale task in these fields:<ul style="list-style-type: none"><li>- Image</li><li>- Product Name</li><li>- Description</li><li>- Price of food</li><li>- Quantity</li><li>- Button Remove &amp; Modify</li></ul></td></tr></table>			Step	Actor Action	System Response	1	In home page, actor inputs keyword to search textbox	System shows a list of sale tasks containing the keyword	2	Actor clicks on a sale task card	System show details of the sale task in these fields: <ul style="list-style-type: none"><li>- Image</li><li>- Product Name</li><li>- Description</li><li>- Price of food</li><li>- Quantity</li><li>- Button Remove &amp; Modify</li></ul>
Step	Actor Action	System Response										
1	In home page, actor inputs keyword to search textbox	System shows a list of sale tasks containing the keyword										
2	Actor clicks on a sale task card	System show details of the sale task in these fields: <ul style="list-style-type: none"><li>- Image</li><li>- Product Name</li><li>- Description</li><li>- Price of food</li><li>- Quantity</li><li>- Button Remove &amp; Modify</li></ul>										
Exceptions:												

	No	Cause	System Response
	1	There is no sale tasks containing the keyword	System shows a message: "No data."
Business Rules:	<ul style="list-style-type: none"> <li>- If Quantity is equal to 0, the sale task card shows the words out of stock on the image.</li> <li>- If the food is expired, the sale product can only be viewed or removed</li> </ul>		
Assumptions:	Actor's internet connection is available		

### UC-27: Create new Sale Product

ID and Name:	UC-27: Create new Sale Product		
Created By:	PhucLM	Date Created:	11/10/2022
Primary Actor:	Distributor	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Often
Trigger:	The actor wants to release near-expired food to customers		
Description:	This feature allows actor to fill in information of near-expired food then push to dashboard		
Preconditions:	The actor login to Dissovery System as Distributor by account that has Distributor role		
Post-conditions:	Success: The system shows messages: " Add successfully ", " Thank you for your request.We will respond to you soon ".		
Normal Flow:	Step	Actor Action	System Response
	1	In the task bar menu, the actor clicks on the home page.	System gets to the product page and show full list of products that they and other distributors have created

	2	The actor clicks on the “ + SALE REQUEST “ button.	System shows a form of adding information of products including name, description ,quantity, original price, sales price, produced & expired date
	3	Actor clicks on a field Product Name ,Description, Price, Quantity [Alternative 1]	System shows keyboard to modify the field
	4	Actor clicks on a field Produced and Expired date. [Alternative 2]	System gives the actor a calendar.
	5	The actor fills in all the textboxes of the form and clicks the “Create” .	System creates the sale request and changes to the next page, notice “Create successfully !”
Alternative Flow:			
	Step	Actor Action	System Response
	1	Actor types the keyboard	System shows what the actor types from the keyboard on each respective field.
	2	Actor selects each field and drags up or down to select what the actor wants.	System displays according to the information that the actor drags up or down.
Exceptions:			
N/A			

Business Rules:	<ul style="list-style-type: none"> <li>- Default view of the list of products is those created by the actor.</li> <li>- The products are arranged by time.</li> </ul>
Assumptions:	Actor's internet connection is available

## UC-28: Modify Own Sale Product

ID and Name:	UC-29: Modify Own Sale Product		
Created By:	PhucLM	Date Created:	11/10/2022
Primary Actor:	Distributor	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Often
Trigger:	The actor wants to modify sale task		
Description:	This feature allows actor to modify the sale tasks created by Distributor		
Preconditions:	The actor must login to their account as a “Distributor” role.		
Post-conditions:	Success: System shows message: “The sale tasks have been updated” and shows the new information. Fail:		
Normal Flow:			
	Step	Actor Action	System Response
	1	In home page, Actor clicks on a sale task card	System show details of the sale task in these fields: <ul style="list-style-type: none"><li>- Image</li><li>- Product Name</li><li>- Description</li><li>- Price of food</li><li>- Quantity</li></ul>

	2	Actor clicks on a field Product Name ,Description [Alternative 3]	System shows keyboard to modify the field
	3	Actor clicks on a field Price, Quantity [Alternative 4]	System shows the plus and minus button, the attribute number of this product.
	4	Actor clicks on a field Produced and Expired date. [Alternative 5]	System gives the actor a calendar.
	5	Actor clicks "Save" button	System saves the new information and changes to the next page, notices message: "The sale tasks has been updated"
Alternative Flow:			
	Step	Actor Action	System Response
	1	In home page, Actor inputs keyword into search textbox	System shows a list of sale tasks containing the keyword
	2	Actor clicks on a sale task card	System show details of the sale task in these fields: <ul style="list-style-type: none"> <li>- Image</li> <li>- Product Name</li> <li>- Description</li> <li>- Price of food</li> </ul>



			- Quantity						
	3	Actor types the keyboard	System shows what the actor types from the keyboard on each respective field.						
	4	Actor selects the plus and minus buttons	System displays the increase or decrease number corresponding to the field.						
	5	Actor selects each field and drags up or down to select what the actor wants.	System displays according to the information that the actor drags up or down.						
Exceptions:	<table><tr><th>No</th><th>Cause</th><th>System Response</th></tr><tr><td>1</td><td>There is no sale task containing the keyword</td><td>System informs “No data”</td></tr></table>			No	Cause	System Response	1	There is no sale task containing the keyword	System informs “No data”
No	Cause	System Response							
1	There is no sale task containing the keyword	System informs “No data”							
Business Rules:	<ul style="list-style-type: none"><li>- Except for Description, other fields can’t be blank.</li><li>- Price must be higher than 0.</li><li>- Quantity must be higher or equal 0.</li><li>- If the food is expired, the sale task can only be viewed or removed</li></ul>								
Assumptions:	Actor’s internet connection is available								

#### UC-29: Delete Own Sale Tasks

<b>ID and Name:</b>	<b>UC-30: Remove Own Sale Product</b>		
Created By:	PhucLM	Date Created:	11/10/2022

Primary Actor:	Distributor	Secondary Actors:	N/A
Priority:	Normal	Frequency of Use:	Often
Trigger:	The actor wants to remove sale task .		
Description:	This feature allows actors to remove a sale task created by a Distributor.		
Preconditions:	Actor login into an App with an account has a distributor role.		
Post-conditions:	Success: System displays a message “Delete successfully !”.		
Normal Flow:			
	Step	Actor Action	System Response
	1	Actor clicks on a sale task card	System show details of the sale task in these fields: <ul style="list-style-type: none"> <li>- Image</li> <li>- Product Name</li> <li>- Description</li> <li>- Price of food</li> <li>- Quantity</li> </ul>
	2	Actor clicks “Remove” [Alternative 1,2]	System shows a message “ Are you sure you want to remove this product ? ”
Alternative Flow:			
	Step	Actor Action	System Response
	1	Actor clicks “Yes”	System deletes the product and changes to the next page, notice “Delete successfully !”
	2	Actor clicks “No”	System returns to the page detail product.

Exceptions:	N/A
Business Rules:	- Actor is sure of this decision because the actor wants to display this product, and the actor has to create a sale request again.
Assumptions:	Actor's internet connection is available

### UC-30: View Latest Order List

ID and Name:	UC-30: View latest Order List		
Created By:	NguyenTH	Date Created:	11/10/2022
Primary Actor:	Distributor	Secondary Actors:	N/A
Priority:	High	Frequency of Use:	Often
Trigger:	The actor wants to view order that has been checked out		
Description:	This feature allows the actor to view the order that has been made in the past by customers sorted by date from nearest		
Preconditions:	- The actor must login to their account as “Distributor” role		
Post-conditions:	Success: System shows a list of created order Fail: The system shows message “No data”		
Normal Flow:			
	Step	Actor Action	System Response
	1	In home page, actor clicks on profile section	System redirects to profile page containing view history action.
	2	Actor clicks “View Order” button	System redirects history page showing a list of created order sorted by date from nearest
Alternative Flow:	N/A		

Exceptions:			
	No	Cause	System Response
	1	There is no order.	System informs “No data”
Business Rules:	N/A		
Assumptions:	Actor’s internet connection is available		

### UC-31: View Latest Order Detail

ID and Name:	UC-31: View Latest Order detail		
Created By:	QuangVDM	Date Created:	11/10/2022
Primary Actor:	Distributor	Secondary Actors:	N/A
Priority:	Normal	Frequency of Use:	Often
Trigger:	The actor wants to view details of an order.		
Description:	This feature allows actors to view detail of orders created by customer		
Preconditions:	<ul style="list-style-type: none"><li>- Guests must login and use the function as “Distributor”.</li><li>- The actor has checked out at least 1 card before.</li></ul>		
Post-conditions:	Success: System shows details of an order card.		
Normal Flow:			
	Step	Actor Action	System Response
	1	In home page, actor clicks on profile section	System redirects to profile page containing view history action.

	2	Actor clicks “View Order” button	System redirects history page showing a list of created order sorted by date from nearest
	3	Actor clicks an order card	System redirects to detail page showing: <ul style="list-style-type: none"> <li>- Food card(name, number, price,...)</li> <li>- Total Price</li> <li>- Phone number</li> <li>- Address</li> <li>- Date created</li> <li>- Processing status(paid, Pending, Canceled)</li> </ul>
Alternative Flow:	N/A		
Exceptions:	N/A		
Business Rules:	<ul style="list-style-type: none"> <li>- Actor’s payment method must be valid.</li> <li>- After synchronisation, actor has to wait 1 month after next alteration.</li> </ul>		
Assumptions:	Actor’s internet connection is available		

## 4. Non-Functional Requirement

### 4.1. Usability

- All text, messages should be written in English.
- The mobile app Interface should follow the Figma prototypes’ design.
- Each perspective of the app must load within 2 seconds.

- The user doesn't need specialized skills to use the app.
- The error rate of users submitting their payment details at the checkout page mustn't exceed 10 percent
- The app can be downloaded and used as long as there is internet connection.
- The user should be able to understand all the functions within 1 hour of self-experiencing.

#### **4.2. Reliability**

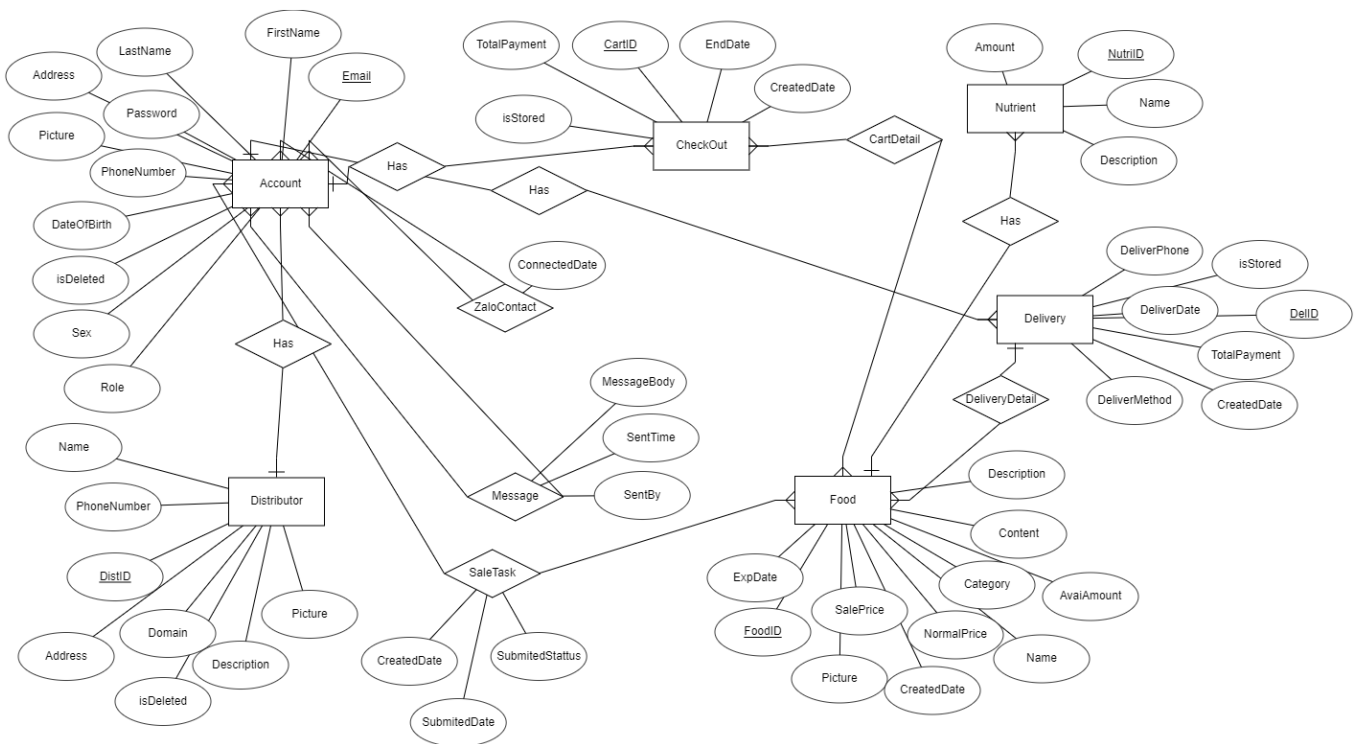
- The availability of the system is over 85%.
- Mean Time Between Failures (MTBF) should be more than 125 hours.
- Mean Time To Repair (MTTR) should be less than 12 hours .
- Precision in the system's output must be 100%.
- Response time for a transaction must be 5 seconds or less
- Capacity for the number of users the system can accommodate is at least 500 users.
- Resource utilization, such as memory, disk, communications, and so forth must be smoothie and with minimal error.

#### **4.3. Performance**

- System should return result for each request within 5 seconds
- Processing time is from 03 to 05 seconds for each frame.

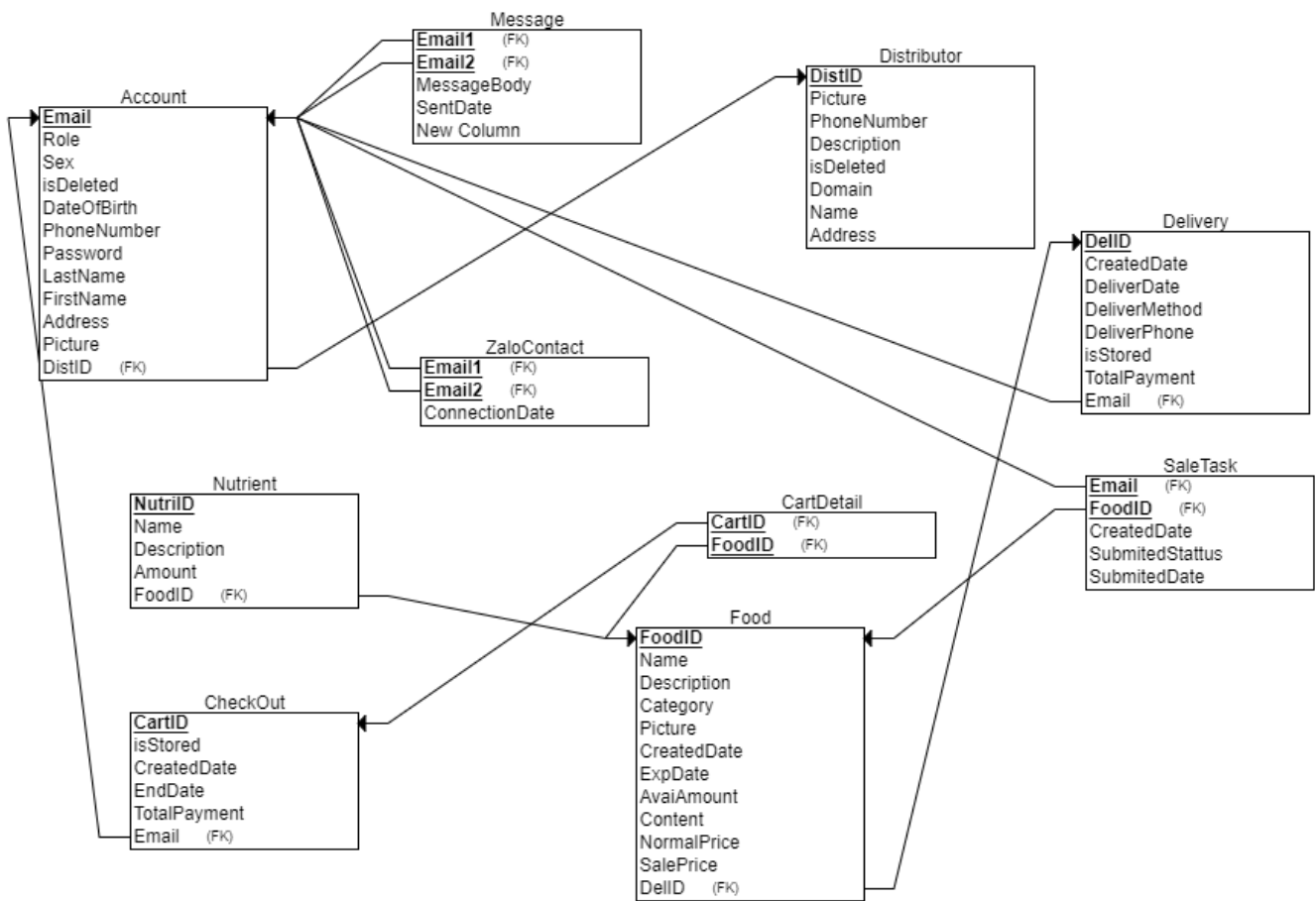
## 5. Database


### 5.1. ERD



Full Resolution:  Dissovery\_ERD.png

### 5.2. Relational Schema



Full Resolution:  Dissovery\_Schema.png



