This Form is for Both the General & Medical Informatics Programmes

SE-I Course Project (Phase 2 Cover Sheet)

Discussions Scheduled for Week 12 *(more details will be announced later)***.**

- Print 1 copy of this cover sheet and attach it to a printed copy of the documentation *(SRS, ... etc.)*. You must also submit softcopies of all your documents *(as PDFs)*; details will be announced later.
- Please write all your names in Arabic.
- Please make sure that your students' IDs are correct.
- Handwritten Signatures for the attendance of all team members should be filled in before the discussion.
- Please attend the discussion on time *(announced separately)*, late teams will lose 5 grades.

Project Name: **An Application for Online Food & Grocery Delivery Service [Inspired by Talabat]**

Team Information ***(typed not handwritten, except for the attendance signature)*****:**

	ID [Ordered by ID]	Full Name [In Arabic]	Attendance [Handwritten Signature]	Final Grade
1	20210188	اندرو ايمن انسي عبد الملاك		
2	20210104	احمد محمد عبد المعز عبدالستار		
3	20210133	ادم محمد عطيه عبدالغني		
4	20210142	اسامة محمود عز الدين محمد عبدالله ربيع		
5	20210151	اسراء سعد حافظ محمد		
6	20210153	اسراء علي سيد محمود		
7	20210137	اروى شريف محمد احمد		

Grading Criteria:

5 Items		**Grade**	**Notes**
1. Updated Use-case Diagrams & Descriptions, Activity Diagrams, Object Diagrams, Sequence Diagrams, System Architecture	**2**		
2. Collaboration/Communication Diagrams	**2**		
3. Class Diagram *(3 versions)* 1) An initial version based on the requirements and Use-Case/Activity diagrams. [Submitted in phase 1] **2) An intermediate version based on the interaction diagrams.** **3) A final version, after applying the design patterns and any other modifications.**	**3**		
4. Three Mandatory Design Pattern Applied *(Including a typed description)*	**4**		
5. Front End Design for all Functions *(HTML, Bootstrap)*	**2**		
6. Implementation <u>based on the submitted Requirements & Design</u>. Should include at least 4 of the following modules (in addition of course to modules specific to your individual projects): **1) User Role Management Module.** **2) User manipulation Module** *(Login, Add / Delete / Update / Search, List).* **3) Controlling Resources Module** *(Rooms, Orders, Products, … etc.).* **4) Reservation and Rescheduling Module.** **5) Generating Reports Module** *(PDFs, … etc.).* **6) Sending Emails or Notifications Module.**	**7**		

N.B. .. **<u>You must update and resubmit the initial part of the documentation submitted in phase 1</u>** (including the Functional / Non-Functional requirements, Use-case Diagrams & Descriptions, Activity Diagrams, Class Diagram, Object Diagrams, Sequence Diagrams, System Architecture, .. etc.).

Teaching-Assistant's Signature: ________________________

$\frac{\quad}{20}$

1.Introduction :

In today's fast-paced world, people are always on the go and have very little time to spare for activities such as grocery shopping and cooking. To address this need, online food and grocery delivery services have become increasingly popular. In this project, we will be presenting an application for an online food and grocery delivery service inspired by Talabat. Our goal is to provide a convenient and seamless experience for customers to order food and groceries from the comfort of their homes.

1.1 System overview

Our online food and grocery delivery service is a web-based system that consists of a customer interface, a vendor interface, and a delivery management system. Customers can order food and groceries through the customer interface, vendors can manage their products and orders through the vendor interface, and the delivery management system handles the delivery process. The system provides a convenient and efficient solution for customers to order and receive food and groceries from the comfort of their homes.

2– Requirements Specification

2.1 Functional Requirements

1. The app must allow the admin to manage users and restaurants profiles.
2. The app must allow the admin to add restaurant, delete restaurant, update restaurant profile.
3. The app must allow the admin to add or edit on the restaurant's menu.
4. The app must allow the admin to view customer profile, delete customer profile.
5. The app must allow the admin to add w new delivery boy to be assigned by a customer in ordering Process.
6. The app must allow the user to sign up if he do not have an account or log in if he has one.
7. The app must allow the user to update his profile information.
8. The app must allow the customer to add his location.
9. The app must allow customers to discover and search for restaurants.

10. The app must allow customers to select a restaurant.

11. The app must allow customers to view the menu, prices, and images of food items from each restaurant.

12. The app must allow customers to place an order.

13. The app must allow customers to show the cart and edit or remove from it.

14. the app must allow customers to checkout from the cart with a selected payment method (credit, cash)

15. The app must allow customers to track the status of their order.

16. The app must allow customers to leave feedback.

17. The app must allow customers to add restaurant to their favourite list.

18. The app must allow customers to access their order history and favourite restaurants.

19. The app must provide the No Cutlery feature that lets customers option out of receiving disposable cutlery with their order.

20. The app must provide the GEM feature that shows the customers offers and recommendations from restaurants.

21. The app must allow the customer to assign the order to the available delivery partners.

22. The app must have a FAQ page to help the user.

23. The app must have a contact us page to help the user to connect.

2.2 Non-Functional Req:

Performance Requirements

- The website should provide greater performance with no delay. For food, who would be using web app on their desktop or phones, the performance should be good.
- The system must respond to the operation for user (admin) quickly.
- The application shall show success message if the operation was done correctly.
- The system should be compatible with all modern browsers.
- The system should respond to the operation messages to the users quickly.

Safety Requirements

- The system must handle safe login and logout through session.
- The database should be secured from SQL injection to prevent leak or loss of information.
-

Reliability

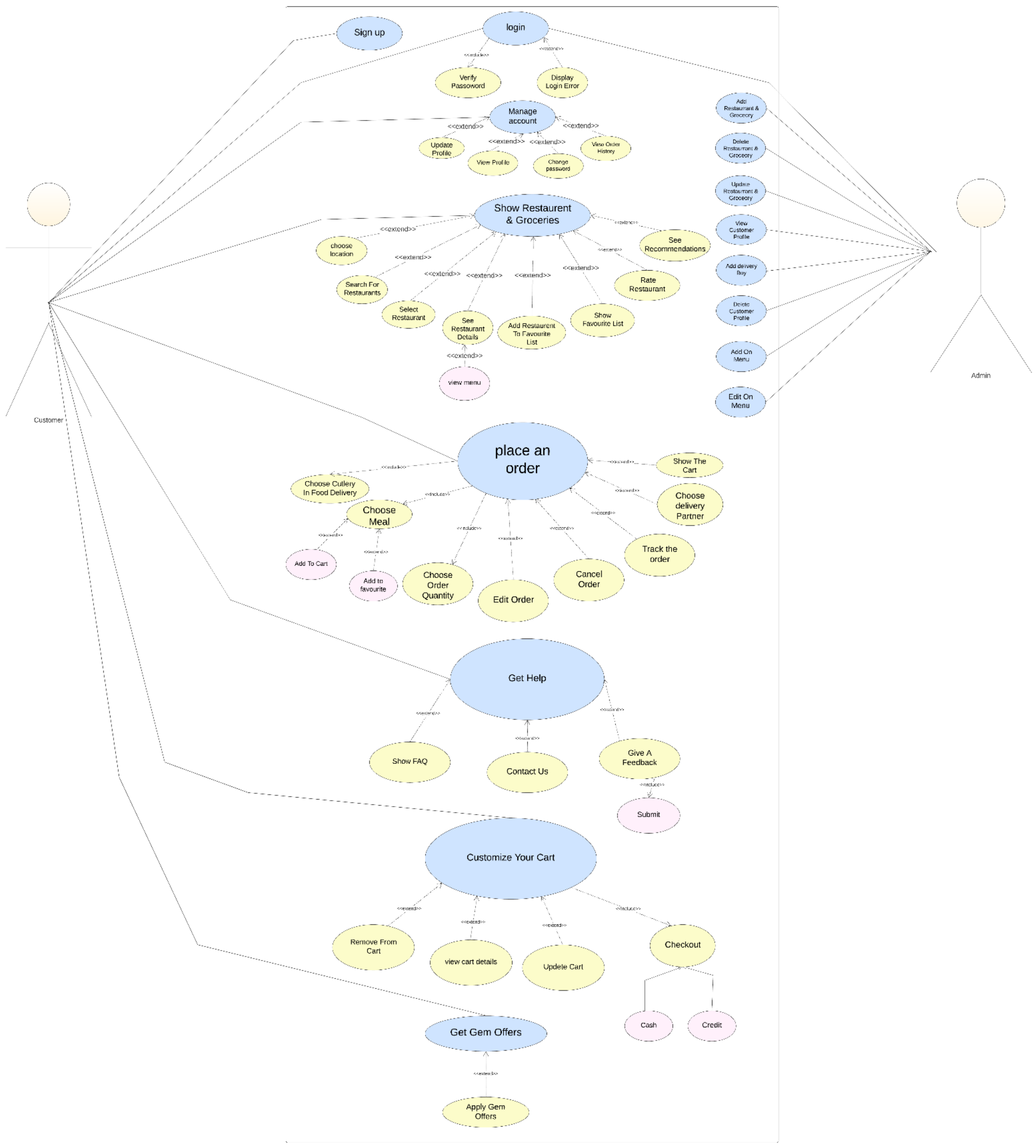
- The website should be reliable and provide catching of exceptions so that unintended results do not occur such as system crashes, or data validation failures.

Usability

- UI designed in easy way, so the user can use the system smoothly after spending an hour on it.

3 - Use Case:

3.1 Use case diagram



3.2 detailed Use case

1-sign up

Name	**Sign up**
Intiatior	Customer.
Descritption	The system allows the customer to create an account.
Pre-Condition	The customer does not have an account
Post-Condition	an account has been created
Main success scenario	1- The customer opens the website. 2- The customer creates an account. 3- An account has been created.
Alternative scenario	1-the system fails to save the customer data to database 2-the customer enter invalid information

2- Log In

Name	**Log In**
Intiatior	Admin and customer.

Descritption	The system will allow the user to log in with username and password.
Pre-Condition	The user opens the website.
Post- Condition	The user have his access on his profile on website.
Main success scenario	1. The user opens the website. 2. The user enters his username and password. 3. If his information is correct, he will enter the website successfully
Alternative scenario	1. The user enters wrong information. 2. The system can't verify the entered data

3- View customer

Name	**View customer**
Intiatior	Admin.
Descritption	The system will allow the admin to view a choosen customer
Pre-Condition	The admin logged in
Post- Condition	The customer profile is displayed to the admin

Main success scenario	1. The admin opens the website 2. The admin logged in. 3. The admin selects to view the customer profile

4- Delete customer

Name	**Delet customer**
Intiatior	Admin.
Descritption	The system will allow the admin to delete a choosen customer
Pre-Condition	The admin logged in and view customer
Post- Condition	The customer profile is succssfully deleted .
Main success scenario	1. The admin opens the website. 2. The admin logged in. 3. The admin views a customer. 4. The admin selects to delete this customer
Alternative scenario	1.admin can not access the customer profile 2. the sytem doesn't allow the admin to delete the customer profile

5-search restaurant

Name	**Search restaurant**
Intiatior	Customer
Descritption	The system allow the customer to search for a restaurant.
Pre-Condition	The customer logged in if he has an account or sign up if he has not.
Post-Condition	The system display the restaurant the customer searched for.
Main success scenario	1- The customer opens the website. 2- The customer signs up or log in if he already has an account. 3- The customer decides to search for his restaurant
Alternative scenario	1- The customer cannot find the restaurant he want. 2- The system fails to show the search bar to the customer.

6-see recommendations

Name	**See recommendations**
Intiatior	Customer.

Descritption	The system allows the customer to ask for a recommendation
Pre-Condition	The customer logged in if he has an account.
Post-Condition	The system displays the recommended restaurants.
Main success scenario	1- The customer opens the website. 2- The customer logs in if he has an account. 3- The customer asks for recommendation. 4- The system displays the recommended restaurants.
Alternative scenario	1- The system doesn't display any recommendations.

7- choose location

Name	**Choose location**
Intiatior	Customer.
Descritption	The system will ask the user about the location then the system will save the information to the database.
Pre-Condition	The customer opens the webtsite.
Post- Condition	customer write his location and saved by the system

Main success scenario	1. The customer opens the website. 2. The customer logged in. 3. The system ask customer for his location 4. The customer adds his location 5. The system submits his location
Alternative scenario	1. The customer cannot find his location in map 2. The system can't submit the customer location

8-select restaurant

Name	**Select restaurant**
Intiatior	Customer.
Descritption	The system shows the customer alot of restaurant to select from
Pre-Condition	The customer opens website and choose from restaurants.
Post-Condition	the customer selected the restaurant successfully and can access menu to order .
Main success scenario	1. The customer logged in successfully. 2. The system allows the customer to search restaurant. 3. The system allows the customer to select any restaurant
Alternative scenario	1.the system doesn't allow the customer to select a resturant

9-Select meal

Name	**Select meal**
Intiatior	Customer
Descritption	The System must allow the user to Choose a meal from a menu of food
Pre-Condition	the customer enter to the website and choose the restaurant.
Post-Condition	The customer choose his meal
Main success scenario	1- The customer opens the website. 2- The customer log in 3- The customer chooses a restaurant. 4- The system allows the customer to choose a meal. 5- The system shows the customer the selected meal
Alternative scenario	1- The customer can't find the meal he wants. 2- The meal was sold out.

10-choose delivery boy

Name	**Choose delivery boy**

Intiatior	Customer.
Descritption	The system allow the customer to choose a delivery boy.
Pre-Condition	The customer select a restaurant , order and add it to cart
Post-Condition	The customer has chosen the delivery boy he need.
Main success scenario	1- The customer selects a restaurant. 2- The customer selects an order. 3- The customer adds the order to his cart. 4- The customer chooses the delivery boy from the options.
Alternative scenario	1- The customer cannot find any available delivery boy.

11- Toggle-not-cutlery

Name	**Toggle-not-cutlery**
Intiatior	Customer
Descritption	The system allow the customer to decide he need a cutlery or not.
Pre-Condition	The customer select a restaurant and his meal

Post-Condition	The system has known he need a cutlery or not.
Main success scenario	1- The customer selects a restaurant. 2- The customer selects an order. 3- The customer adds the order to his cart. 4-The customer decide he need cutlery or not.
Alternative scenario	1- The system fails to show the option to the customer.

12-Add to cart

Name	**Add to cart**
Intiatior	Customer.
Descritption	The system allows the customer to add his order to cart
Pre-Condition	The customer choose a restaurant and a meal
Post-Condition	the customer added his meal to cart
Main success scenario	1- The customer opens the website. 2- The customer logged in. 3- The customer chooses the restaurant. 4- The customer chooses and add the meal to cart.

Alternative scenario	1- The customer can not find his order in the cart 2- The system doesn't response to the customer instractions

13-remove from cart

Name	**Remove from cart**
Intiatior	Customer
Descritption	The system allow the customer to remove a product from his cart
Pre-Condition	The customer select a restaurant and his order and add it to cart
Post-Condition	The product is removed from the cart.
Main success scenario	1- The customer selects a restaurant. 2- The customer selects an order. 3- The customer adds the order to his cart. 4- The customer removes the order from his cart.
Alternative scenario	1- The customer cannot find his order in the cart. 2- The system doesn't response to the customer instructions

14- cancel order

Name	**Cancel order**
Intiatior	Customer.
Descritption	The System must allow the customer to Cancel his Order at any time**.**
Pre-Condition	The customer must have opened the cart
Post-Condition	The order will be deleted from the cart
Main success scenario	1. The customer chooses the restaurant. 2. The customer chooses the meal. 3. The customer opens the cart. 4. The system allows to user to delete the order from cart. 5-the order is canceled
Alternative scenario	1.the system doesn't response to the order canceling process

15-check out

Name	**Check out**

Intiatior	Customer.
Descritption	The customer can choose the food and add to the cart and then choose the way who want pay by it
Pre-Condition	The customer would choose the food and choose the way of payment.
Post-Condition	The order will be saved
Main success scenario	1-the customer choose the food 2-add the food to cart 3-choose way of payment 4-system will save the information
Alternative scenario	1-the customer entered a wrong card number 2-the card hasn't enough sufficient funds 3-the system fails to save the information

16-track order

Name	**Track order**
Intiatior	Customer.
Descritption	The system allows the customer to track his order.
Pre-Condition	The customer will know about his order status.
Post- Condition	customer can follow his order untill it reach

Main success scenario	1. The customer chooses the order. 2. The customer can enable gps 3. Then can follow his order
Alternative scenario	1.the system fails to show the status of the order

17- rate restaurant

Name	**Rate restaurant**
Intiatior	Customer.
Descritption	The System must allow customer to Rate The Restaurant and the Service.
Pre-Condition	The customer choose the restaurant and the meal, and his order is ended.
Post-Condition	The customer gives his opinion of the restaurant
Main success scenario	1- The customer chooses the restaurant. 2- The customer chooses the meal. 3- The order has ended. 4- The system allows the customer to write his feedback.
Alternative scenario	1-the system fails to save the rate that have been submitted by the user

18- Add to fav

Name	**Add to fav**
Intiatior	customer.
Descritption	The system will allow the customer to add a restaurant to his favorite list
Pre-Condition	The customer logged in and select a restaurant
Post- Condition	The customer added a restaurant to his favorite list .
Main success scenario	1. The customer opens the website. 2. The customer logged in. 3. The customer search for a restaurant 4. The customer selects a restaurant. 5. The customer adds this restaurant to his favorite list.
Alternative scenario	1.the customer could not select the restaurant 2.the customer could not put the restaurant to his favorite list 3.the system doesn't response to customer instraction

19-feedback

Name	**Feedback**

Intiatior	Customer.
Descritption	the customer record reaction to a restaurant, food, or performance of the service
Pre-Condition	The customer must log in to the website
Post-Condition	feedback sent to admin
Main success scenario	1. The customer logs in 2. The customer writes feedback. 3. Submit
Alternative scenario	1-the system fails to save the feedback 2-the system doesn't allow the customer to leave his feedback

20-Apply gem offers

Name	**Apply gem offers**
Intiatior	Customer.
Descritption	The system allows the customer to use the gem offers.
Pre-Condition	The customer logged in and check gem offers
Post-Condition	The customer has an offer in his order .

Main success scenario	1- The customer opens the website. 2- The customer logs in if he has an account. 3- The customer checks gem offers. 4- The customer applies gem offers.
Alternative scenario	1-the customer didn't find any available offers. 2-the system doesn't provide any offers.

21- Add Restaurant

Name	**Add Restaurant**
Intiatior	Admin.
Descritption	The system will allow the admin to add a restaurant to the restaurants list
Pre-Condition	The admin logged in and select to add restaurant
Post- Condition	The restaurant is successfully added to the restaurant list .
Main success scenario	1. The admin opens the website. 2. The admin logged in. 3. The admin selects to add a new restaurant
Alternative scenario	1-the system can't save the data 2-the system has error and can not response to admin instractions

22- Delete Restaurant

Name	**Delete Restaurant**
Intiatior	Admin.
Descritption	The system will allow the admin to delete a specific restaurant form the restaurants list.
Pre-Condition	The customer logged in and select a restaurant
Post- Condition	The restaurant is successfully deleted from the restaurant list.
Main success scenario	1. The admin opens the website. 2. The admin logged in. 3. The admin selects a restaurant. 4. The admin selects to delete this restaurant from the list
Alternative scenario	1-the system can't save the data 2-the system has error and can not response to admin Instractions 3-the admin can't select the restaurant

23- view Menu

Name	**View Menu**
Intiatior	customer.
Descritption	The system will allow the customer to view the menu of a selected restaurant

Pre-Condition	The customer logged in and select a restaurant
Post- Condition	The system displays the menu to the customer
Main success scenario	1. The customer opens the website. 2. The customer logged in. 3. The customer search for a restaurant 4. The customer selects a restaurant. 5. The customer selects to view the menu.
Alternative scenario	1.the system doesn't response to customer Instraction 2.The customer can't access the menu 3.the menu has been deleted

24- Add menu

Name	**Add menu**
Intiatior	Admin.
Descritption	The system will allow the admin to add a new menu to a selected restaurant
Pre-Condition	The admin logged in and select a restaurant
Post- Condition	The restaurant is successfully has a new menu

Main success scenario	1. The admin opens the website. 2. The admin logged in. 3. The admin selects a restaurant. 4. The admin selects to add a menu for this restaurant.
Alternative scenario	1-the system can't save the data 2-the system has error and can not response to admin Instractions 3-The admin can't find the menu he wanna add

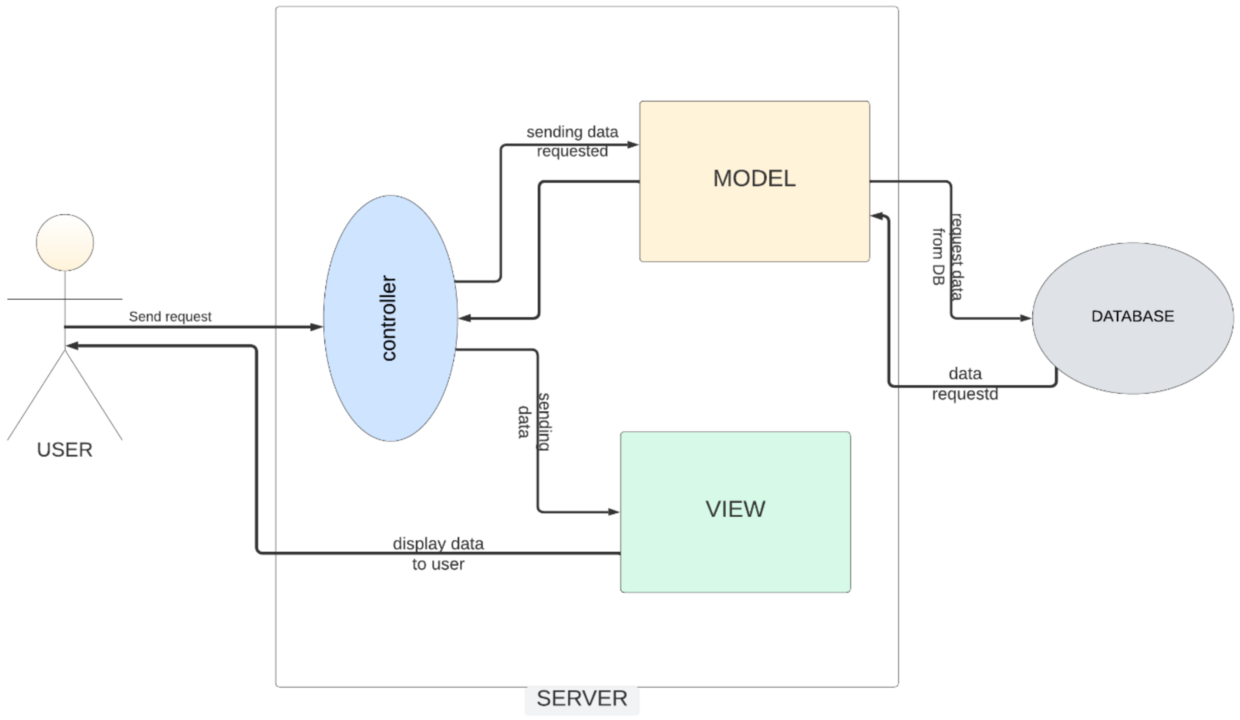
24- Edit Menu

Name	**Edit Menu**
Intiatior	Admin.
Descritption	The system will allow the admin to edit on exisited menu from a restaurant
Pre-Condition	The customer logged in and select a restaurant
Post- Condition	The selected restaurant menu is edited with the new items
Main success scenario	1. The admin opens the website. 2. The admin logged in 3. The admin select restaurant 4. The admin select to edit on a current menu
Alternative scenario	1- The system doesn't accept the modified data 2- The admin can't access the menu

25- Add Delivery

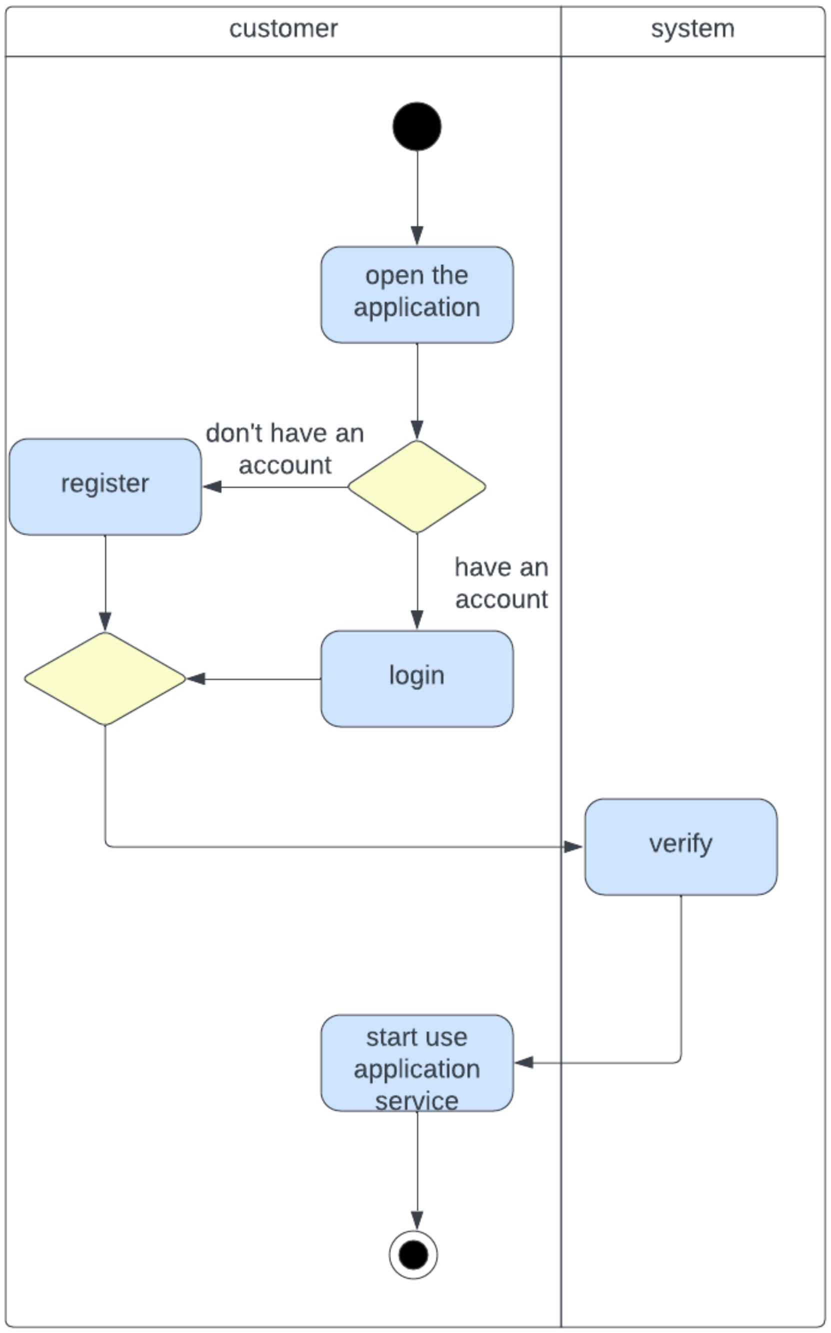
Name	**Add delivery**
Intiatior	Admin.
Descritption	The system will allow the admin to add a new delivery boy
Pre-Condition	The admin logged in
Post- Condition	The new delivery boy will be avalibale to assiggn by a customer while ordering proccess.
Main success scenario	1. The admin opens the website 2. The admin logged in 3. The admin select to add a new delivery boy
Alternative scenario	1. The system doesn’t response to the instruction

4 – System Architecture :

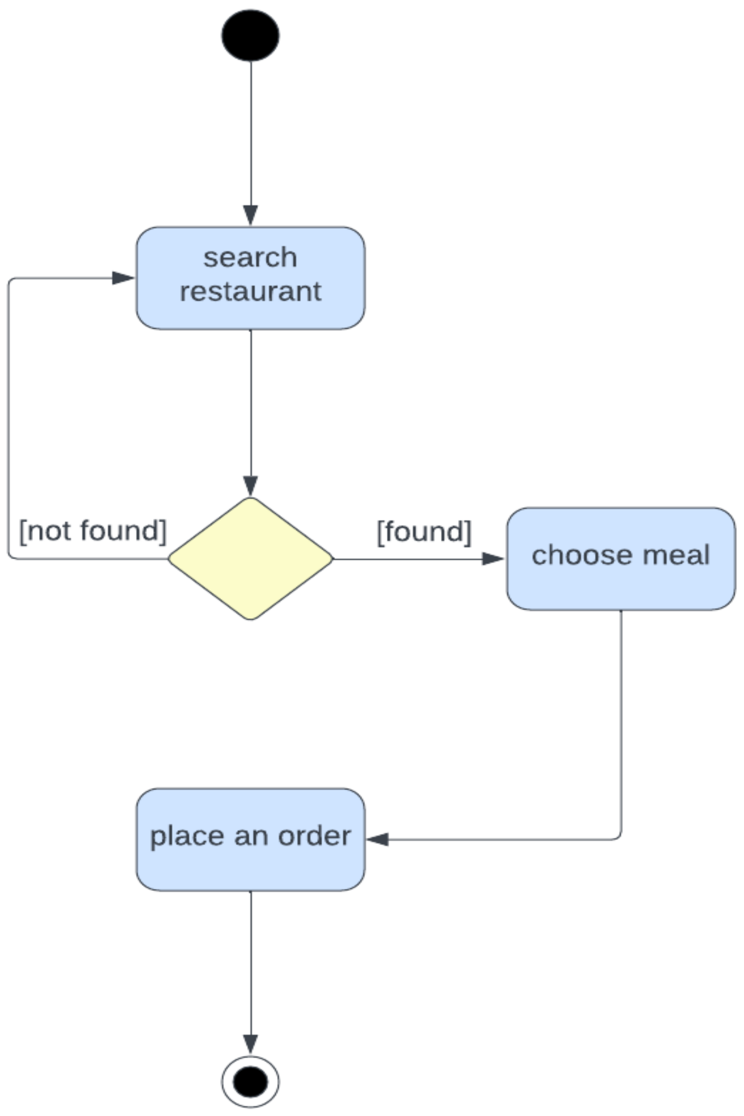


5 – Activity diagram

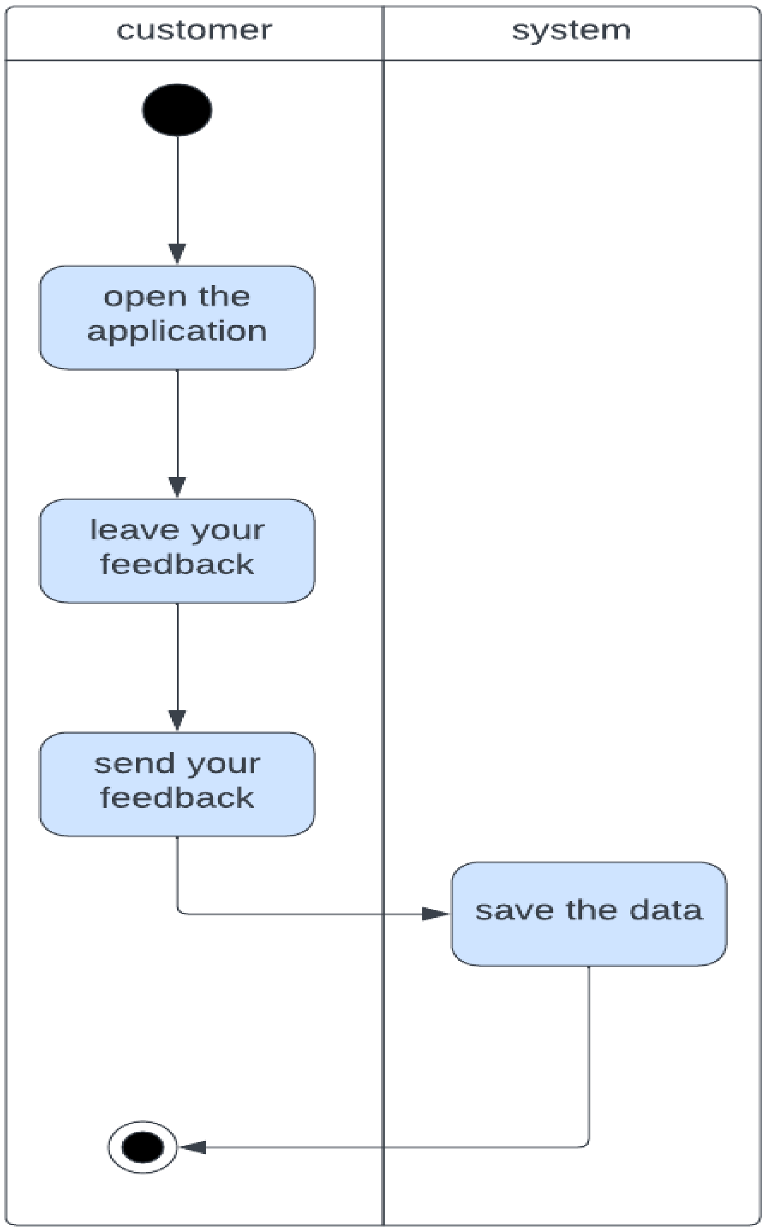
1 – Login & Sign Up



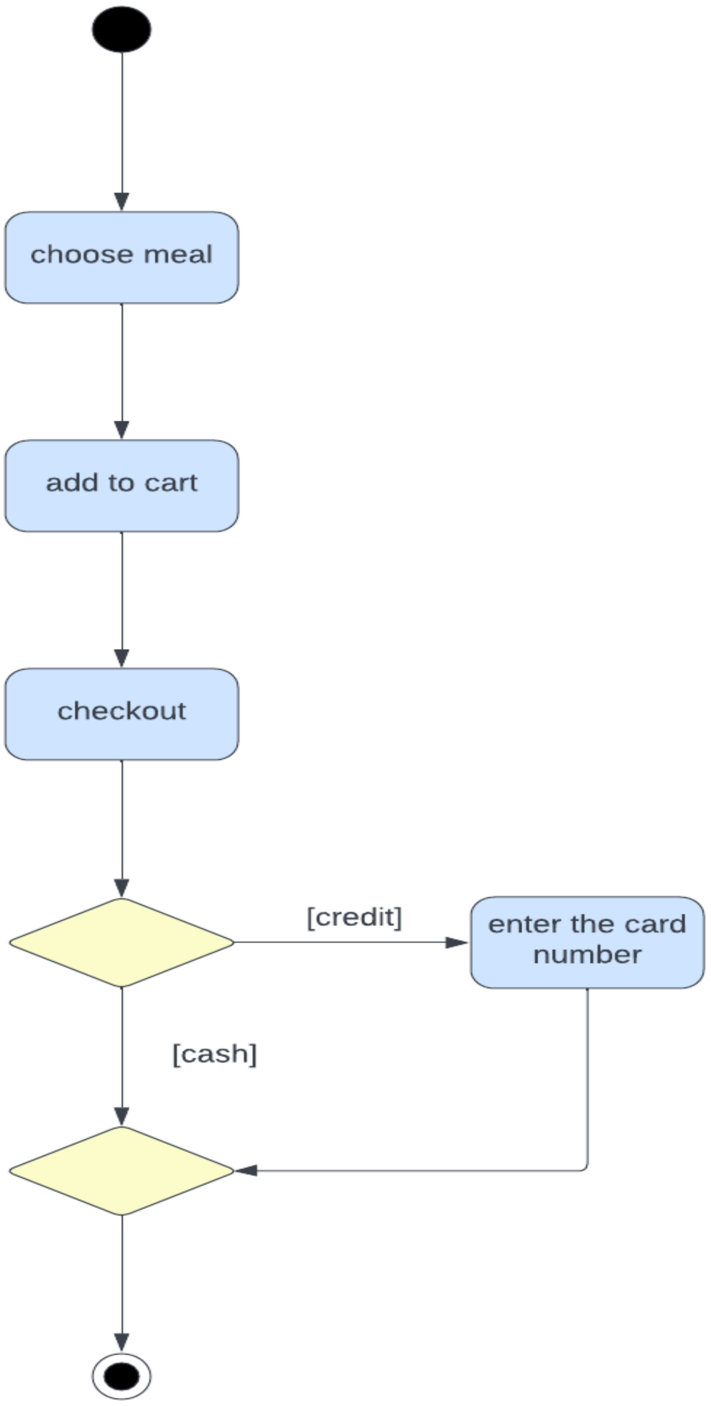
2 – Search Restaurant



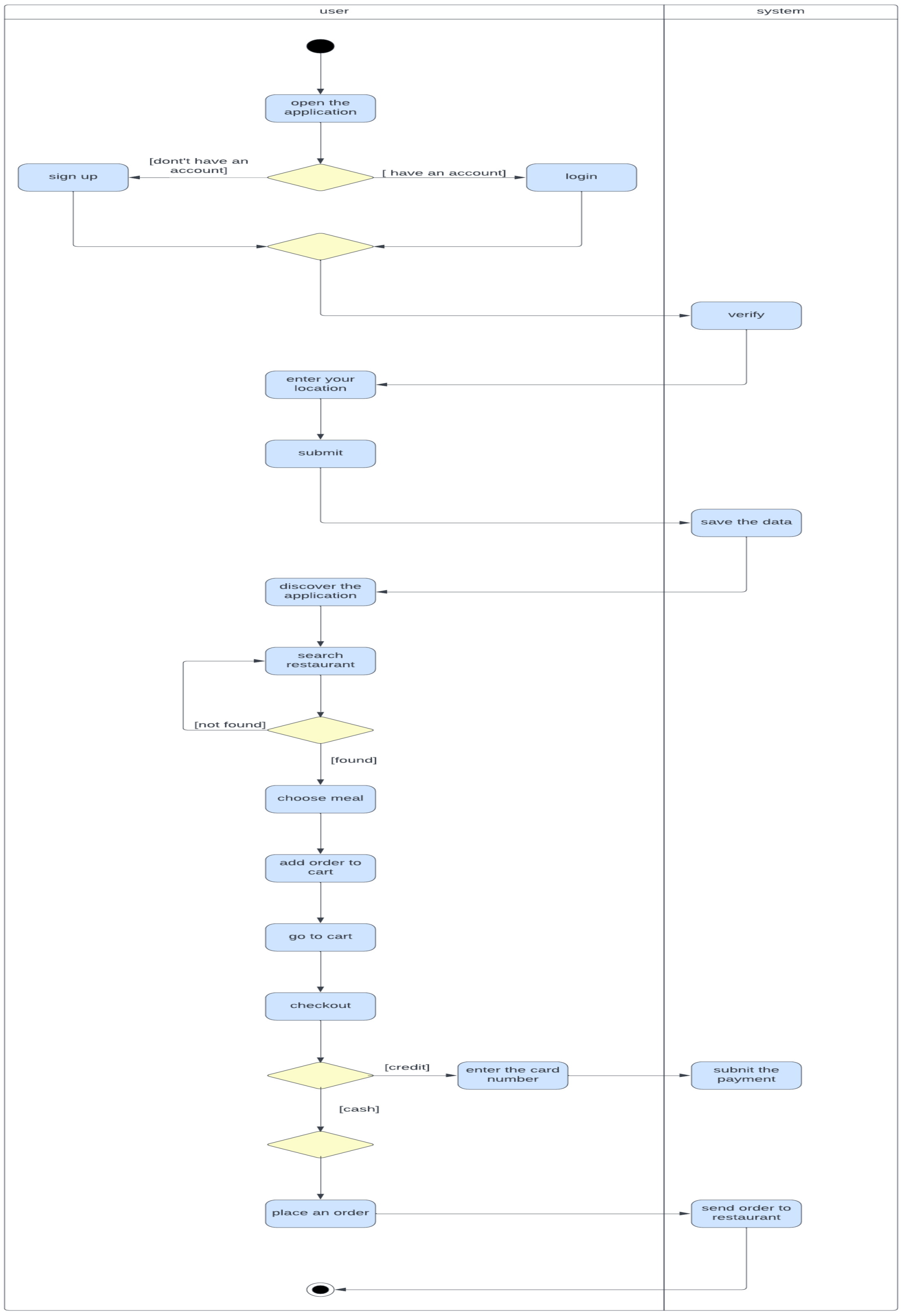
3 – feedback


customer
system
open the application
leave your feedback
send your feedback
save the data

4 – Checkout

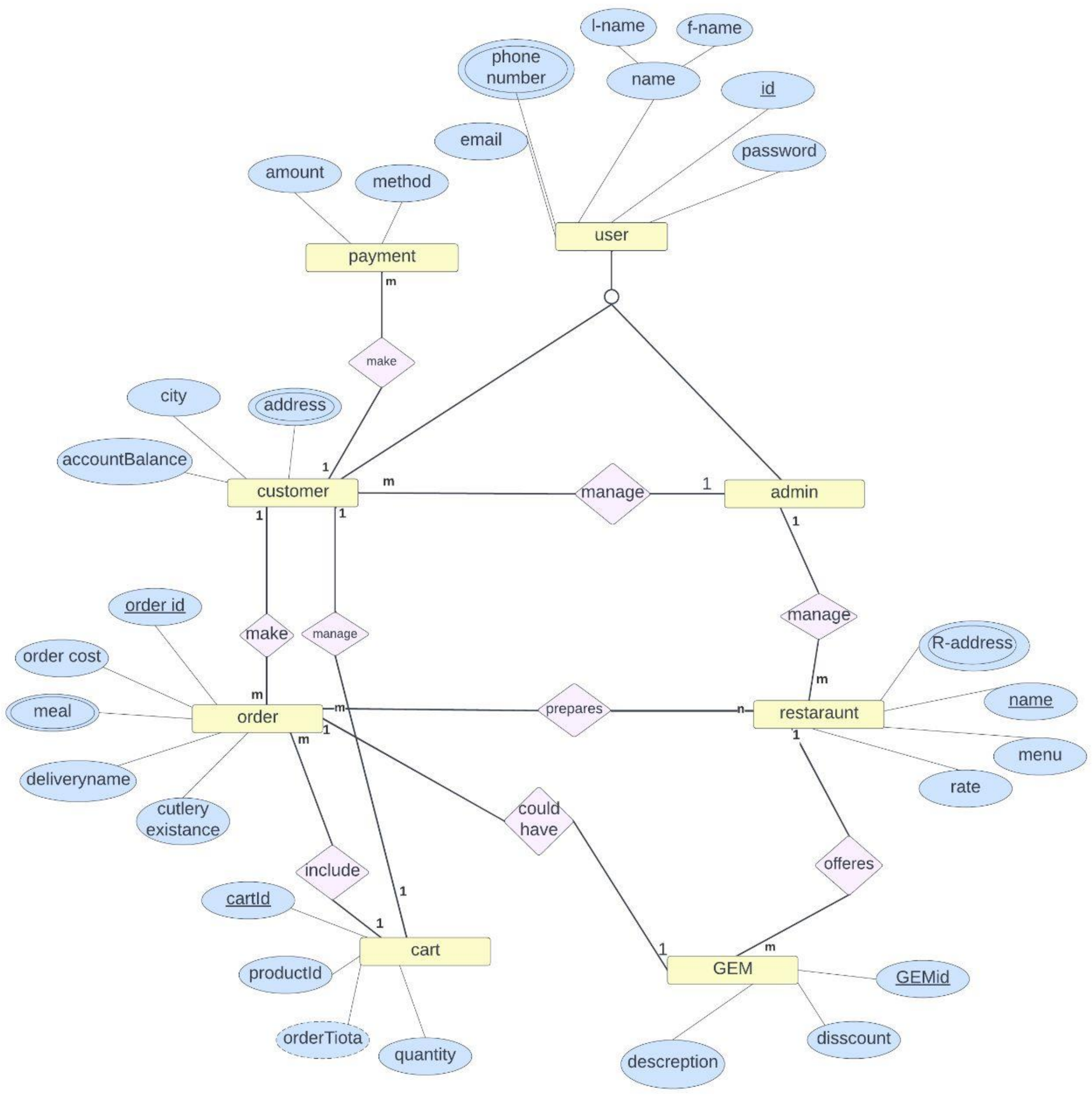

choose meal
add to cart
checkout
[credit]
enter the card number
[cash]

5 – Activity diagram

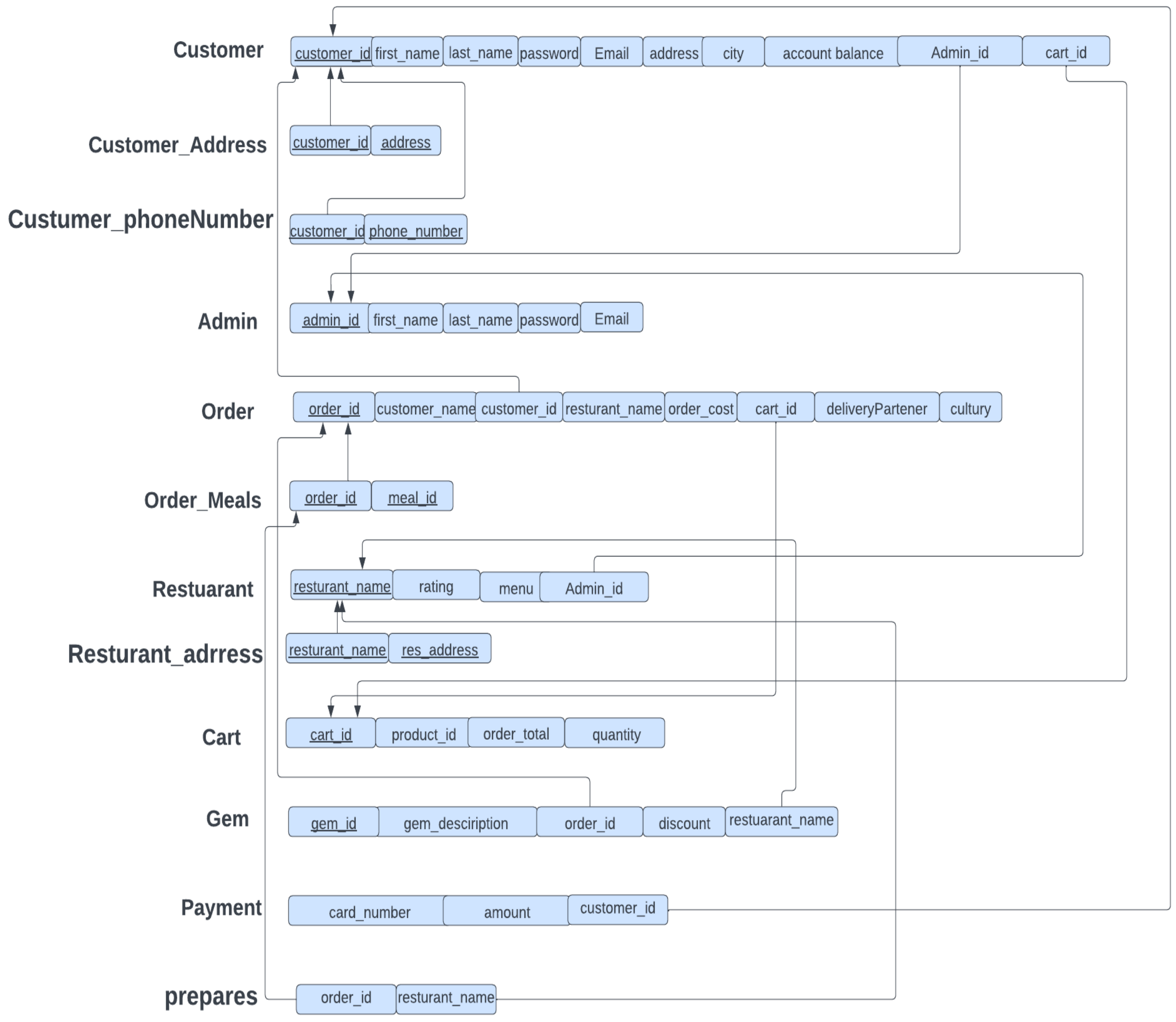


6 - Database Specification:

6.1 ERD :

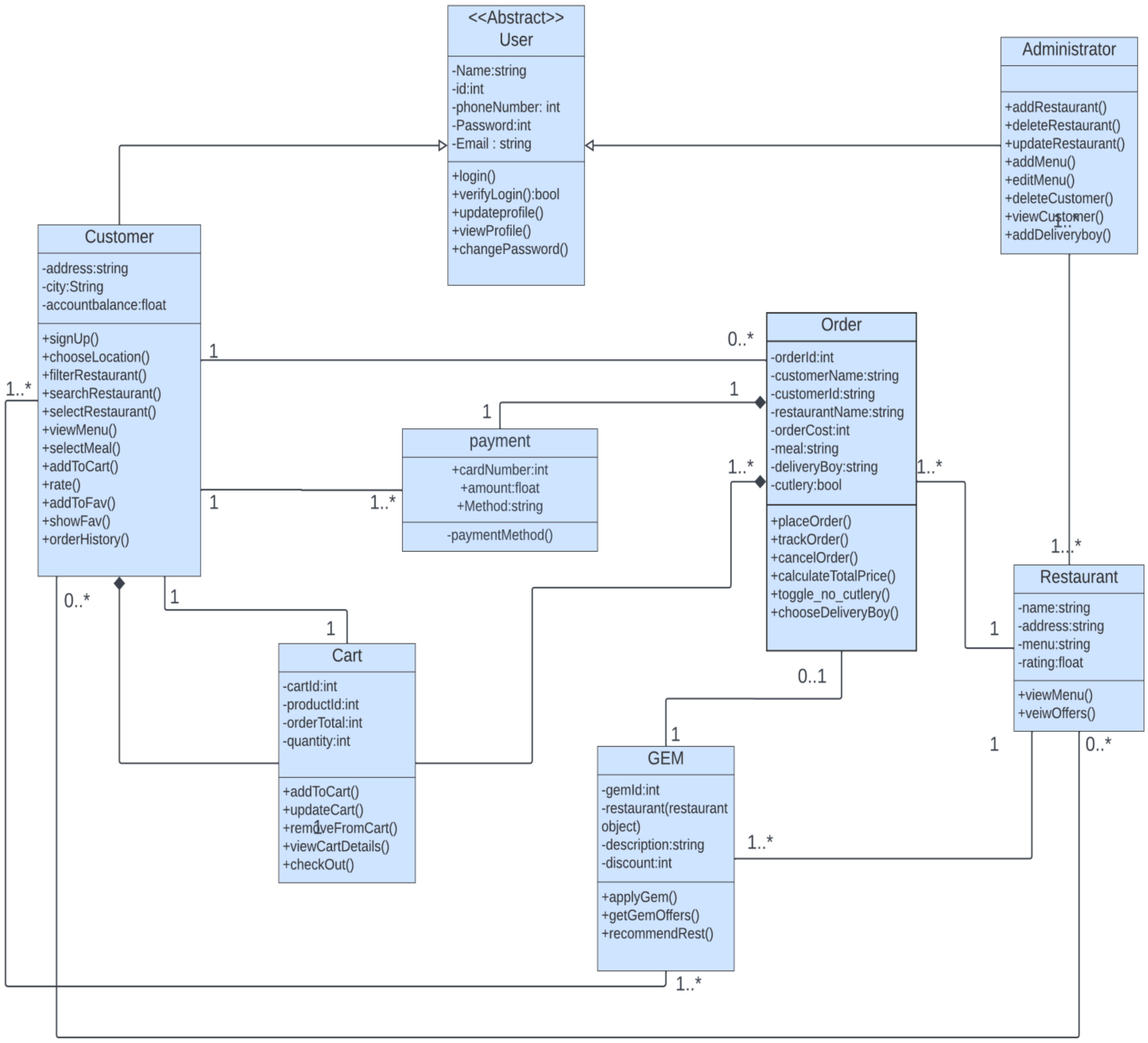


6.2 ERD Tables

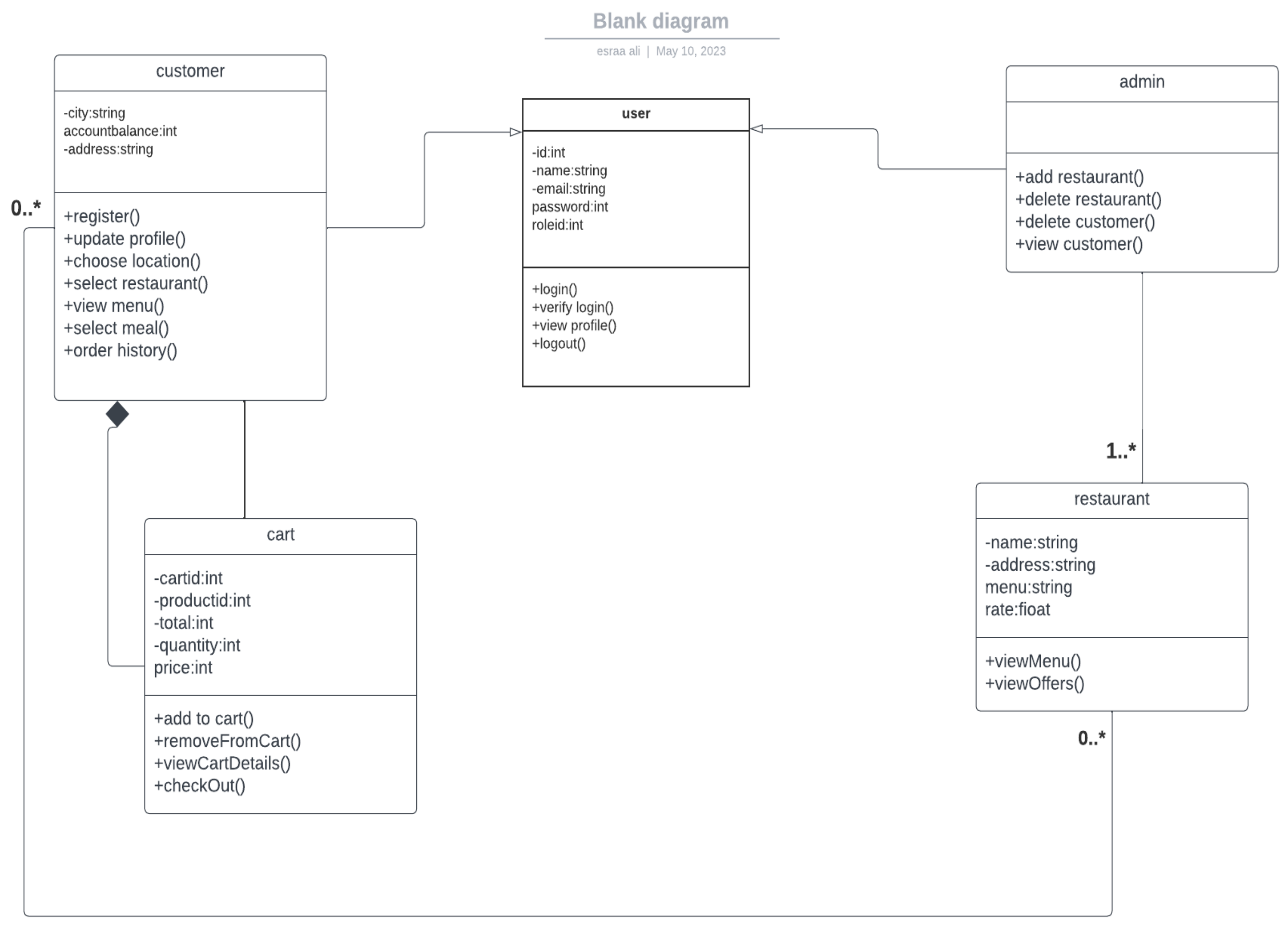


7 – Class diagram

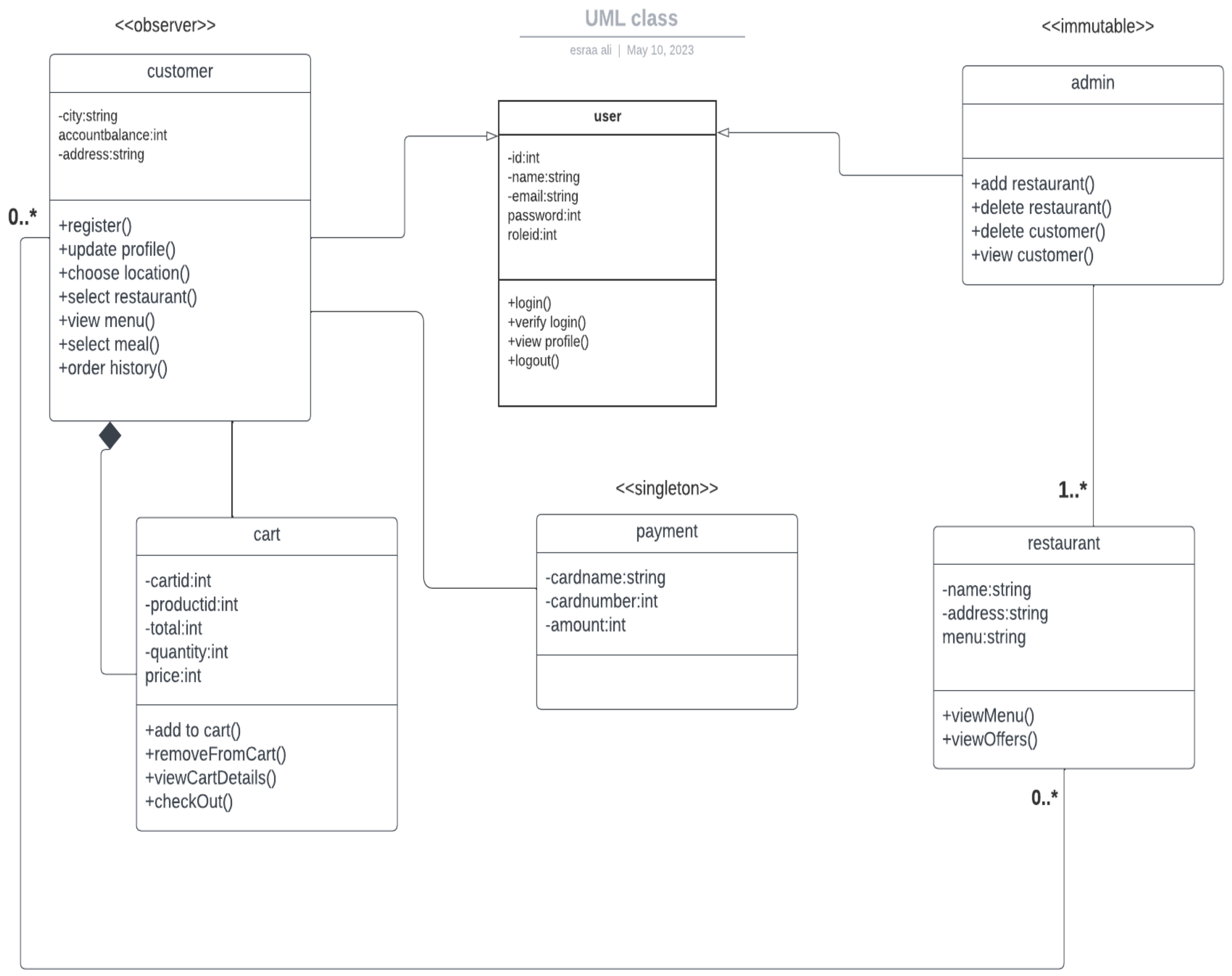
7.1 Intial Version :



7.2 intermediate version

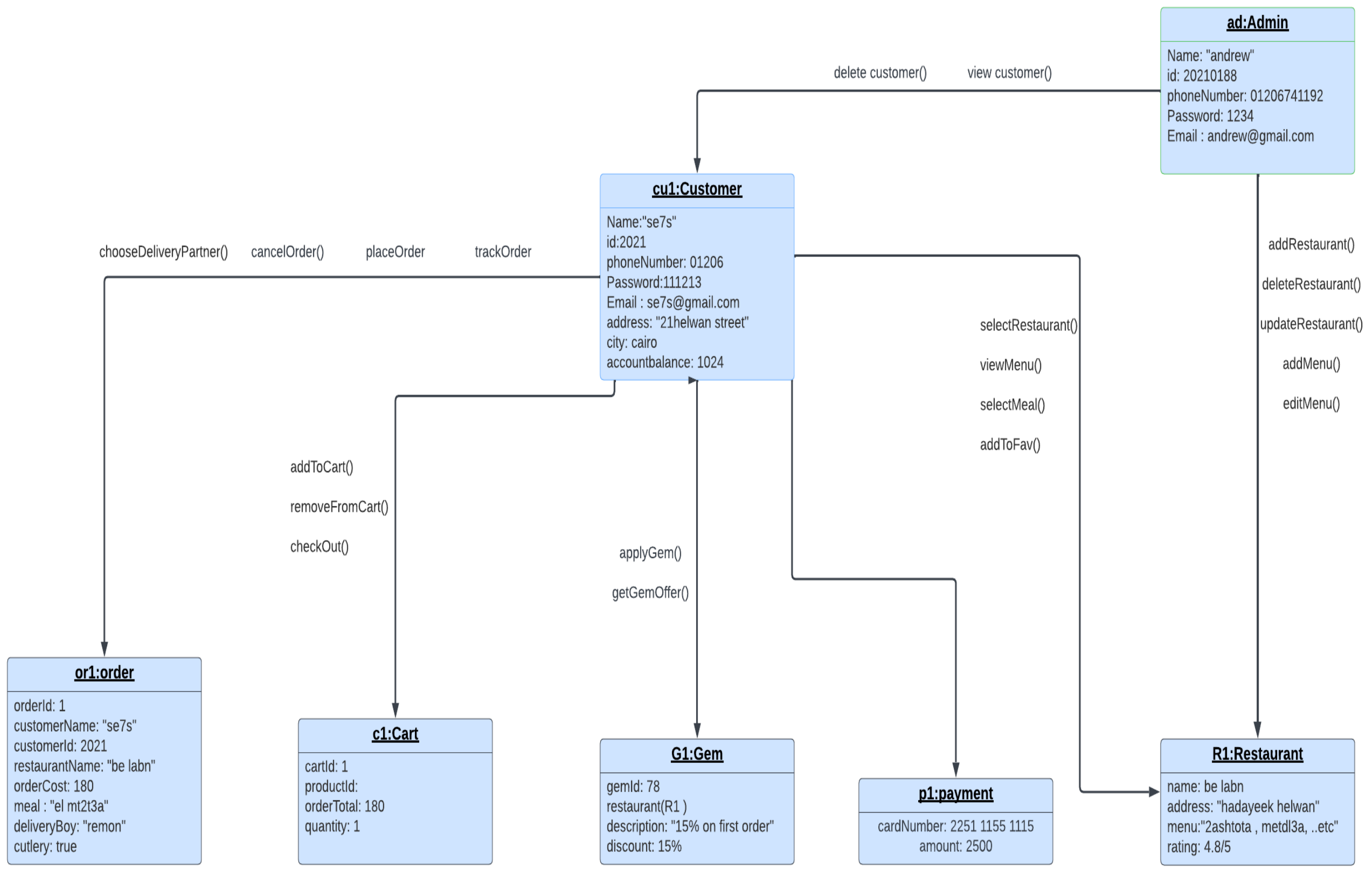


final version :

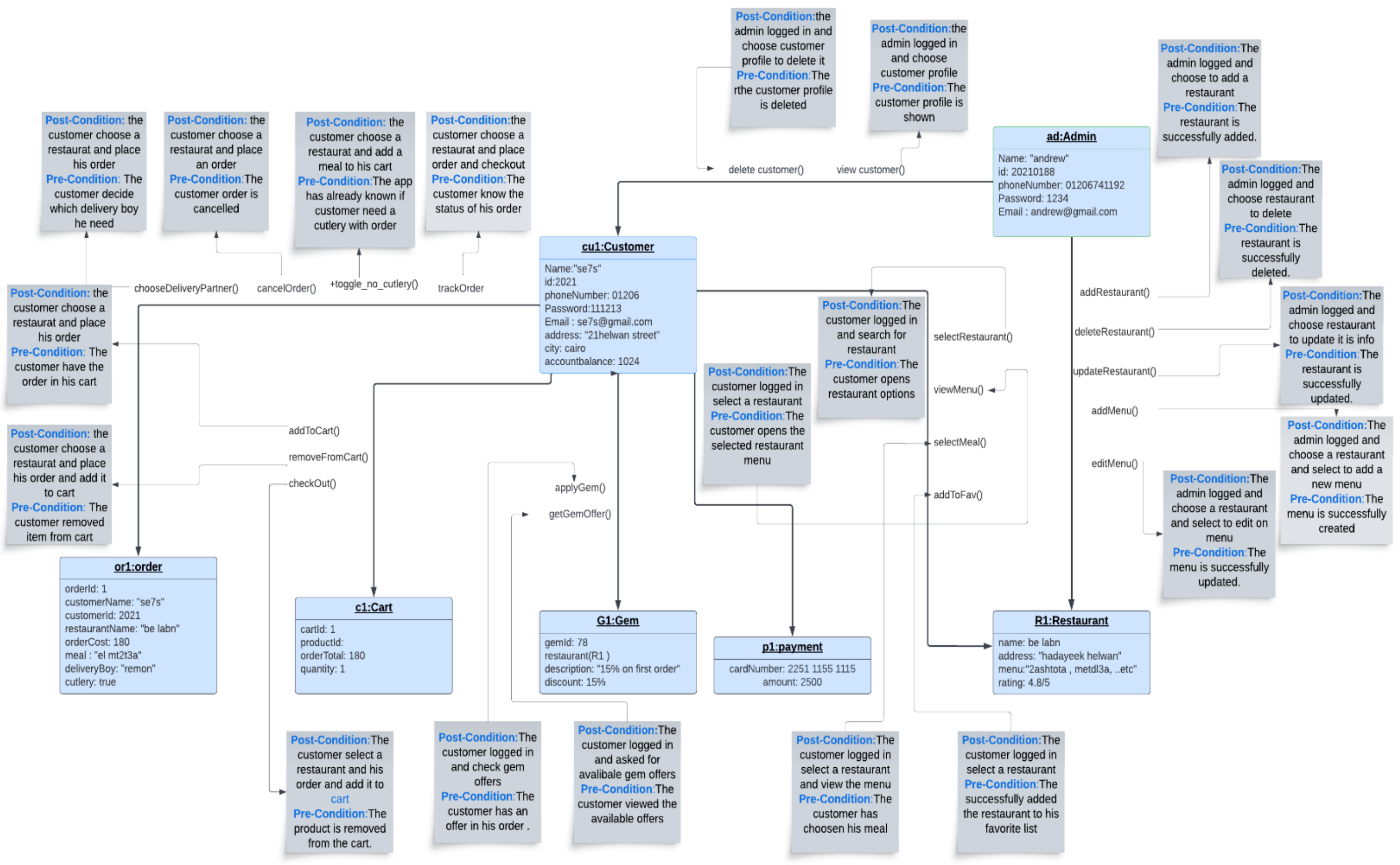


8 – Object diagram

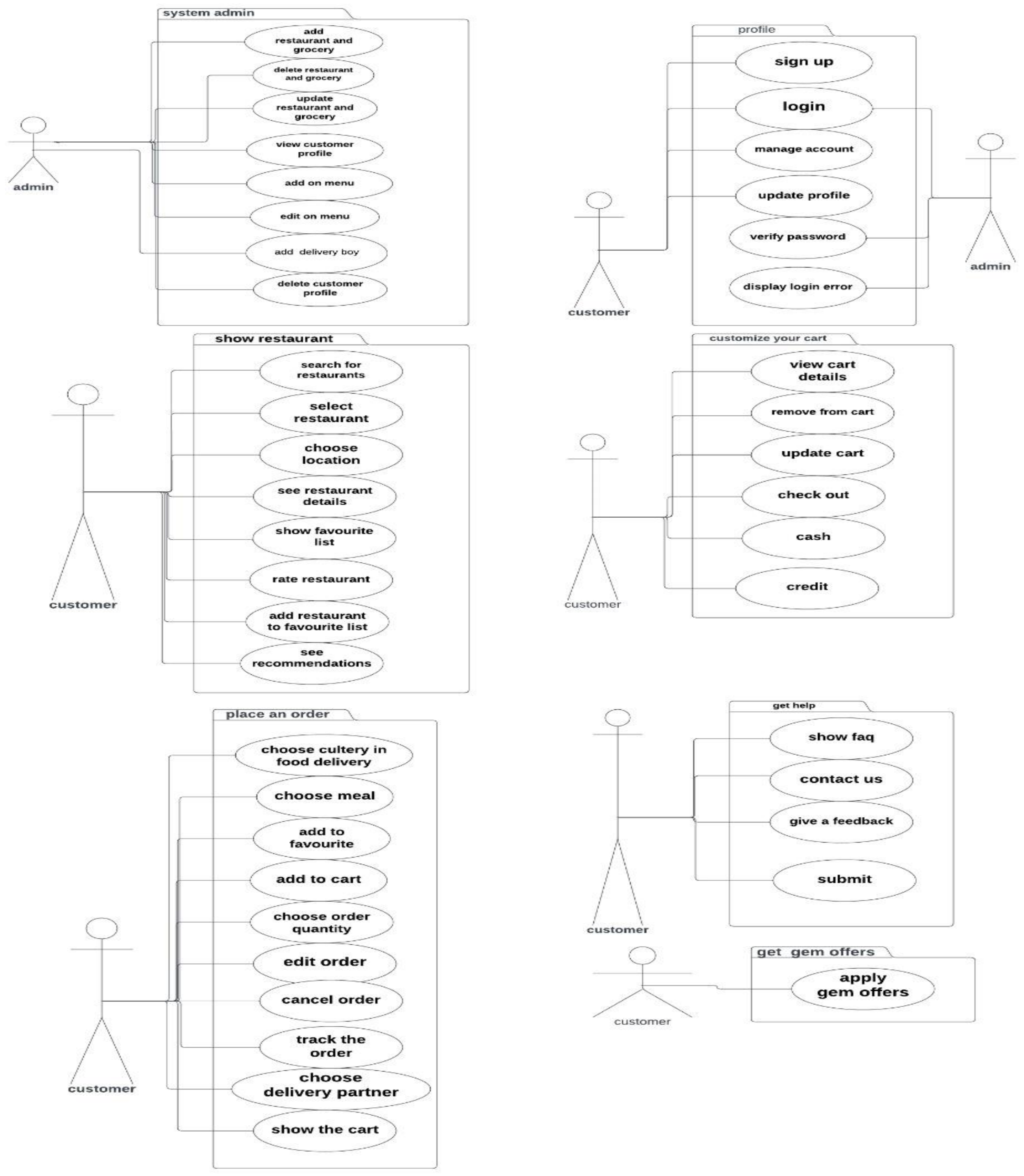
1-



2-

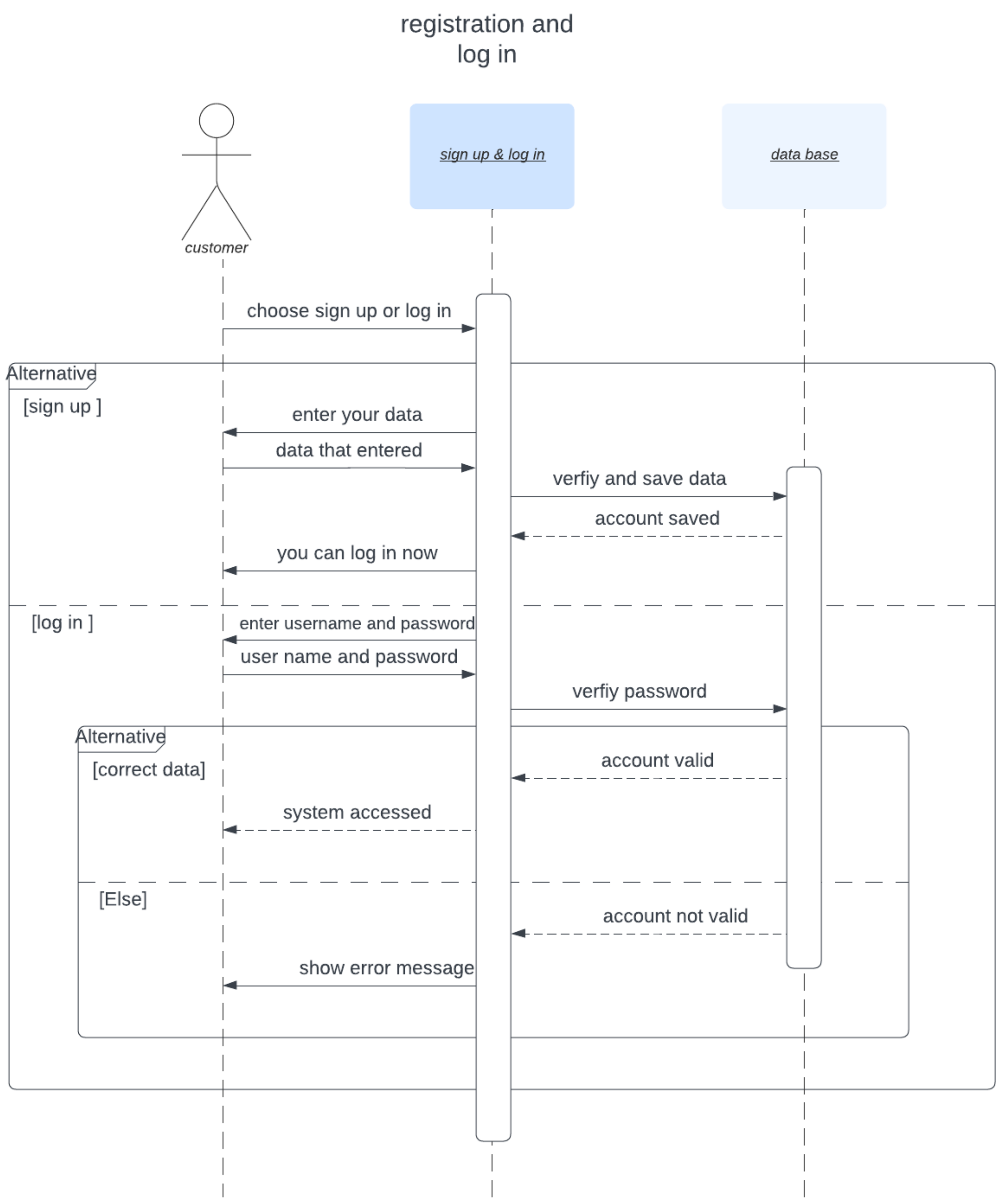

Post-Condition: the customer choose a restaurat and place his order Pre-Condition: The customer decide which delivery boy he need
Post-Condition: the customer choose a restaurat and place an order Pre-Condition:The customer order is cancelled
Post-Condition: the customer choose a restaurat and add a meal to his cart Pre-Condition:The app has already known if customer need a cutlery with order
Post-Condition:the customer choose a restaurat and place order and checkout Pre-Condition:The customer know the status of his order
Post-Condition:the admin logged in and choose customer profile to delete it Pre-Condition:The rthe customer profile is deleted
Post-Condition:the admin logged in and choose customer profile Pre-Condition:The customer profile is shown
Post-Condition:The admin logged and choose to add a restaurant Pre-Condition:The restaurant is successfully added.
delete customer()
view customer()
ad:Admin
Name: "andrew"
id: 20210188
phoneNumber: 01206741192
Password: 1234
Email : andrew@gmail.com
Post-Condition:The admin logged and choose restaurant to delete Pre-Condition:The restaurant is successfully deleted.
chooseDeliveryPartner()
cancelOrder()
+toggle_no_cutlery()
trackOrder
cu1:Customer
Name:"se7s"
id:2021
phoneNumber: 01206
Password:111213
Email : se7s@gmail.com
address: "21helwan street"
city: cairo
accountbalance: 1024
Post-Condition: the customer choose a restaurat and place his order Pre-Condition: The customer have the order in his cart
Post-Condition:The customer logged in and search for restaurant Pre-Condition:The customer opens restaurant options
Post-Condition:The customer logged in select a restaurant Pre-Condition:The customer opens the selected restaurant menu
selectRestaurant()
viewMenu()
selectMeal()
addToFav()
addRestaurant()
deleteRestaurant()
updateRestaurant()
addMenu()
editMenu()
Post-Condition:The admin logged and choose restaurant to update it is info Pre-Condition:The restaurant is successfully updated.
Post-Condition:The admin logged and choose a restaurant and select to add a new menu Pre-Condition:The menu is successfully created
Post-Condition:The admin logged and choose a restaurant and select to edit on menu Pre-Condition:The menu is successfully updated.
addToCart()
removeFromCart()
checkOut()
applyGem()
getGemOffer()
Post-Condition: the customer choose a restaurat and place his order and add it to cart Pre-Condition: The customer removed item from cart
or1:order
orderId: 1
customerName: "se7s"
customerId: 2021
restaurantName: "be labn"
orderCost: 180
meal : "el mt2t3a"
deliveryBoy: "remon"
cutlery: true
c1:Cart
cartId: 1
productId:
orderTotal: 180
quantity: 1
G1:Gem
gemId: 78
restaurant(R1)
description: "15% on first order"
discount: 15%
p1:payment
cardNumber: 2251 1155 1115
amount: 2500
R1:Restaurant
name: be labn
address: "hadayeek helwan"
menu:"2ashtota , metdl3a, ..etc"
rating: 4.8/5
Post-Condition:The customer select a restaurant and his order and add it to cart Pre-Condition:The product is removed from the cart.
Post-Condition:The customer logged in and check gem offers Pre-Condition:The customer has an offer in his order .
Post-Condition:The customer logged in and asked for avalibale gem offers Pre-Condition:The customer viewed the available offers
Post-Condition:The customer logged in select a restaurant and view the menu Pre-Condition:The customer has choosen his meal
Post-Condition:The customer logged in select a restaurant Pre-Condition:The successfully added the restaurant to his favorite list

9 – Package diagram

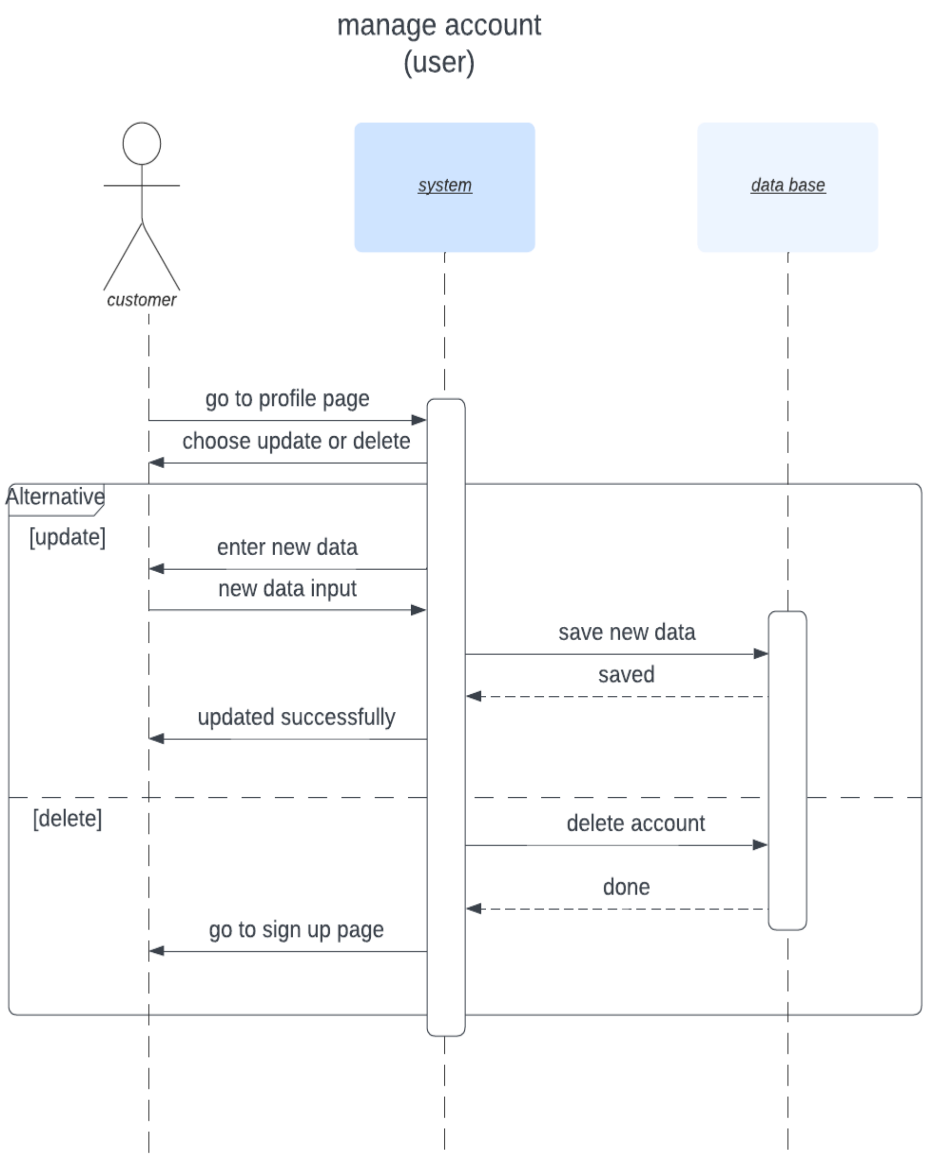


10 - Sequence Diagram:

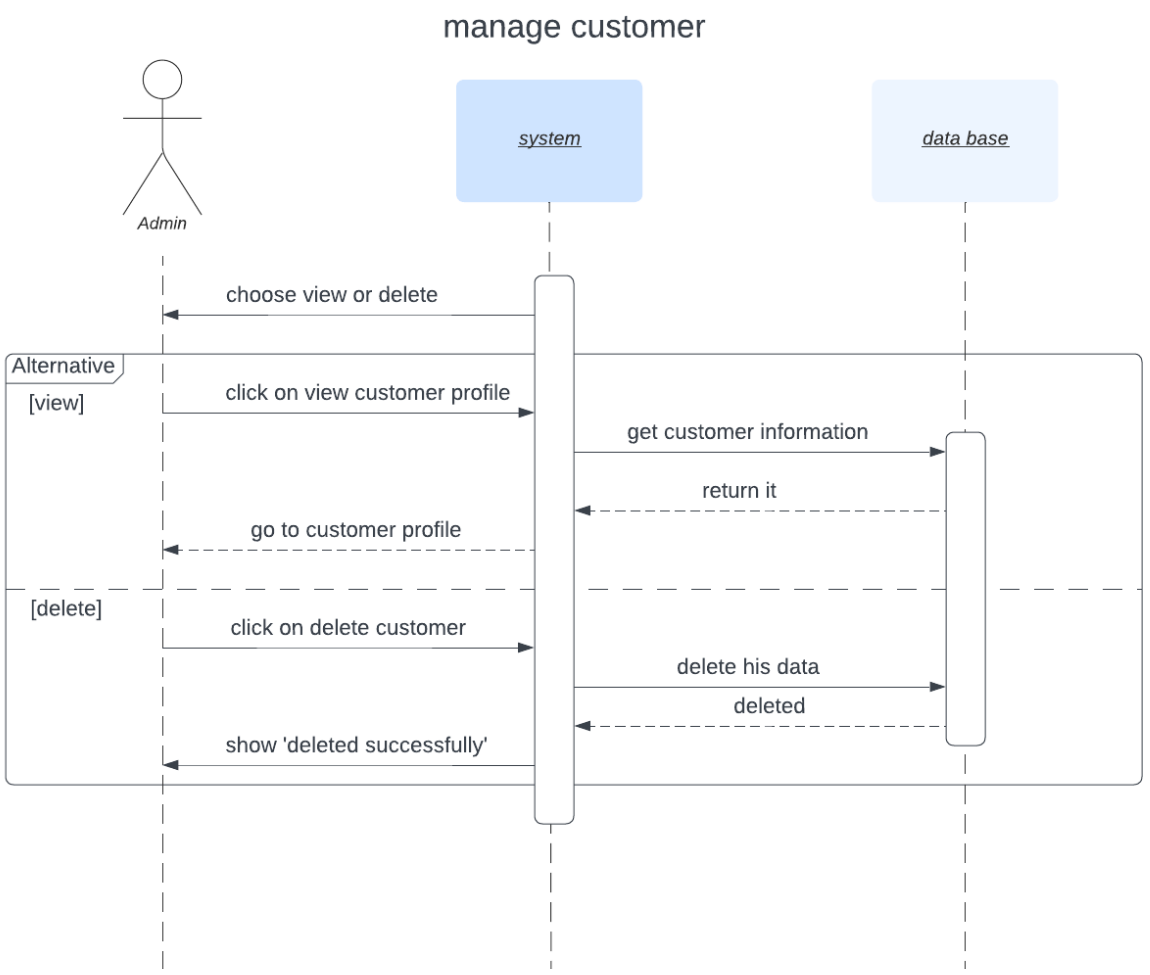
1 – Registration:



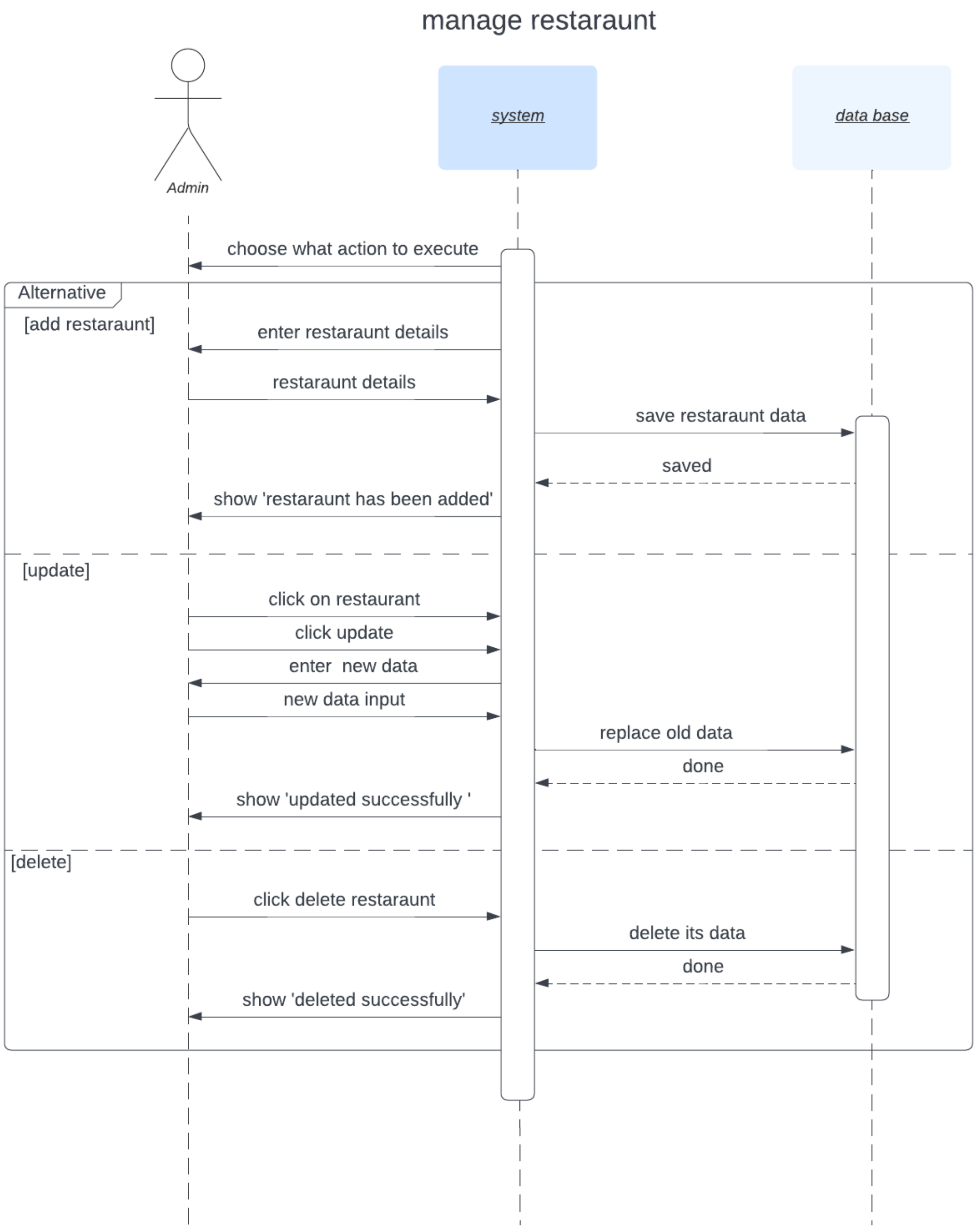
2-Manage account (user)



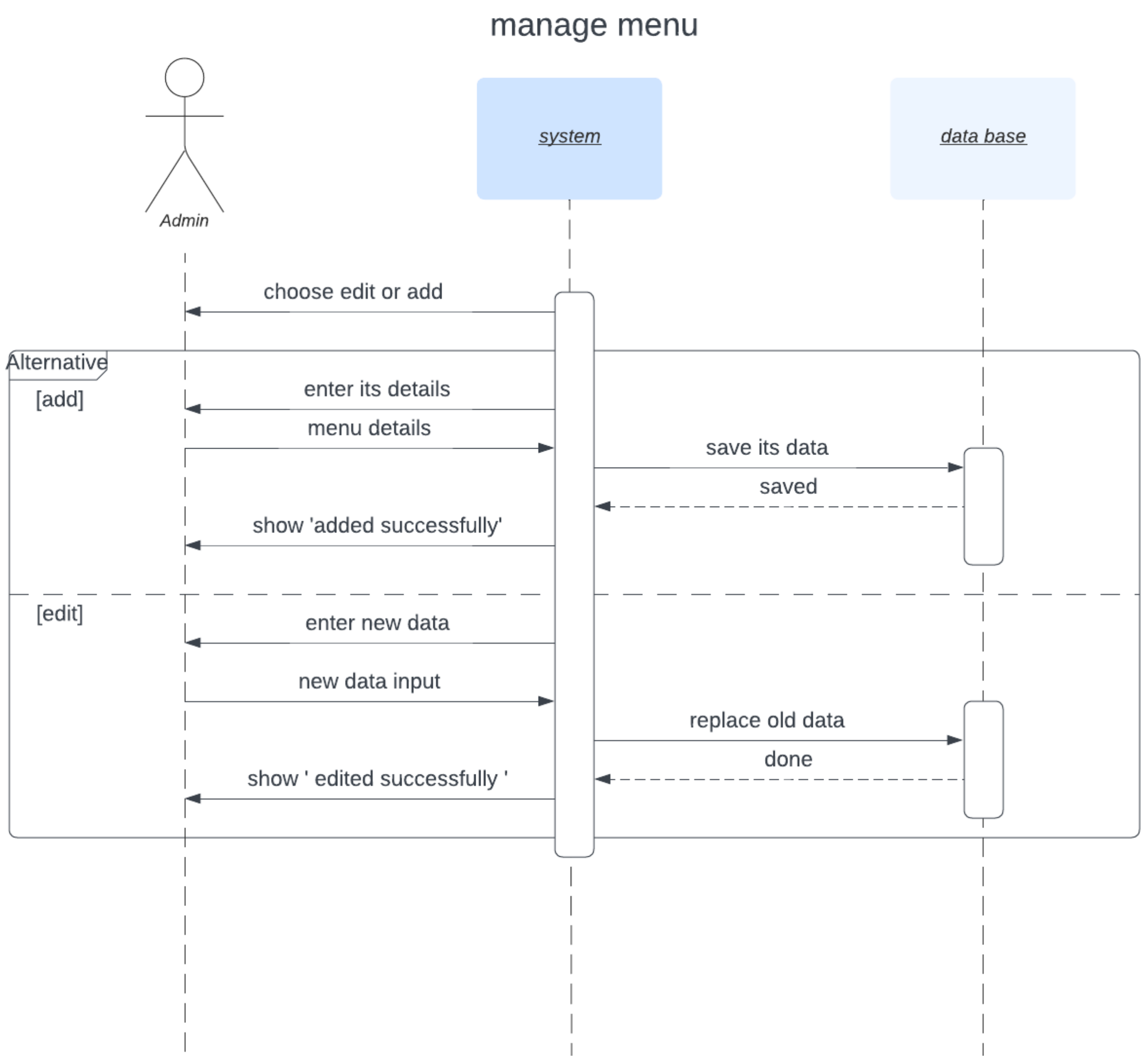
3-Manage customer (admin)



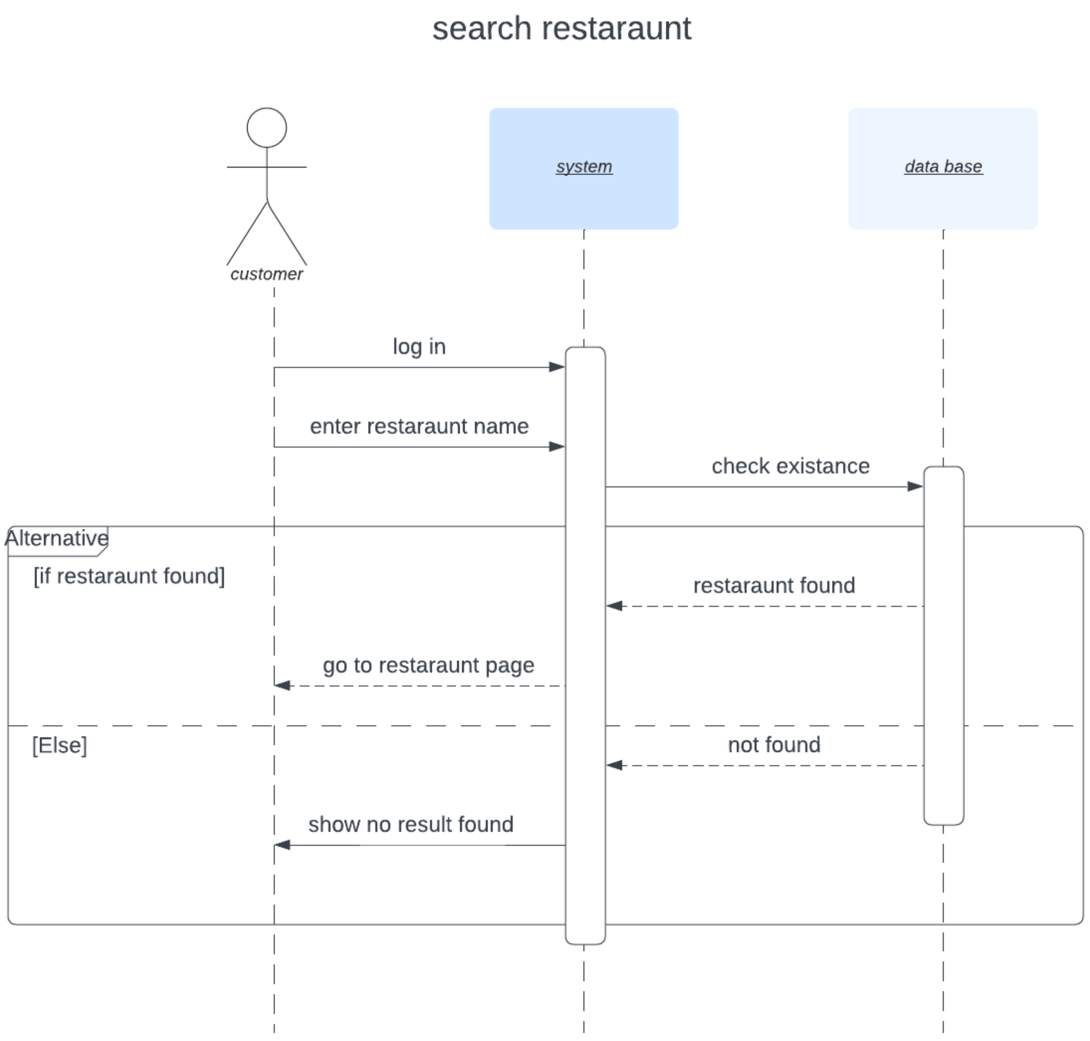
4-Manage Restaurant (admin)



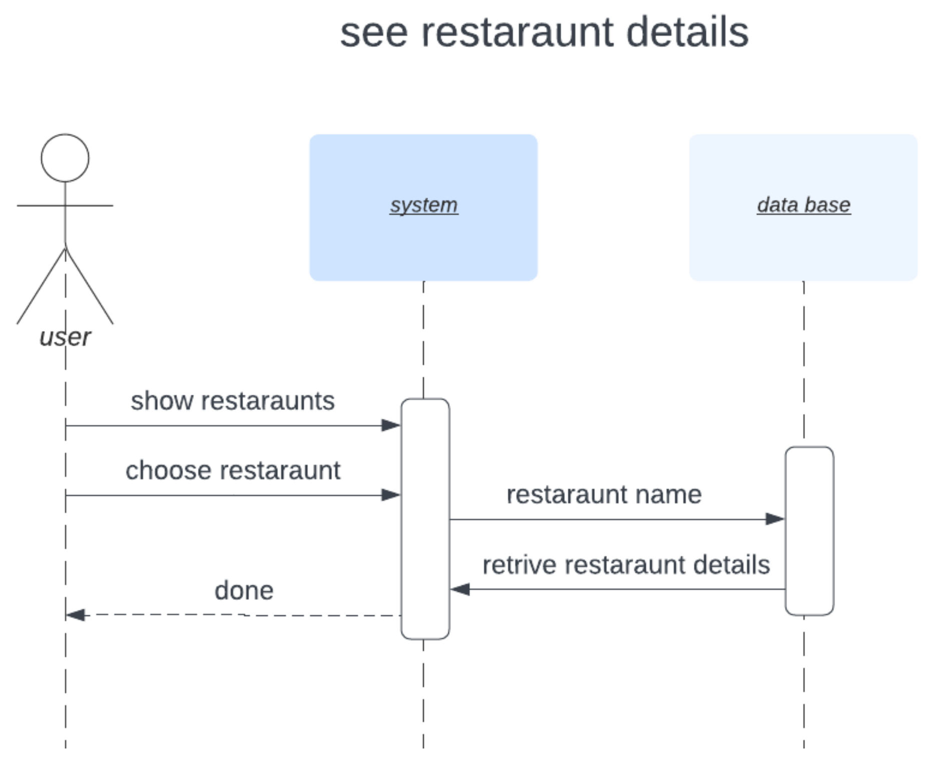
5 – Manage Menu (admin)



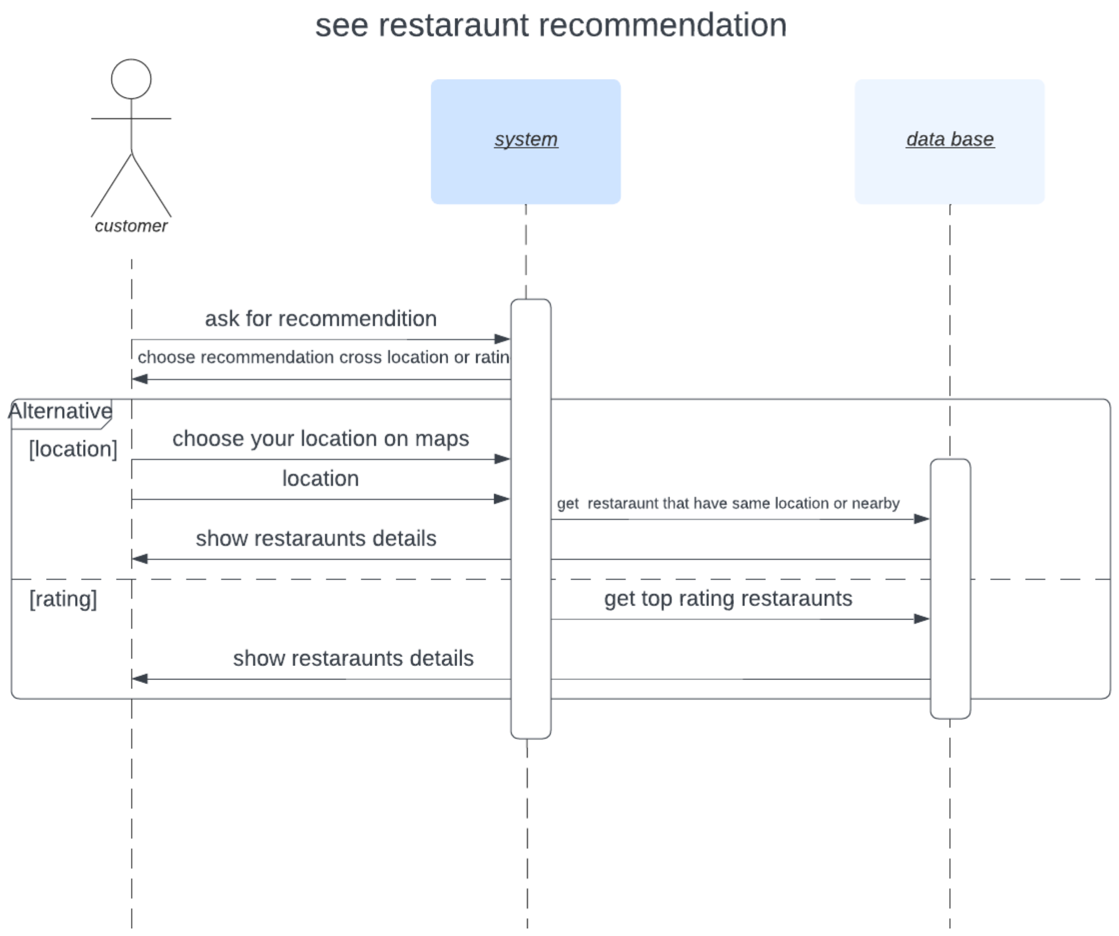
6- Search restaurant



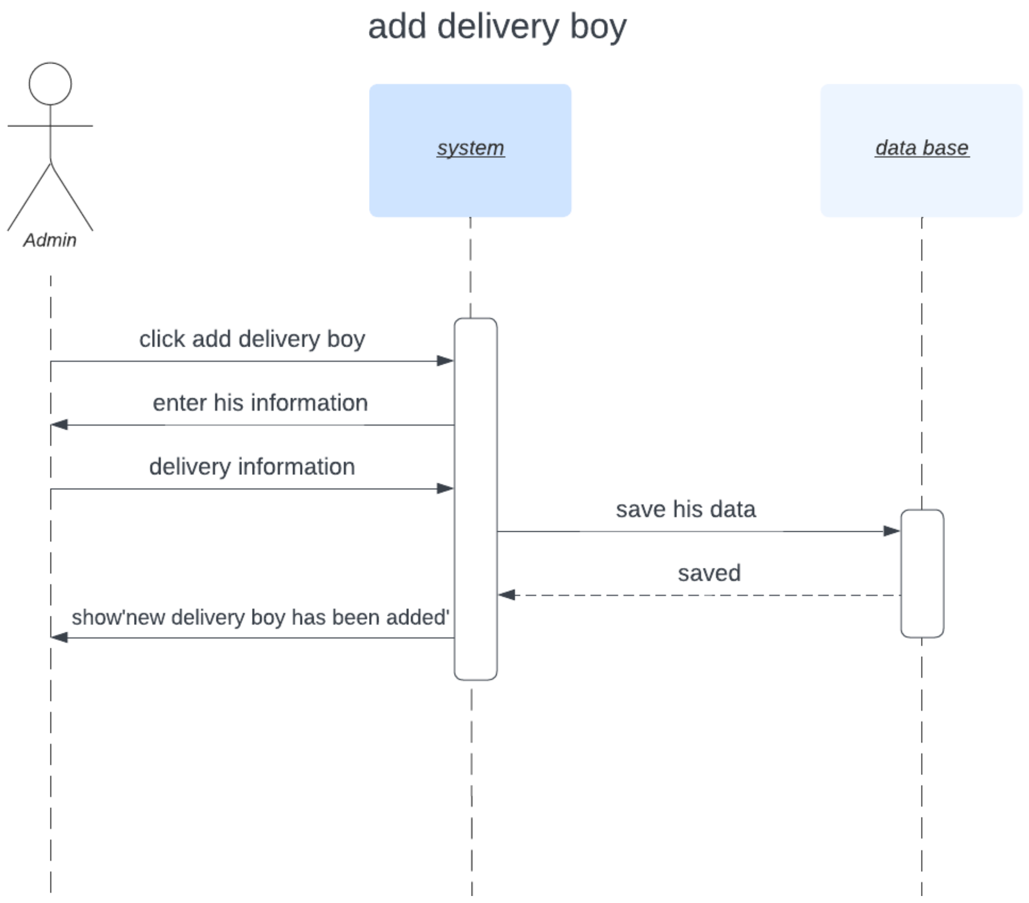
7- Restaurant Details



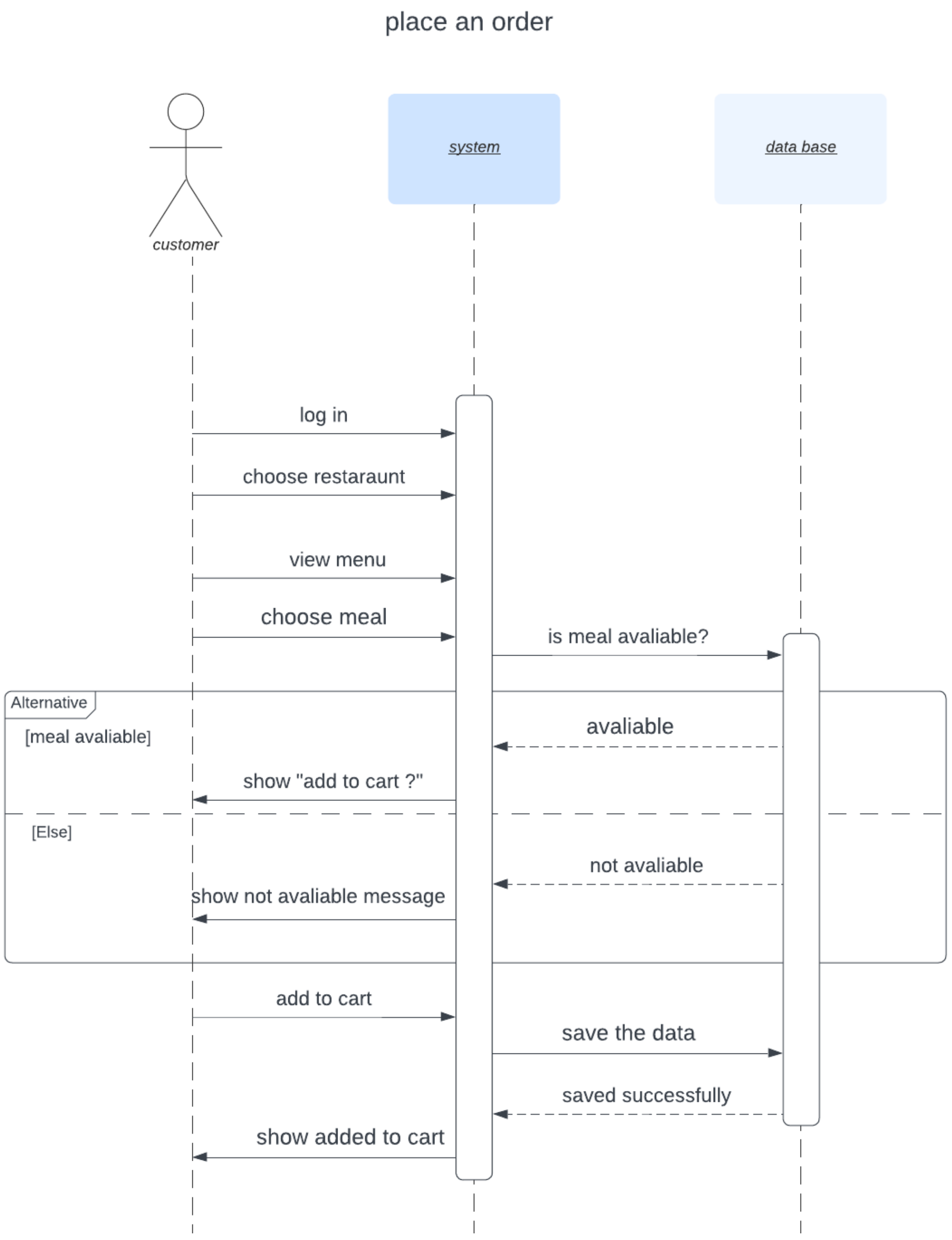
8 – see recommendations


see restaraunt recommendation
customer
system
data base
ask for recommendition
choose recommendation cross location or ratin
Alternative
[location]
choose your location on maps
location
get restaraunt that have same location or nearby
show restaraunts details
[rating]
get top rating restaraunts
show restaraunts details

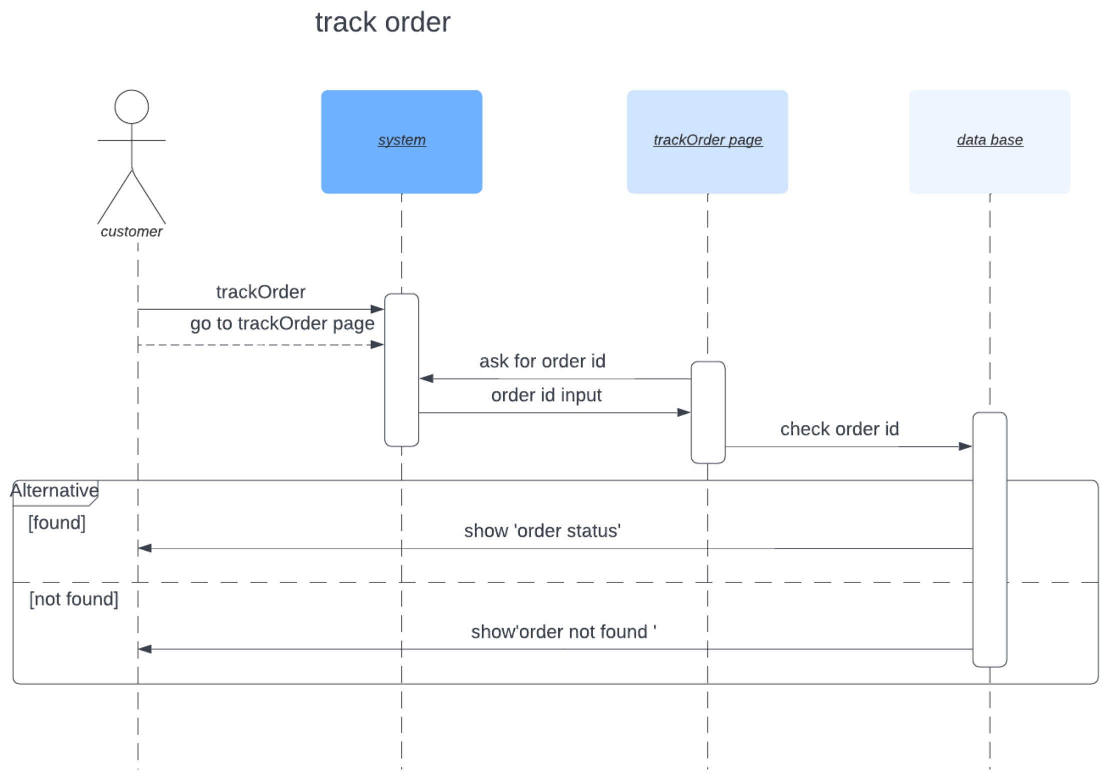
9 -Add delivery



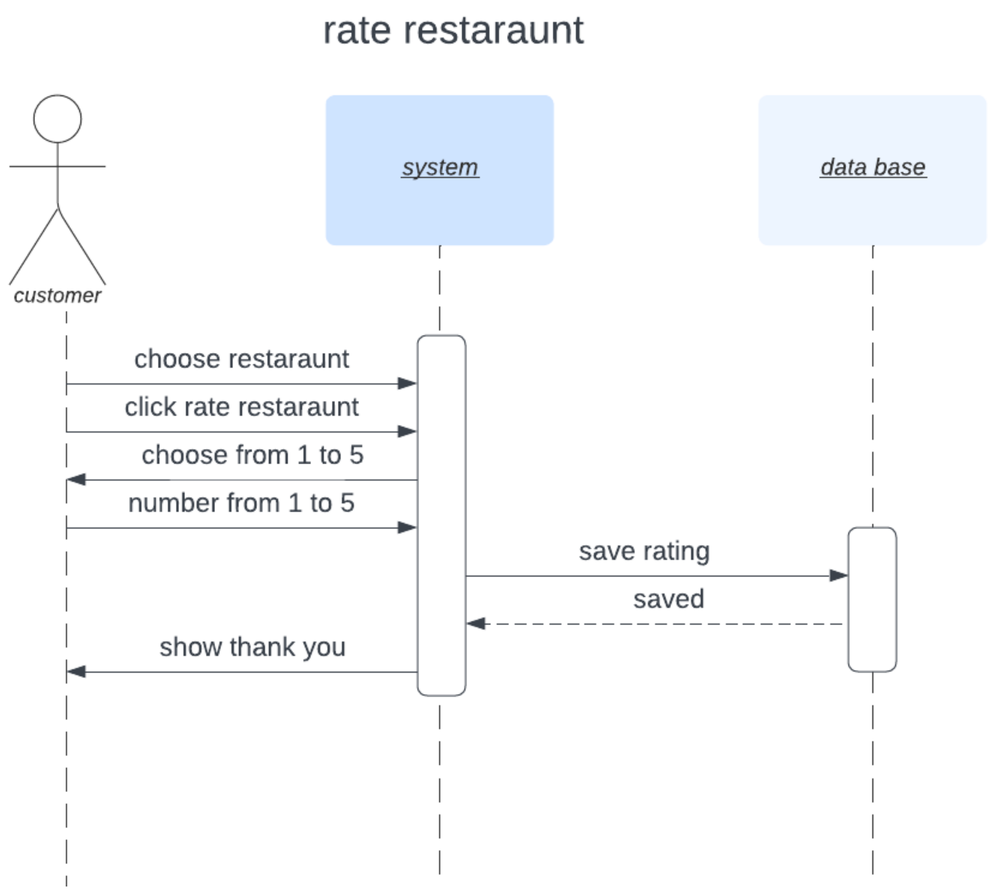
10 – Place order



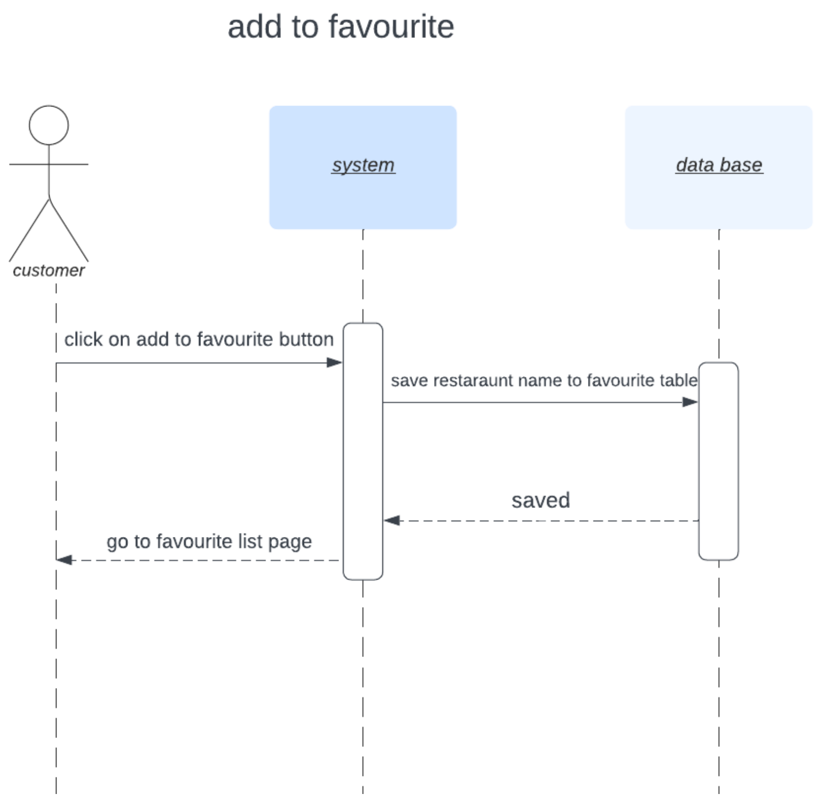
11 – Track Order



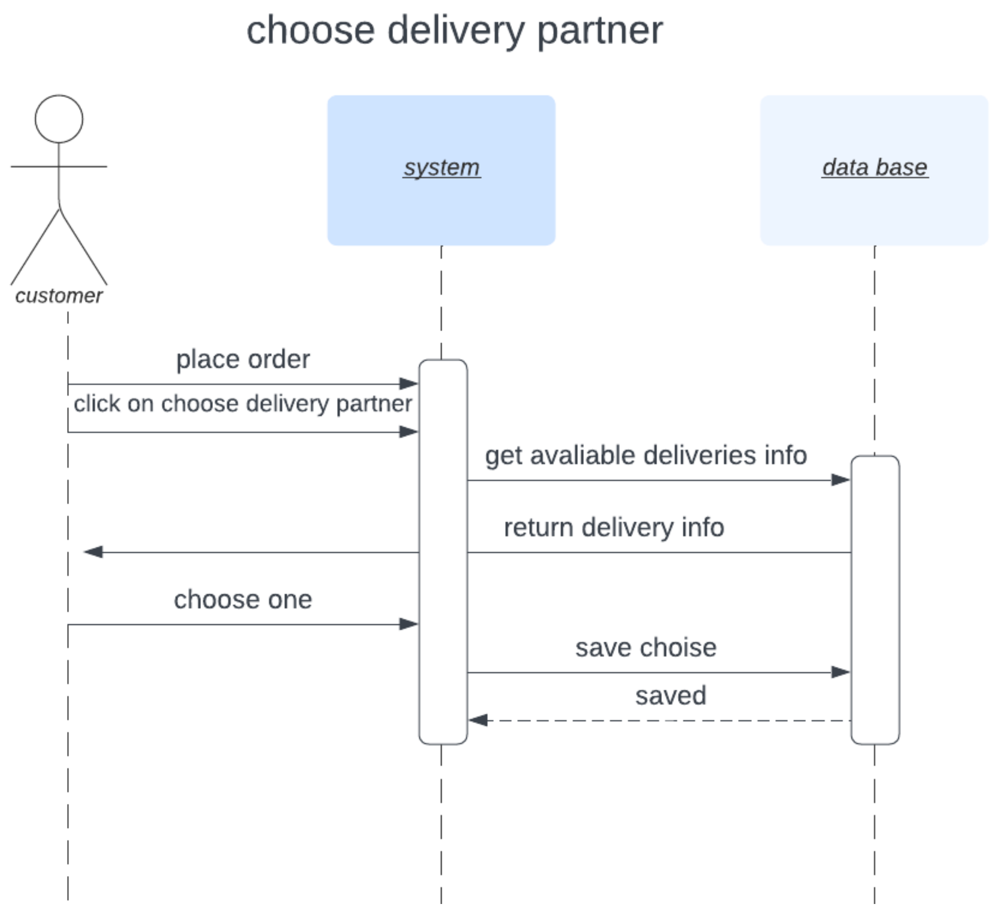
12 – Rate Restaurant



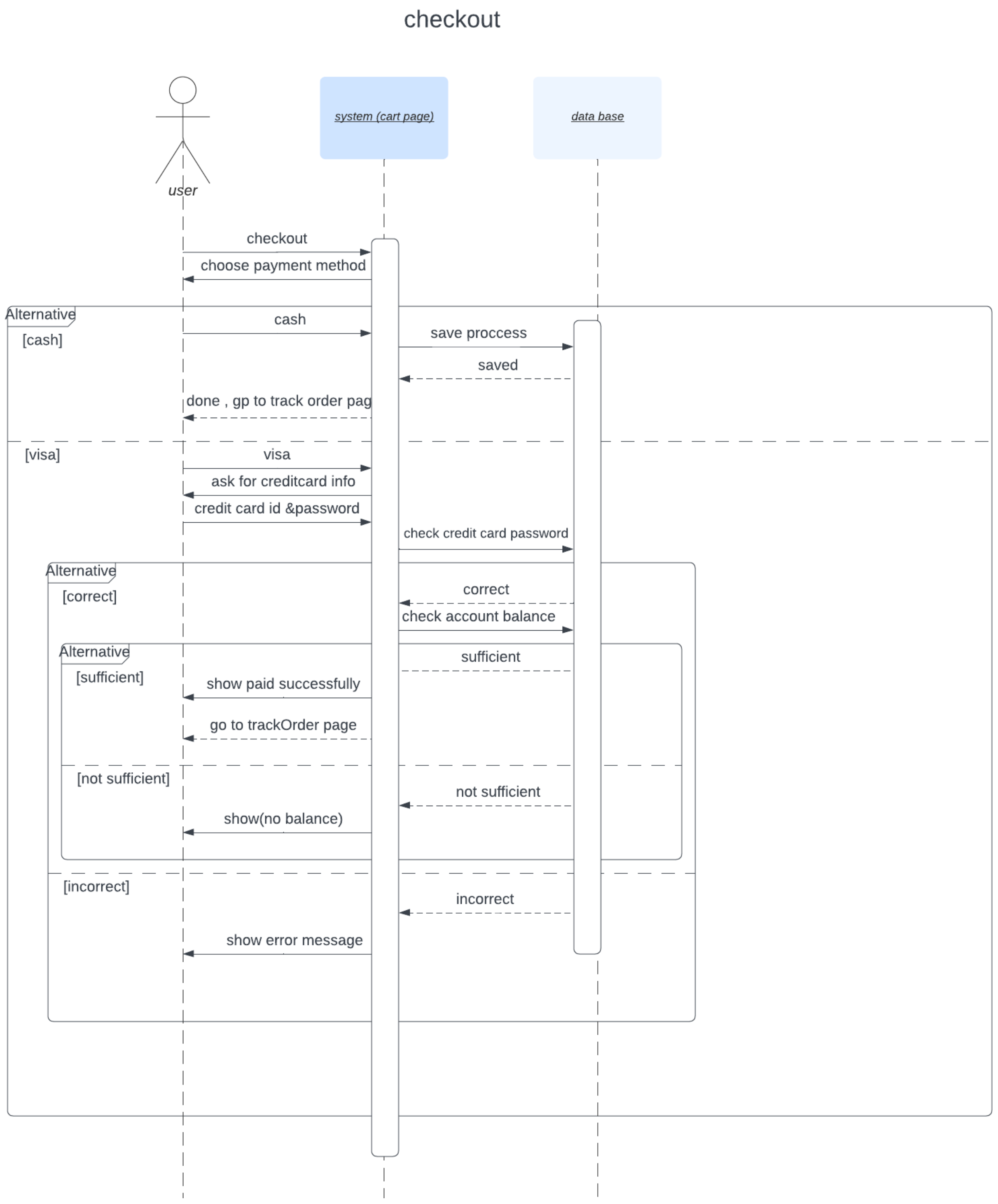
13 – Add to Favourite


add to favourite
customer
system
data base
click on add to favourite button
save restaraunt name to favourite table
saved
go to favourite list page

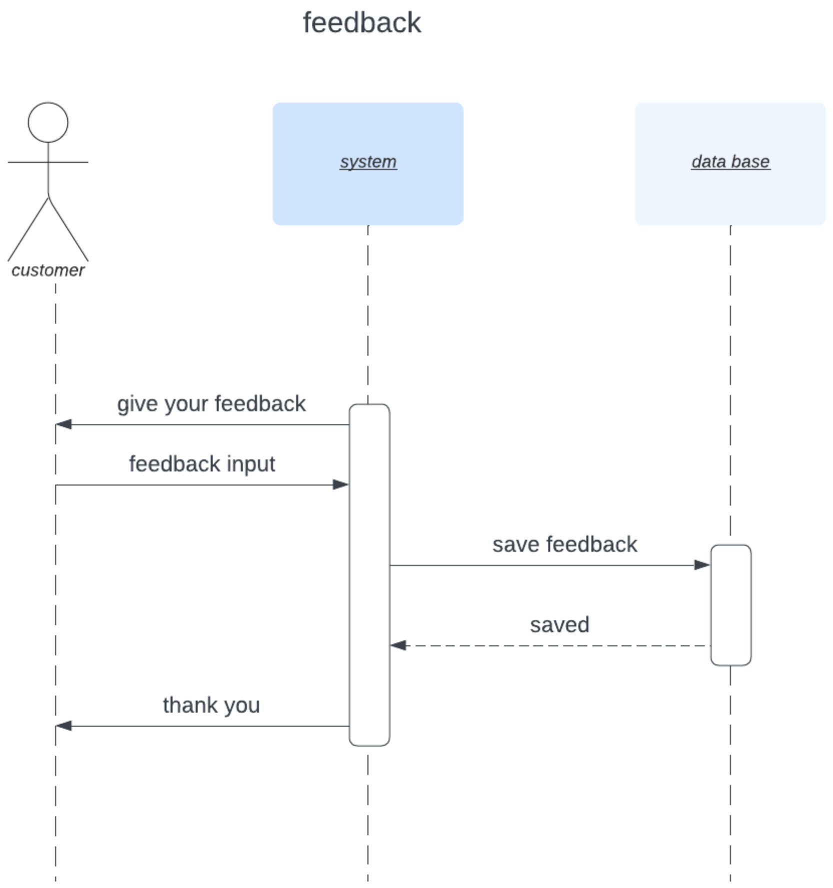
14 – Choose Delivery



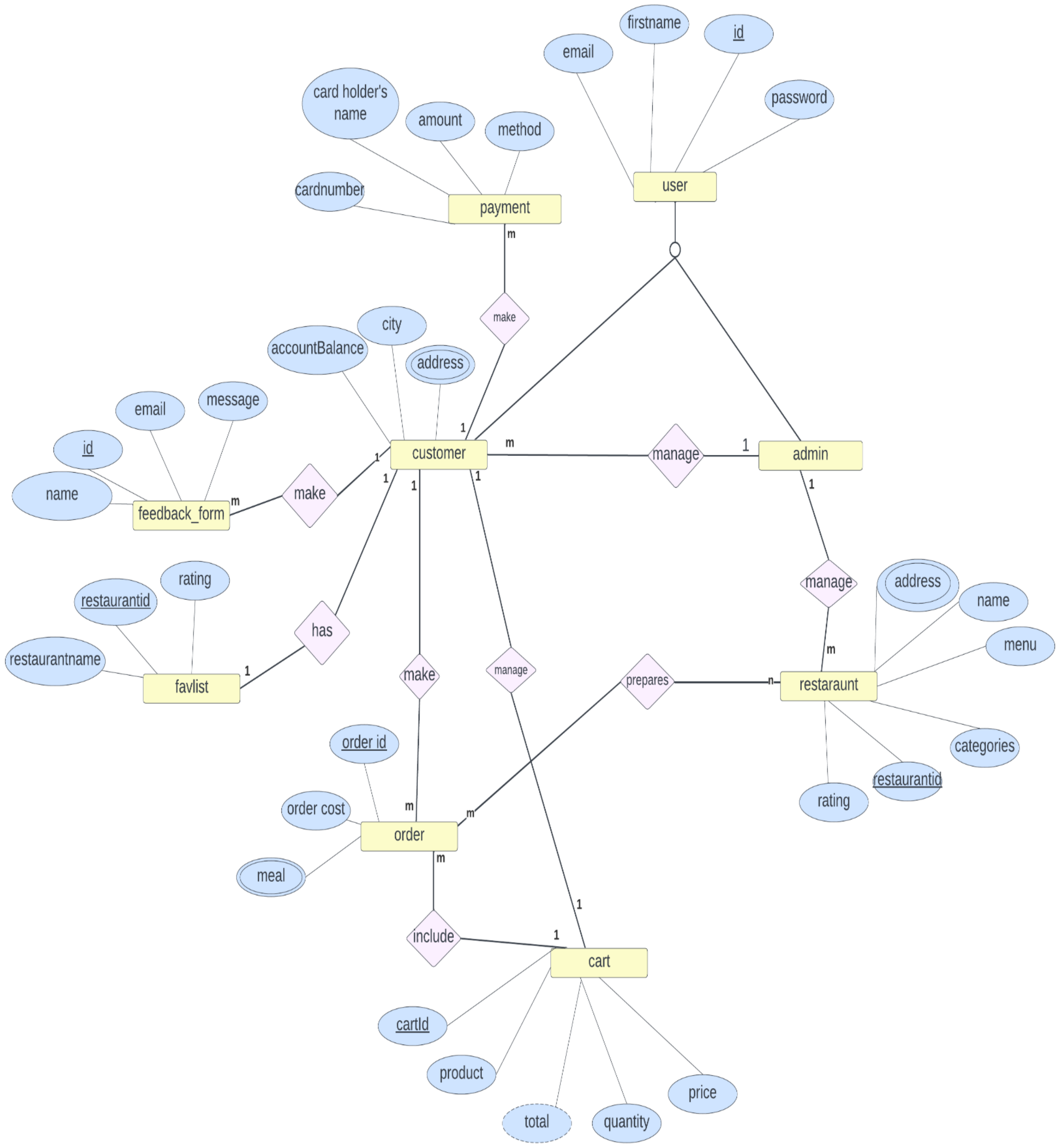
15 – Checkout


checkout
user
system (cart page)
data base
checkout
choose payment method
Alternative
[cash]
cash
save proccess
saved
done , gp to track order pag
[visa]
visa
ask for creditcard info
credit card id &password
check credit card password
Alternative
[correct]
correct
check account balance
Alternative
[sufficient]
sufficient
show paid successfully
go to trackOrder page
[not sufficient]
not sufficient
show(no balance)
[incorrect]
incorrect
show error message

16 – Feedback

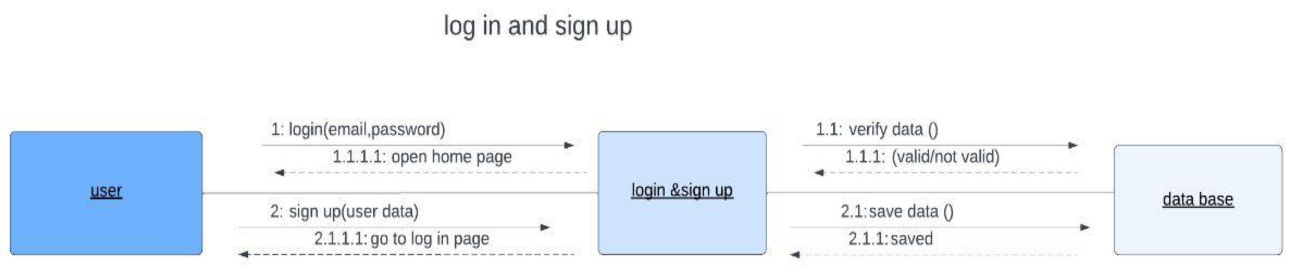

feedback
customer
system
data base
give your feedback
feedback input
save feedback
saved
thank you

11 – Updated ERD :

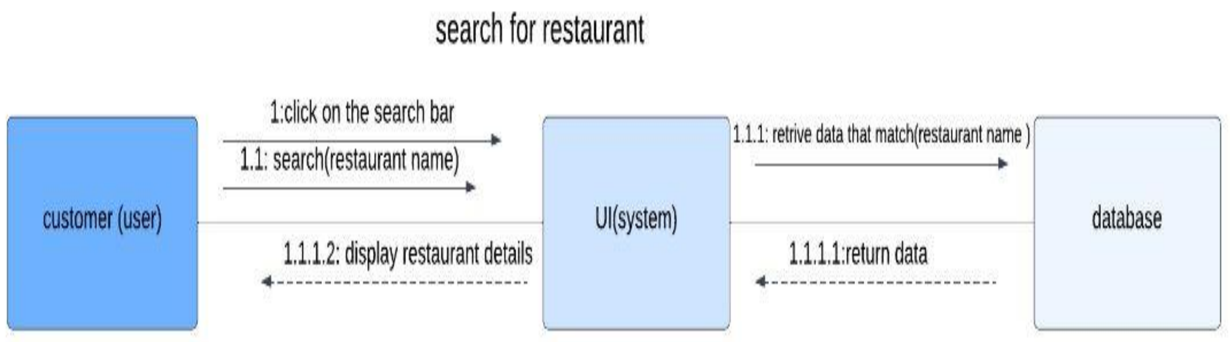


12 – Collaboration:

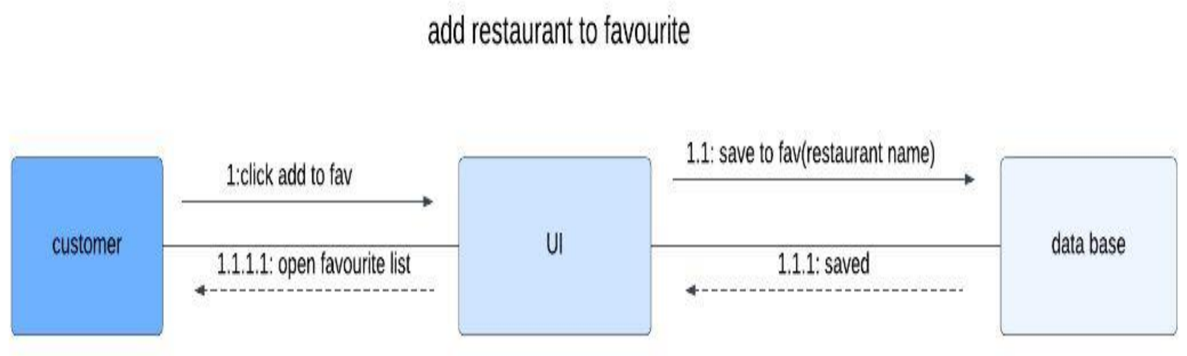
1.Log in and sign up :



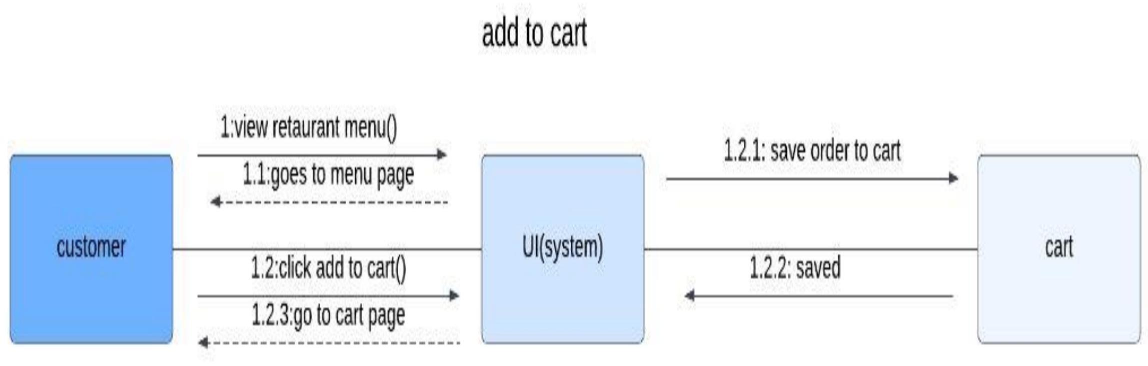
2.search for restaurant:


search for restaurant
customer (user)
1:click on the search bar
1.1: search(restaurant name)
1.1.1.2: display restaurant details
UI(system)
1.1.1: retrive data that match(restaurant name)
1.1.1.1:return data
database

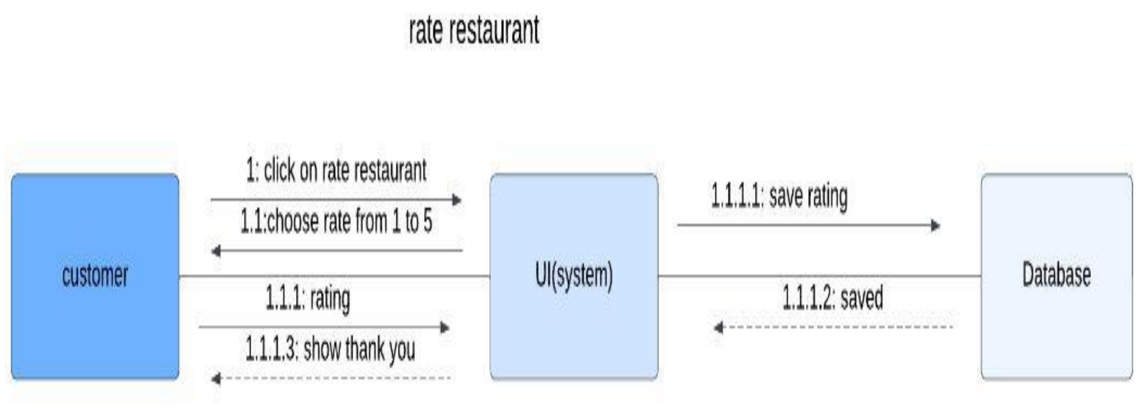
3.add restaurant to favorite


add restaurant to favourite
1:click add to fav
1.1: save to fav(restaurant name)
customer
UI
data base
1.1.1.1: open favourite list
1.1.1: saved

4.add to cart



5.rate restaurant



13 – Design Patterns :

1-Singleton:

1-Context

It is very common to find classes for which only one instance should exist (*singleton*).

Examples: Payment class.

2-Problem

We should ensure that it is never possible to create more than one instance of Admin class And provide a global point of access to it.

3-Forces

1-The use of a public constructor cannot guarantee that no more than one instance will be created.

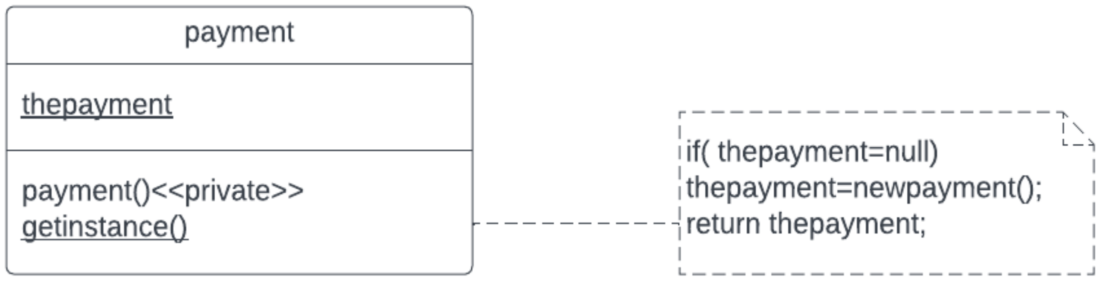
2-The singleton instance must also be accessible to all classes that require it, therefore it must often be public

4-Solution

1-Have the constructor private "payment()" to ensure that no other class will be able to create an instance of the class singleton.

2-Define a public static method "pay()", The first time this method is called ,it creates the single instance of the class "singleton" and stores a reference to that object in a static private variable.

Example:



2-Observer :

Context:

▪**When partitioning a system into individual classes you want the coupling between then to be loose so you have the flexibility to vary them independently.**

Problem:

▪A mechanism is needed to ensure that when the state of an object changes related objects are updated to keep them in step.

Forces:

▪The different parts of a system must keep in step with one another without being too tightly coupled.

Solution:

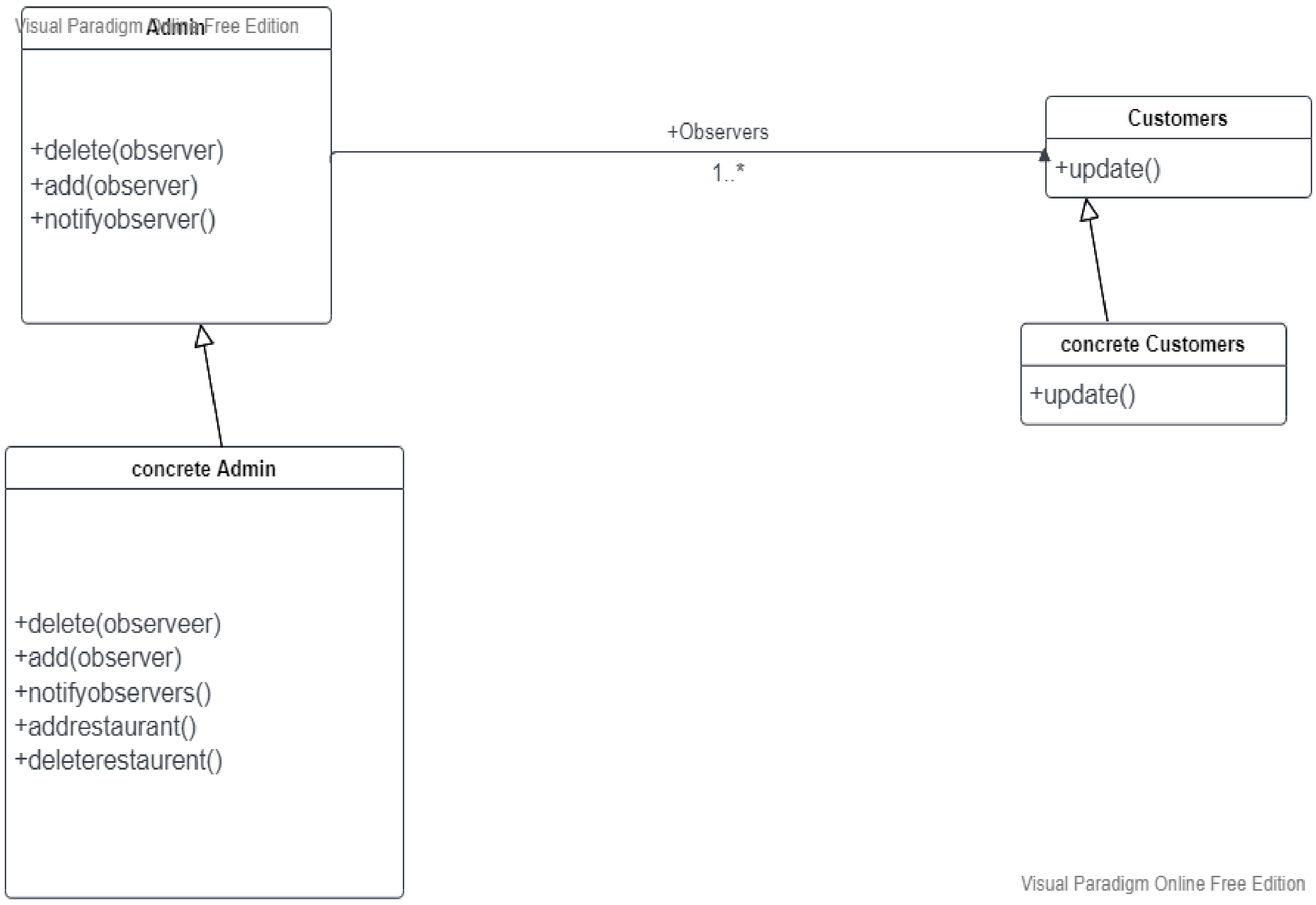
▪ Admin has the role of the subject/publisher and one or more other objects the role of observers/subscribers. The observers register themselves with the subject, & if the state of the subject changes the observers are notified & can the update themselves.

▪ The are two variants:

▪the Push Model where the subject sends the observers detailed information about the change that has occurred, and

▪the Pull Model where the subject simple notifies the observers that there have been changes, and it's the responsibility of the observers to find out the details they need to update themselves.

Example:



3-immutable :

Context:

An immutable object is an object that has a state that never changes after creation.

Problem:

How do you create a class whose instances are immutable?

Forces:

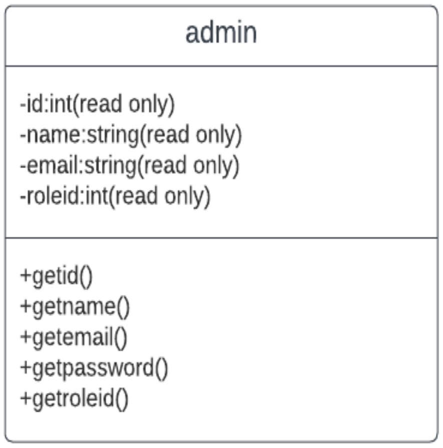
There must be no loopholes that would allow 'illegal' modification of an immutable object.

Solution:

Ensure that the constructor of the immutable class is the only place where the values of instance variables are set or modified.

Instance methods which access properties must not change instance variables.

Example:


admin
-id:int(read only)
-name:string(read only)
-email:string(read only)
-roleid:int(read only)
+getid()
+getname()
+getemail()
+getpassword()
+getroleid()