Software Engineering-project

2022-2023

Carrefour Market Management

By:

1.Mohamed Sabry Mahmoud Shehab Eldean

2.Mohamed Helmy Abd alaziz Ahmed

3.Leqaa Mohamed Abdallah Abdelsalam

4.Fatma Mohamed Elsayed Abdelmaboud

5.Mohamed Salah Abdel Moneim Abdel Fattah

6.Mohamed Sabry Mohamed Ahmed

Supervised by : Eng. Abdallah Mohamed

Instructor : Prof.Shimaa Talat

TABLE OF CONTENTS

1-Introduction

Carrefour market management is popular and large supermarket, it have a thousand of products So we make a system to control and coordinate transaction in this supermarket, we make a system to Make a customer happy and deal with supermarket easily, save time, effort that customer do to buy Something from a supermarket, customer communicate with supermarket efficiently way, essential Point in this system solve a problem of People crowded in supermarket (corona virus widespread in The world), there exist admin in the system have access on any process in the system and can add Customer, stuff to the system and have report of everything about the supermarket, there exist a Person who deliver order to the customer, there exist financial management that responsible for Anything that related to money, there exist stuff may belongs to sales or marketing, stuff sales that Responsible for communicate with customer in the supermarket and every section in supermarket Have stuff sales, stuff marketing that responsible for market supermarket on the internet, update Products on the internet and reply to the customer where that ask question to the system, there Exist two type of customers (normal customer, special customer) a special customer is a person who Buy from the supermarket more than 10000$ and has offers better than offers of normal customers.

2-Problem definition

A **problem** is undesirable situations that prevent an organization from achieving its goals and

May be current or anticipated and there exist problem in the system and we can solve this problem

Such as :

- Transactions in the super market are slow and solve this problem throw Make software that Make transaction in the supermarket easy, efficient, organized and controlled.
- solve crowded in the supermarket throw the customer can serve order online and determine time To take this order (because there exist corona virus spread wide in the world).
- Save time and effort that customer do to request and takes order.
- people doesn't able to go to supermarket we solve this problem and make the system support Deliver order to the house.
- If it happen crowd we solve it throw using management software a barcode scanner to make A process of payment quickly.
- Manage money in the supermarket (import or export) throw Financial Management.
- Hacking the system and solve this problem throw (high security that save this system)
- The main aim of our project to provide a paper-less supermarket up to 70% to 90% and help the owner to know any transaction done in the supermarket and it can also help in decision making by providing all Necessary information about the supermarket

3-System Objectives

It divided into two parts main and subsidiary.

***Start with the main** and it concerned with work section:

1- provide an ease of contact between the work sections and ease of access to the Information that would make the work more efficient.

2- make the processes between the work section faster, more efficient and nearly free Of problems.

3- Help the supervisor to control the section and help to prevent the problems as Possible.

***Then we turn to the subsidiary and it concerned with the customer:**

1- Provide a paper-less supermarket up to 70% to 90% and help the owner to know any transaction done in the supermarket and it can also help in decision making by providing all necessary information about the supermarket.

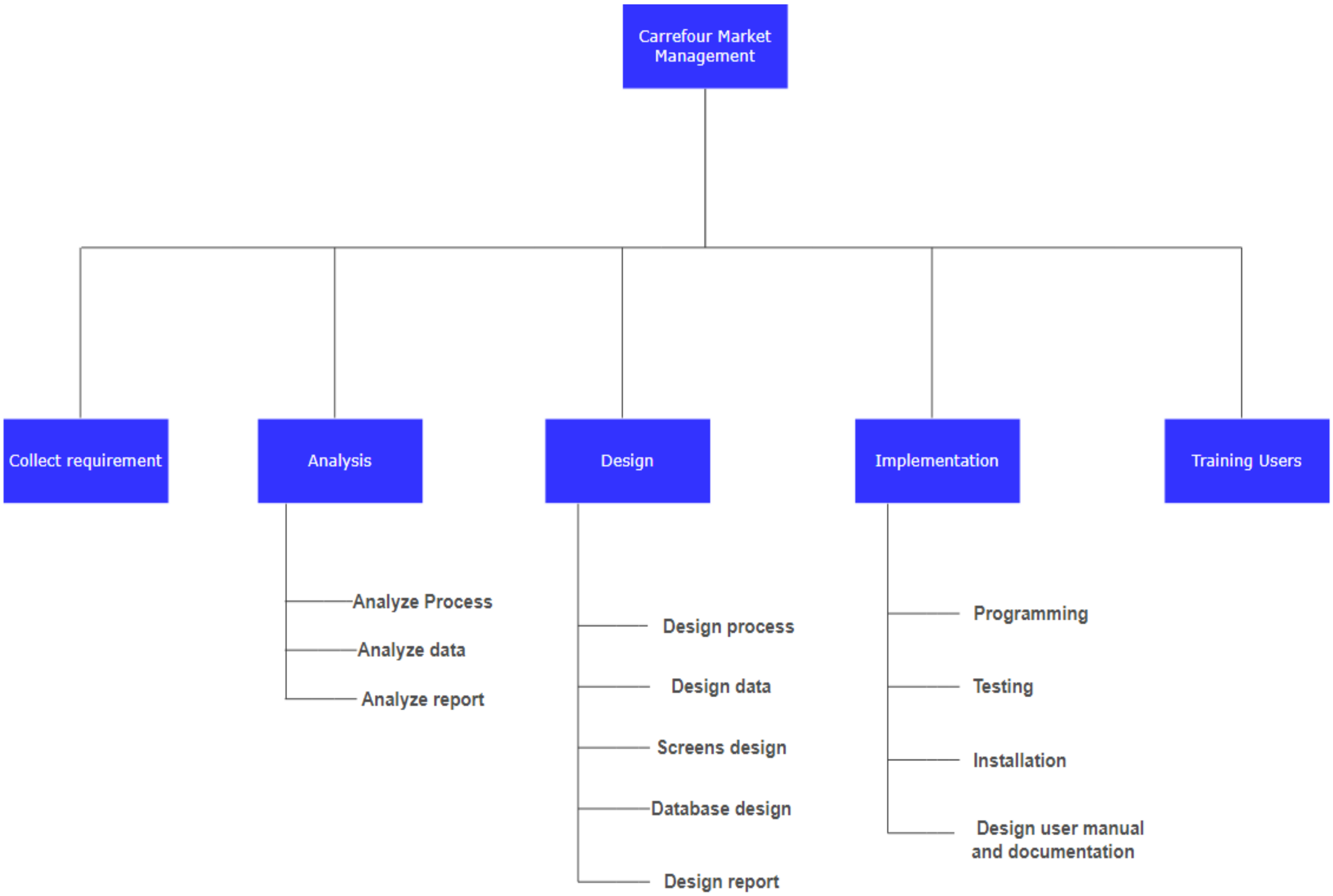
2- Help the customer and meet his needs as possible.

3-try to prevent mistakes as possible and help to handle problems if it happened.

4-provide additional service that would raise the level of service and make the customer satisfied like browsing in the website and make an order then go and take it From the supermarket which solve an undesirable problems like congestion and delivery For people who are not able to go to the supermarket

4-Work Breakdown Structure

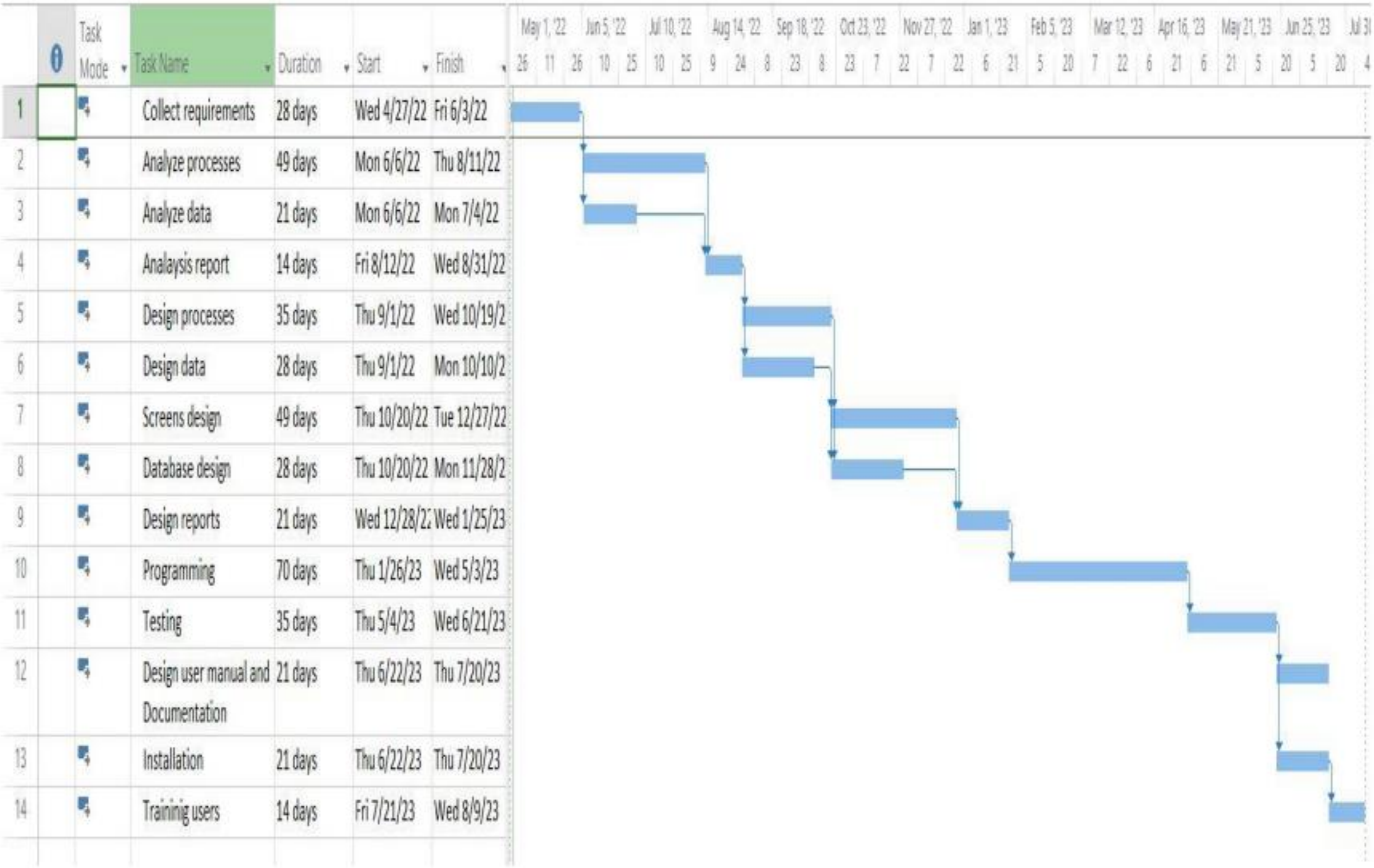
In this phase the project is divided into manageable tasks and then we will use this division to logically order them to ensure a smooth evolution between tasks. Some of these tasks may be performed in parallel, whereas others must follow one another sequentially. All tasks are shown in the next figure.



We used “draw.io” software to make this figure.

5-Gantt Chart

A graphical representation of a project that shows each task as a horizontal bar whose length is proportional to its time for completion. Gantt charts do not show how tasks must be ordered (precedence) but simply show when an activity should begin and end. The next figure will show the gantt chart for the project.



	Task Mode	Task Name	Duration	Start	Finish
1		Collect requirements	28 days	Wed 4/27/22	Fri 6/3/22
2		Analyze processes	49 days	Mon 6/6/22	Thu 8/11/22
3		Analyze data	21 days	Mon 6/6/22	Mon 7/4/22
4		Analaysis report	14 days	Fri 8/12/22	Wed 8/31/22
5		Design processes	35 days	Thu 9/1/22	Wed 10/19/2
6		Design data	28 days	Thu 9/1/22	Mon 10/10/2
7		Screens design	49 days	Thu 10/20/22	Tue 12/27/22
8		Database design	28 days	Thu 10/20/22	Mon 11/28/2
9		Design reports	21 days	Wed 12/28/2:	Wed 1/25/23
10		Programming	70 days	Thu 1/26/23	Wed 5/3/23
11		Testing	35 days	Thu 5/4/23	Wed 6/21/23
12		Design user manual and Documentation	21 days	Thu 6/22/23	Thu 7/20/23
13		Installation	21 days	Thu 6/22/23	Thu 7/20/23
14		Trainig users	14 days	Fri 7/21/23	Wed 8/9/23

We used “Microsoft project” software to make this figure.

6-Project risks

Risk may be good or bad, not always bad as people expect, good project must handle risks

And their solutions.

Risks management consist of processes to control risk:

- Planning risk(strategy)
- Identify risk
- Qualitative analysis(probability, impact)
- Quantitive analysis(Primavera Risk Analysis, Excel)
- Plan risk response(actions)
- Implement risk response
- Monitor/control risk(risk owner)

Planning risk:

Strategy used to act a guideline to manage risk in the project (templates that will use, budget, Exist training or not, timing to manage risk).

Identify risk:

Detect and determine risk in the project.

Qualitative analysis:

Thinking in probability and impact from risk (positive or negative) and estimate probability And impact (no exist numerics in this process) depend on experience.

Quantitive analysis:

Estimate risk numerical and use software depend on monte carlo analysis may be Monte carlo analysis using Primavera Risk Analysis or Excel.

Implement risk response:

Implement actions to deal with risk

Monitor/control risk:

Risk owner control risk and follow people in department (cost, schedule) and send report To project manager

Risk description	probability	impact	strategy
Cost estimates unrealistic	low	high	1-Find more cost-effective solutions 2-find a partner 3-find finance websites ex Zoomaal , kickstarter 4-Reduce requirements
Time estimates unrealistic	Medium	High	1-reduce requirements 2-find a freelancer developer 3-Enable working overtime
Team size	low	low	1-search for new developers that have self-discipline 2-find a freelancer developer 3-divide tasks on available members temporary
Team members Unknowledgeable Of business	low	high	1-provide business courses for team members 2-increase awareness of business problems
Narrow Knowledge Level of users	medium	Low	1-Increase training duration

7-Safety

Safety is a characteristic of a system that reflects the ability of the system to operate, normally or abnormally, without risk of causing human injury or death and without harming the system environment

Safety is one of dependability`s attributes

It is important to consider software integrity because most devices whose failure is critical. now include software-based control systems

Safety requirements are often exclusivist, that is, they rule out undesirable situations rather than specify which system services are required. These generate functional safety requirements

Safety Achievement

the The key to assuring safety is to ensure either that accidents do not occur or that consequences of an accident are minimal. This can be achieved in three ways:

1-Hazard avoidance

The system is designed so that hazards are avoided

-By Maintaining Safe Practices in Supermarkets with Food Monitoring Systems

2-Hazard detection and removal

The system is designed so that hazards are detected and removed before they result in an accident.

-If the customer's visa is not valid or there is no balance, the payment will not be processed through it

3-Damage limitation

The system may include protection features that minimize the damage that may result from an accident.

-If the product is about to expire or is damaged, give me a warning

----

*A system without safety is inefficient and unreliable

Safety is one of the main factors for creating a good reputation for the place, and it is one of the main reasons for attracting customers because of its importance

8-Security

The security of a system property that reflects the system's ability to protect itself from accidental or deliberate external attack.

Security is essential as the most systems are networked so that external access to the system through the internet is possible.

Security is an essential pre-requisites for availability, reliability and safety.

Security assurance:

- vulnerability avoidance

The system is designed so that vulnerabilities do not occur. For example, if there is no external network connection then external attack is impossible.

- Attack detection and elimination

The system is designed so that attacks on vulnerabilities are detected and neutralized before they result in an exposure. For example, virus checkers find and remove viruses before they infect a system.

- Exposure limitation and recovery

The system is designed so that the adverse consequences of a successful attack are minimized .for example; a backup policy allows damaged information to be restore.

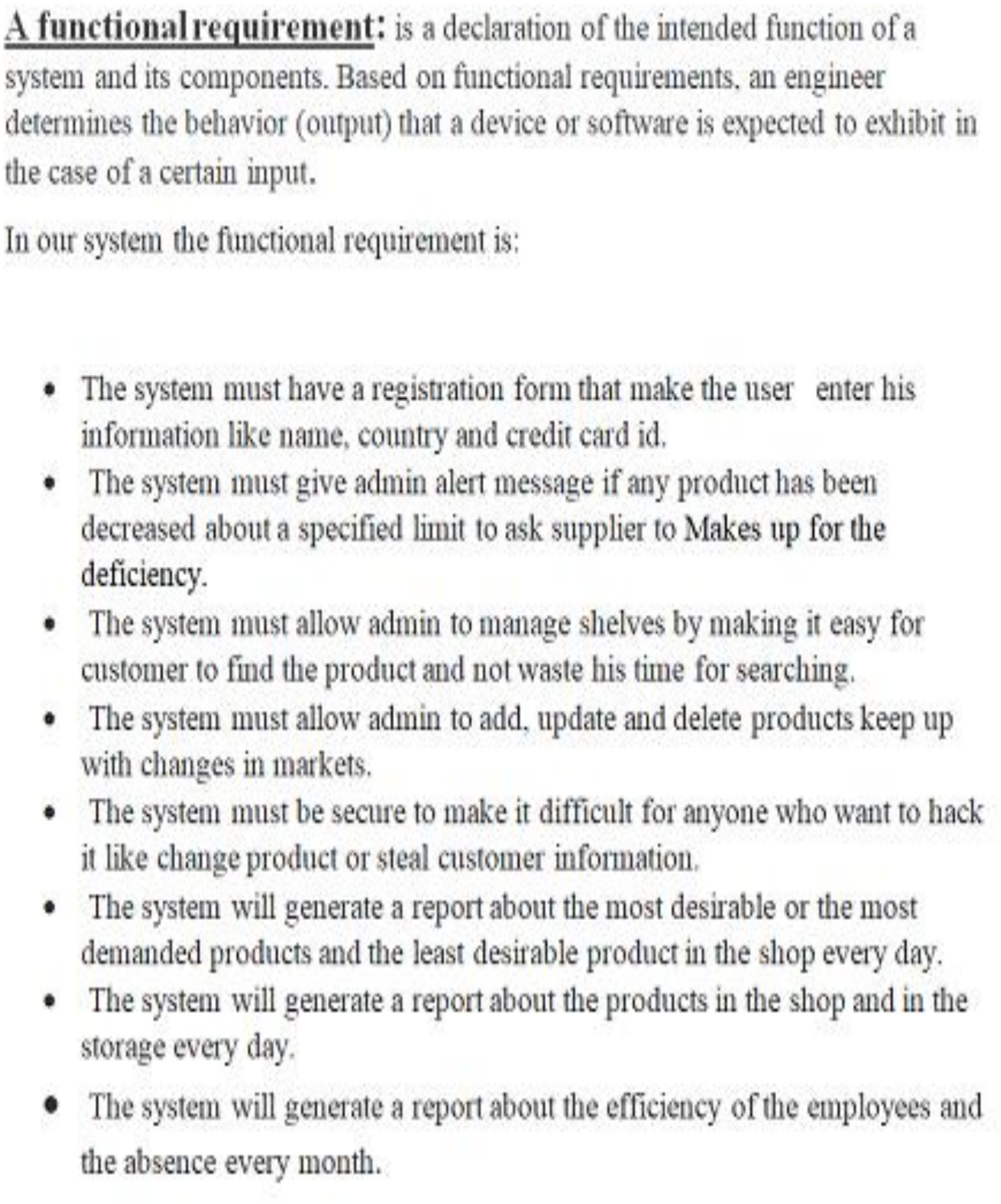
There are many common or routine tasks to improve security :

1. **Change default usernames and passwords.** Most, if not all devices, have default usernames and passwords that can easily be found by a child using a search engine. Default passwords should be changed immediately and even better is to also change, delete or disable the default username as well.
2. **Don't share passwords.** Every user of a system should have their own username and password. This ensures that when that user leaves the organization, their account can simply be disabled or deleted and there is no need to change the shared account details for every system the user had access to (which in reality just doesn't happen).
3. **Use strong authentication.** Using strong passwords that are a reasonable length with a combination of letters, numbers and special characters and don't include dictionary words ensures that passwords can't be easily compromised. Better yet is to use digital certificates with the keys that are generated within a token or device which not only delivers strong authentication but also provides non-repudiation.
4. **Use centralized authentication.** Using a centralized authentication system such as Active Directory, RADIUS or LDAP simplifies the process of managing user information as there is a single system that needs to be administered instead of multiple systems.
5. **Restrict access.** Access to systems should be restricted to the minimum level that is required for a user to perform the tasks they need to perform. In addition, firewalls should be used to segregate and isolate systems so that an issue or attack on one system is less likely to impact other systems.
6. **Integrate redundancy.** Redundancy should be integrated to ensure the system continues to operate in the event of a failure or attack. In many cases, the cost of lost business is far greater than the additional cost of redundancy.
7. **Don't forget physical security.** Security is about layers and one of those layers is physical access. If an attacker can obtain physical access to a system, their task will be made simpler so be sure that physical access is only available to those that need it.
8. **Maintain backups.** From time to time, things will go wrong. And when this happens, it's important that you have a backup that is off-line and stored in a different physical location.

Examples Of Security Terminology (Carrefour management system)

Term	Example
Asset	The records of each Customer that is receiving and received products
Exposure	potential financial loss from the future customer who does not seek to purchase their products from our site because they do not trust the system to keep their data.
vulnerability	A week password system which makes it easy for users to set guessable passwords . User ids that are the same as names.
Attack	An impersonation of an authorized user.
Threat	An unauthorized user will gain access to the system by guessing the credentials (login name and password) of an authorized user.
control	A password checking system that disallows user passwords that are proper names or words that are normally included in a dictionary .

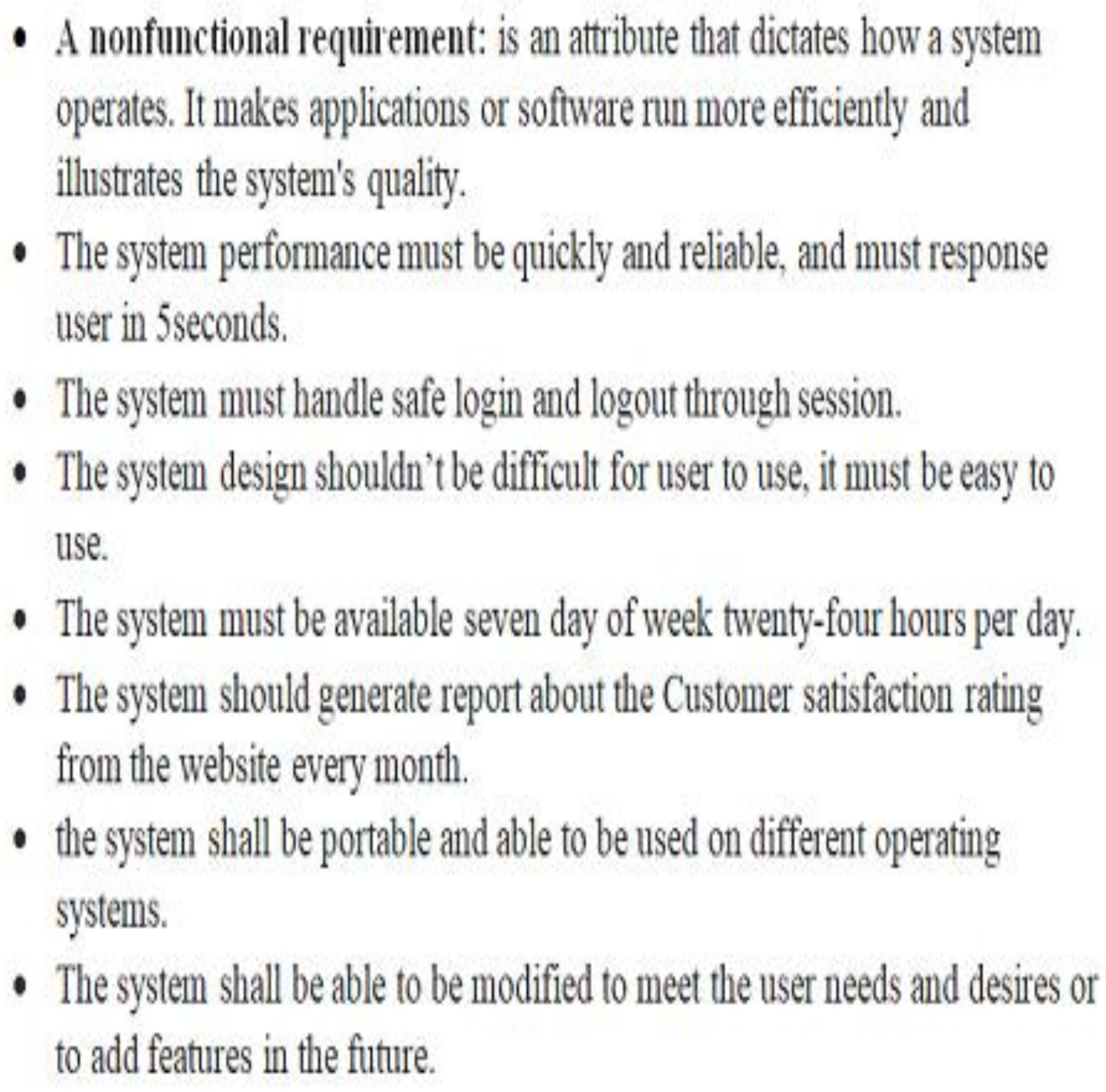
9-Function and nonfunctional requirement



A functional requirement: is a declaration of the intended function of a system and its components. Based on functional requirements, an engineer determines the behavior (output) that a device or software is expected to exhibit in the case of a certain input.

In our system the functional requirement is:

- The system must have a registration form that make the user enter his information like name, country and credit card id.
- The system must give admin alert message if any product has been decreased about a specified limit to ask supplier to Makes up for the deficiency.
- The system must allow admin to manage shelves by making it easy for customer to find the product and not waste his time for searching.
- The system must allow admin to add, update and delete products keep up with changes in markets.
- The system must be secure to make it difficult for anyone who want to hack it like change product or steal customer information.
- The system will generate a report about the most desirable or the most demanded products and the least desirable product in the shop every day.
- The system will generate a report about the products in the shop and in the storage every day.
- The system will generate a report about the efficiency of the employees and the absence every month.



- **A nonfunctional requirement:** is an attribute that dictates how a system operates. It makes applications or software run more efficiently and illustrates the system's quality.
- The system performance must be quickly and reliable, and must response user in 5seconds.
- The system must handle safe login and logout through session.
- The system design shouldn’t be difficult for user to use, it must be easy to use.
- The system must be available seven day of week twenty-four hours per day.
- The system should generate report about the Customer satisfaction rating from the website every month.
- the system shall be portable and able to be used on different operating systems.
- The system shall be able to be modified to meet the user needs and desires or to add features in the future.

10-Introduction to UML

UML Design

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing and documenting the software system and its components. It is a graphical language , which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for:

- Visualizing
- Specifying
- Constructing
- Documenting

Visualizing

Through UML we see or visualize an existing system and ultimately we visualize how the is system going to be after implementation. Unless we think, we cannot implement.UML helps to visualize, how the components of the system communicate and interact with each other.

Specifying

Specifying means building, models that are precise,unambiguous and complete UML addresses the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

Constructing

UML models can be directly connected to a variety of programming language through mapping a model from UML to a programming language like JAVA or C++ or VB. Forward Engineering and Reverse Engineering is possible through UML.

Documenting

The Deliverables of a project apart from coding are some Artifacts, which are critical in controlling,measuring and communicating about a system during its developing

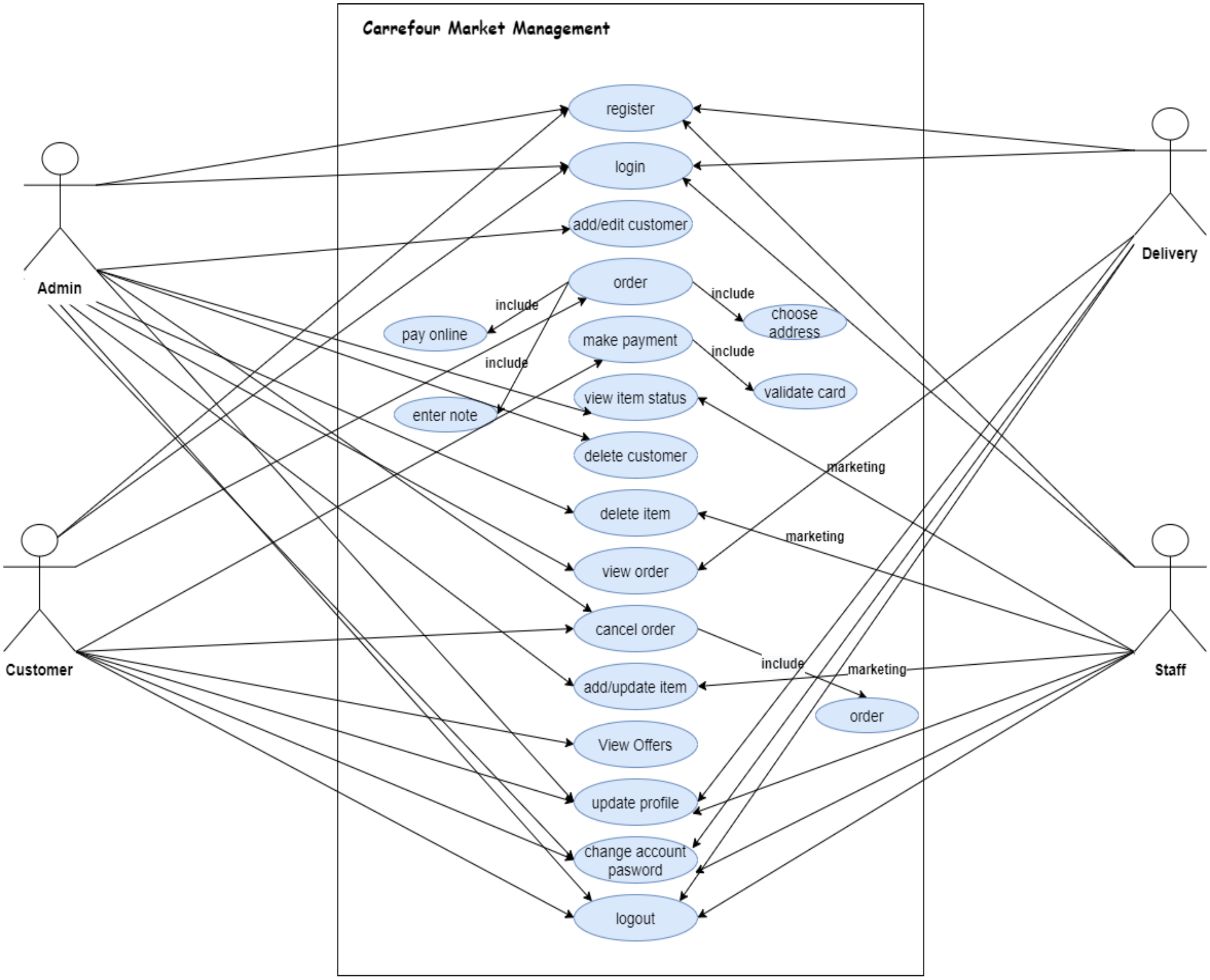
11-UML Diagram

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs . you draw diagram to visualize a system from different perspective, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams , or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system. For this reason, the UML includes nine such diagrams:

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Activity diagram
- Component diagram
- Deployment diagram

12-usecase diagram

A usecase diagram in the Unified Modeling Language(UML) is atype of behavioral diagram defined by and created form a use-case analysis.its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals(represented as use cases),and any dependencies between those use cases.



Use Case#1	**Register**
Primary Actor	Admin,Customer,Delivery,Staff.
Pre-condition	The Admin,Customer,Delivery and staff should have entered valid email and phone To stored in database.
Main Scenario	*The Admin or Customer or Delivery or staff create account throw enter valid email ,phone, strong password,address and credit number. *if the account created successfully The Admin or Customer or Delivery or staff can enter to the system or log in the system anytime.
Alternate Scenarios	*Create account fail. *Enter invalid email or phone or credit number. *if the account doesn't create try to change email,phone and username.

Use Case#2	Login
Primary Actor	Admin,Customer,Delivery,Staff.
Pre-condition	The Admin,Customer,Delivery and staff should have been register to the system
Main Scenario	*The Admin or Customer or Delivery or staff must register to the system before login. *if username and password that entered are correct admin,customer,delivery and staff login successfully to the system. *message info to user he login successfully . * every person has a view and every person has personal transaction.
Alternate Scenarios	*login fail to the system. *Enter invalid username or password . *if the account doesn't enter to the system try to member the correct account.

Use Case#3	ADD/Edit customer
Primary Actor	Admin.
Pre-condition	The Admin should have been login to the system.
Main Scenario	*The Admin should have been login to the system. *admin enter to admin mode and detect each transaction (add/edit)customer to the system. *must enter valid username and password. * save changes username added to the system or updated.
Alternate Scenarios	*login fail to the system. *Enter invalid username or password . *add customer and this customer already exist on the system.

Use Case#4	**Order**
Primary Actor	Customer.
Pre-condition	The customer should have been login to the system.
Main Scenario	*The customer should have been login to the system. *customer view the products and enter id of item that you want. *id of item should be correct. * enter the address that want to order to arrive at him. *enter notes . *save changes and order item.
Alternate Scenarios	*login fail to the system. *Enter invalid id of product . *enter address out of zone.

Use Case#5	**Make payment**
Primary Actor	Customer.
Pre-condition	The customer should have been login to the system.
Main Scenario	*The customer should have been login to the system. *customer choose payment way that you want. *if the customer choose credit card the system check if credit valid and has enough money. * if the customer choose book item select when receiving . *save changes.
Alternate Scenarios	*login fail to the system. *Enter invalid number of credit card . *doesn’t have enough money to buy this item.

Use Case#6	**View item status**
Primary Actor	Admin, Staff.
Pre-condition	The admin,staff should have been login to the system.
Main Scenario	*The admin,staff should have been login to the system. *admin, staff view quantity of item, expire date and date if an item near to be expired law price this item. *save changes.
Alternate Scenarios	*login fail to the system. *item was expired and admin/staff go to delete process and delete this item .

Use Case#7	Delete customer
Primary Actor	Admin.
Pre-condition	The admin should have been login to the system.
Main Scenario	*The admin should have been login to the system. *admin should have been enter valid id of customer if id correct delete this customer have the same id and deleted process performed successfully. *save changes.
Alternate Scenarios	*login fail to the system. *enter id doesn’t exist on the system .

Use Case#8	**Delete item**
Primary Actor	Admin,staff.
Pre-condition	The admin,staff should have been login to the system.
Main Scenario	*The admin,staff should have been login to the system. *admin should have been enter valid id of item if id correct delete this item have the same id and deleted process performed successfully. *save changes.
Alternate Scenarios	*login fail to the system. *enter id doesn't exist on the system .

Use Case#9	View order
Primary Actor	Admin,delivery.
Pre-condition	The admin,delivery should have been login to the system.
Main Scenario	*The admin,delivery should have been login to the system. *admin,delivery see orders and detect if accept this order or reject. *save changes.
Alternate Scenarios	*login fail to the system. *accept order and this order out of zone ..update it and reject.

Use Case#10	Cancel order
Primary Actor	Admin,Customer.
Pre-condition	The admin,customer should have been login to the system.
Main Scenario	*The admin,customer should have been login to the system. *admin,customer enter id of order and this id must be exist on the system and cancel order .order was canceled successfully and detect if he make order or logout. *save changes.
Alternate Scenarios	*login fail to the system. *enter id doesn’t exist on the system . *make another order end enter id doesn’t exist on the system .

Use Case#11	Add/update item
Primary Actor	Admin,staff.
Pre-condition	The admin,staff should have been login to the system.
Main Scenario	*The admin,staff should have been login to the system. *admin,staff detect add new item or update item and enter id of item that you want add or update should id exist on the system and add/update item successfully. *save changes.
Alternate Scenarios	*login fail to the system. *enter id doesn't exist on the system . *add item already exist on the system.

Use Case#12	**View offers**
Primary Actor	customers.
Pre-condition	The customer should have been login to the system.
Main Scenario	*The customer should have been login to the system. *system detect via id if the customer normal or special every type of customers has different offers after view offers customer detect if he want to order item or logout . *save changes.
Alternate Scenarios	*login fail to the system. *enter id doesn't exist on the system when make an order .

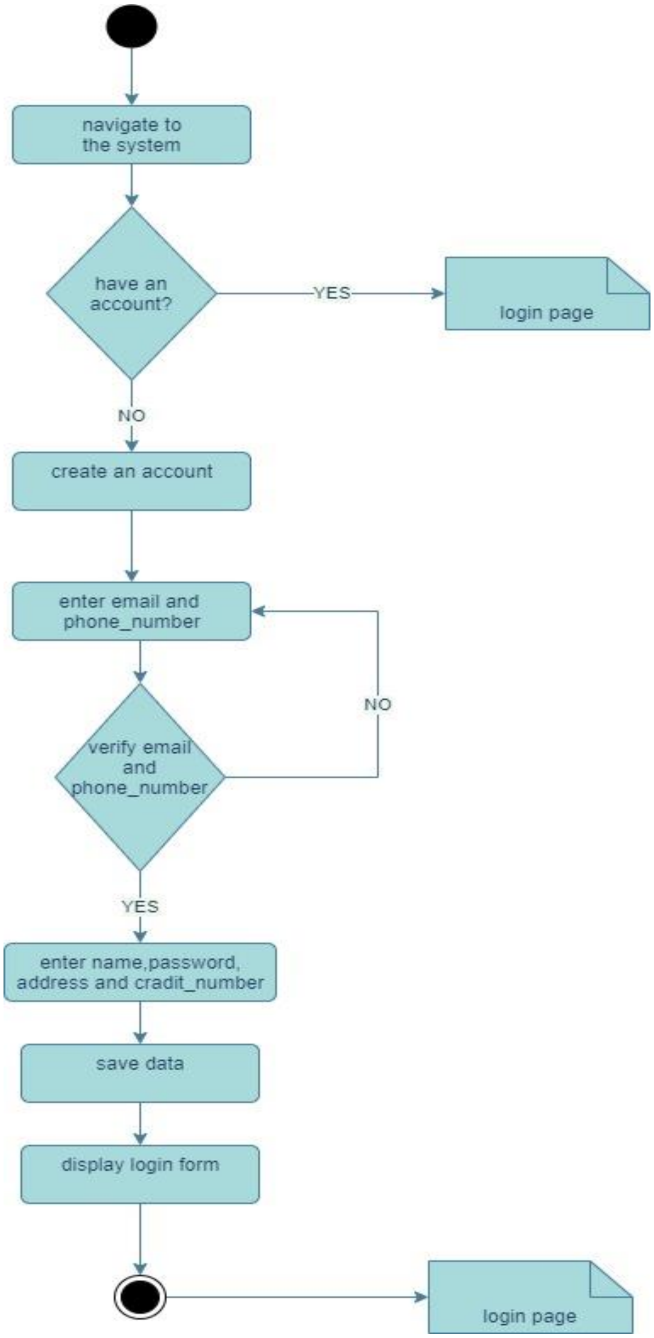
Use Case#13	**Update profile**
Primary Actor	Admin,staff,customer,delivery.
Pre-condition	The admin,staff ,delivery,customer should have been login to the system.
Main Scenario	*The admin,staff,delivery,customer should have been login to the system. *admin,staff,delivery,customer go to setting mode and update their profile profile was updated successfully. *save changes.
Alternate Scenarios	*login fail to the system. *when update profile enter invalid data.

Use Case#14	Change account password
Primary Actor	Admin,staff,customer,delivery.
Pre-condition	The admin,staff ,delivery,customer should have been login to the system.
Main Scenario	*The admin,staff,delivery,customer should have been login to the system. *admin,staff,delivery,customer go to setting mode and change account password and password was updated successfully. *save changes.
Alternate Scenarios	*login fail to the system. *when change password enter weak password or password less than less than 8 character .

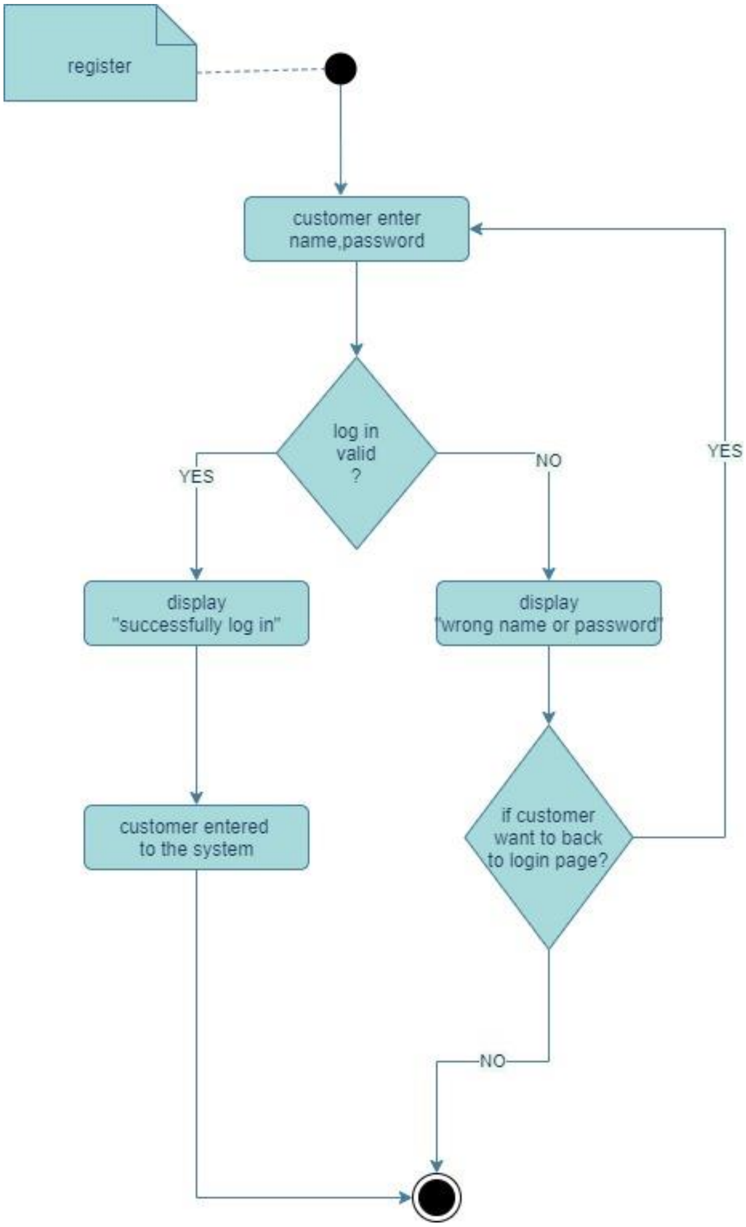
Use Case#15	**Log out**
Primary Actor	Admin,staff,customer,delivery.
Pre-condition	The admin,staff ,delivery,customer should have been exist on the system.
Main Scenario	*admin,staff,delivery,customer go to setting mode log out from the system Message info you log out successfully. *save changes.
Alternate Scenarios	*when log out from the system want to save username and password or not.

13-Activity Diagram

An Activity diagram is another important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart.

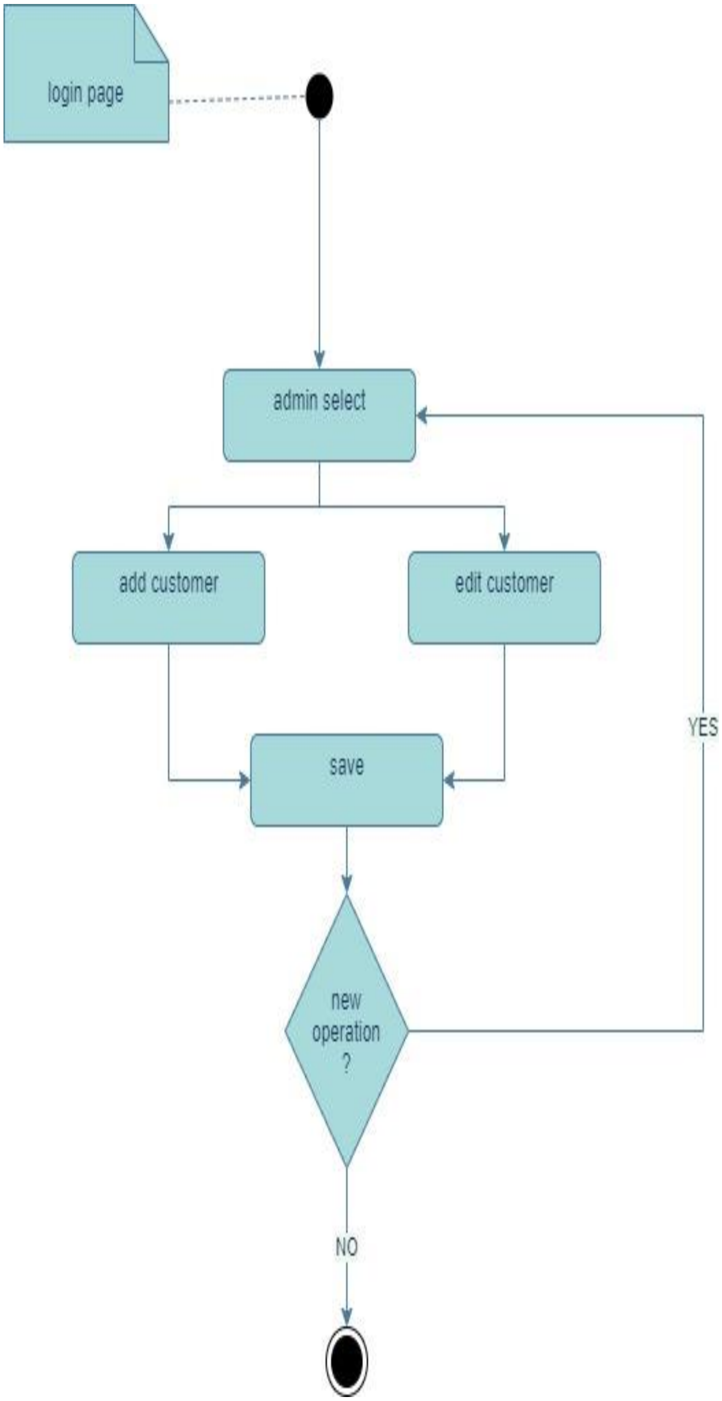


Register Person detect if he has account or not if he hasn't account create account ,enter email ,number if it valid complete and enter name,password,credit number and save it

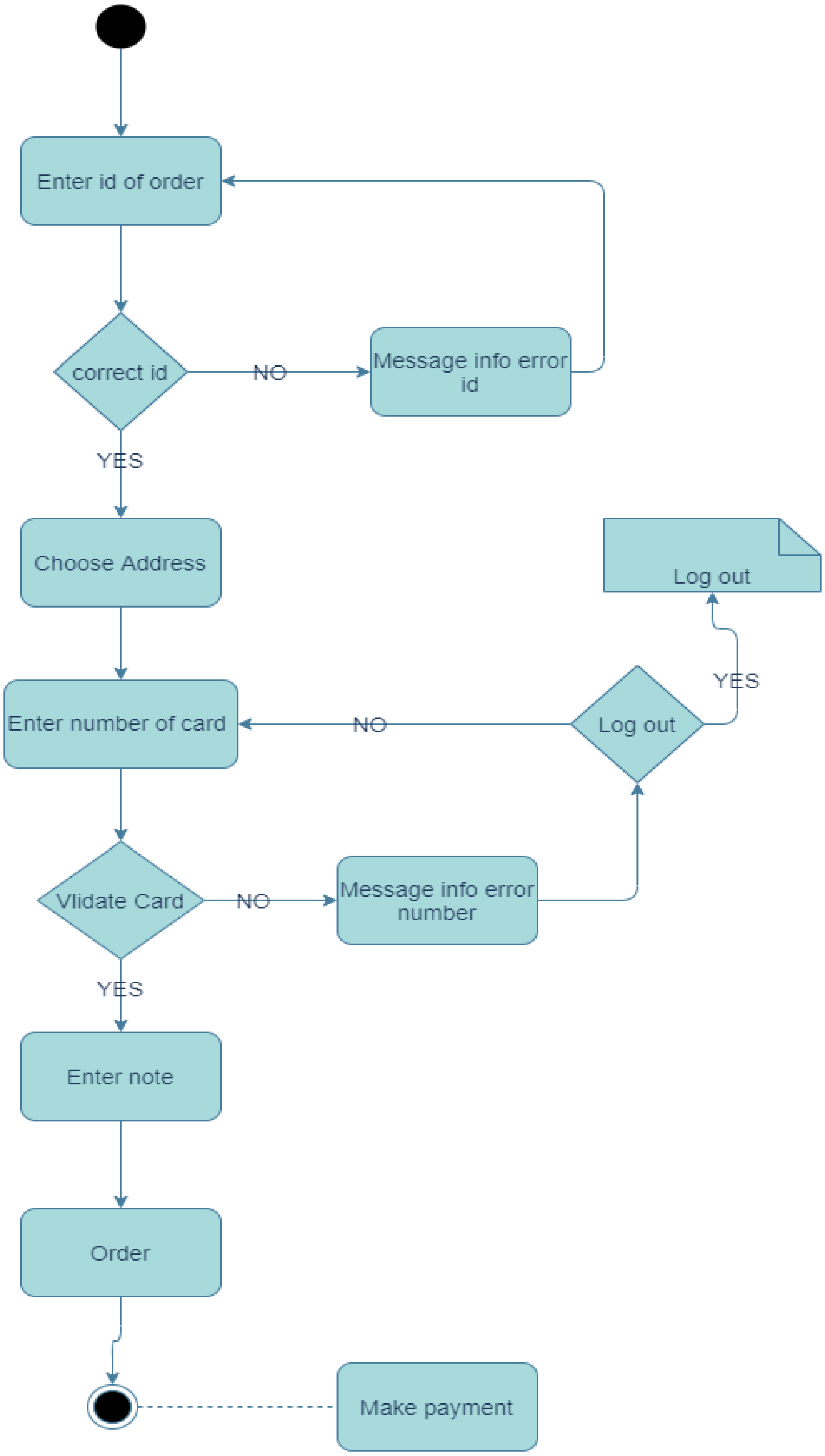

register
customer enter name,password
log in valid ?
YES
NO
display "successfully log in"
display "wrong name or password"
customer entered to the system
if customer want to back to login page?
YES
NO

Login

Person enter name and password if it true login to the system else try again

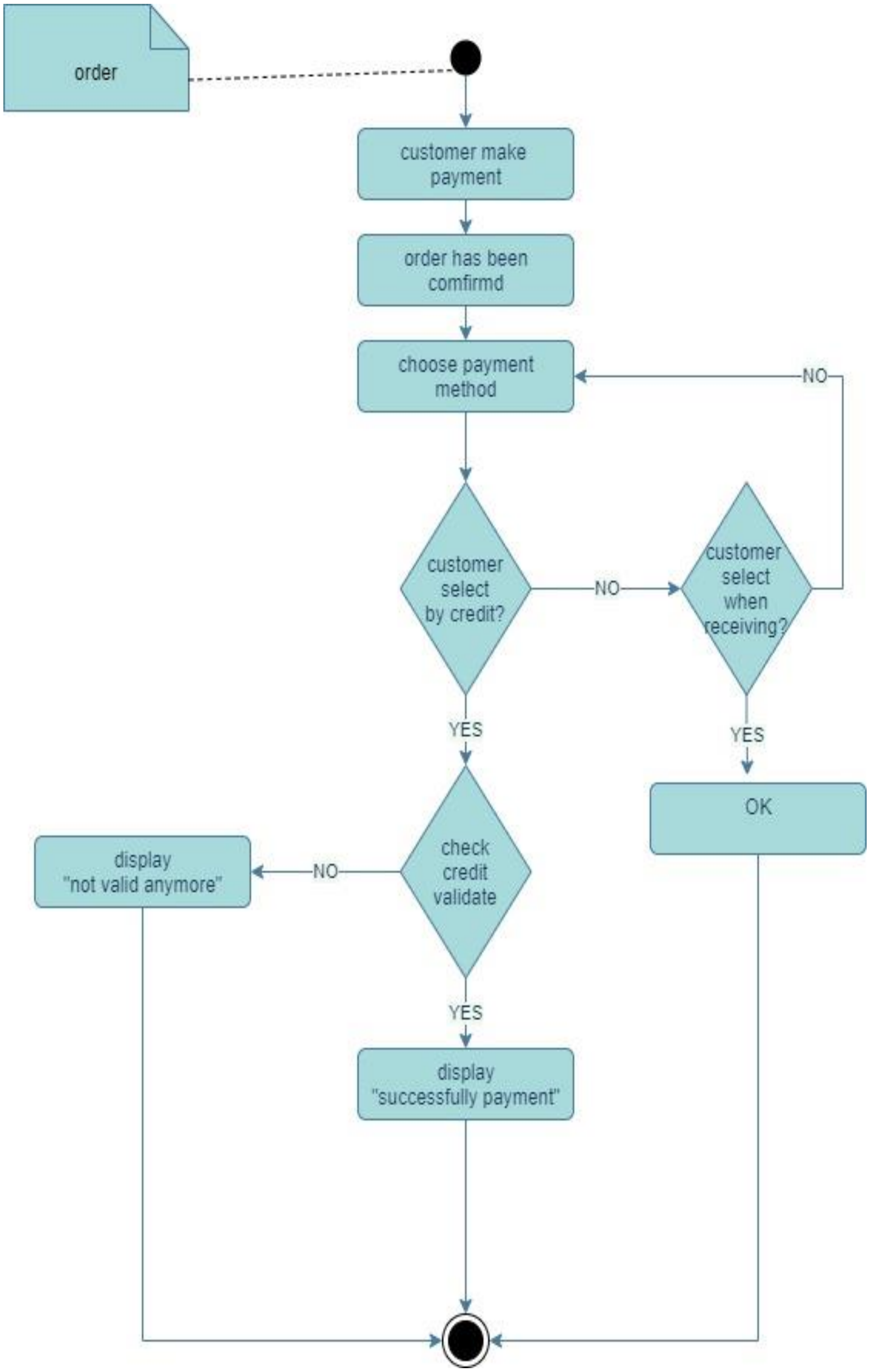


Add/edit customer Admin detect if he want to add or edit customer and save it

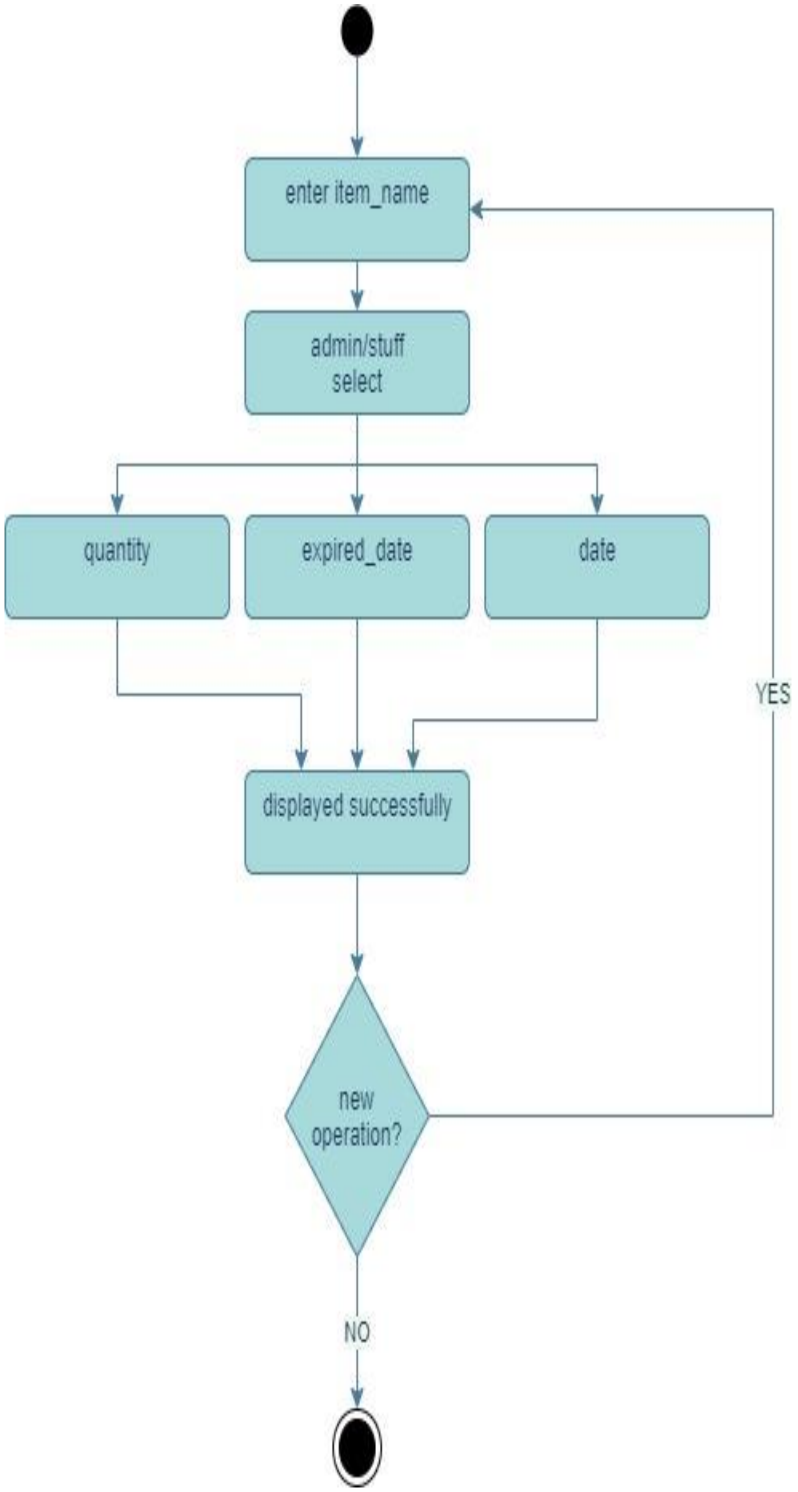


Order

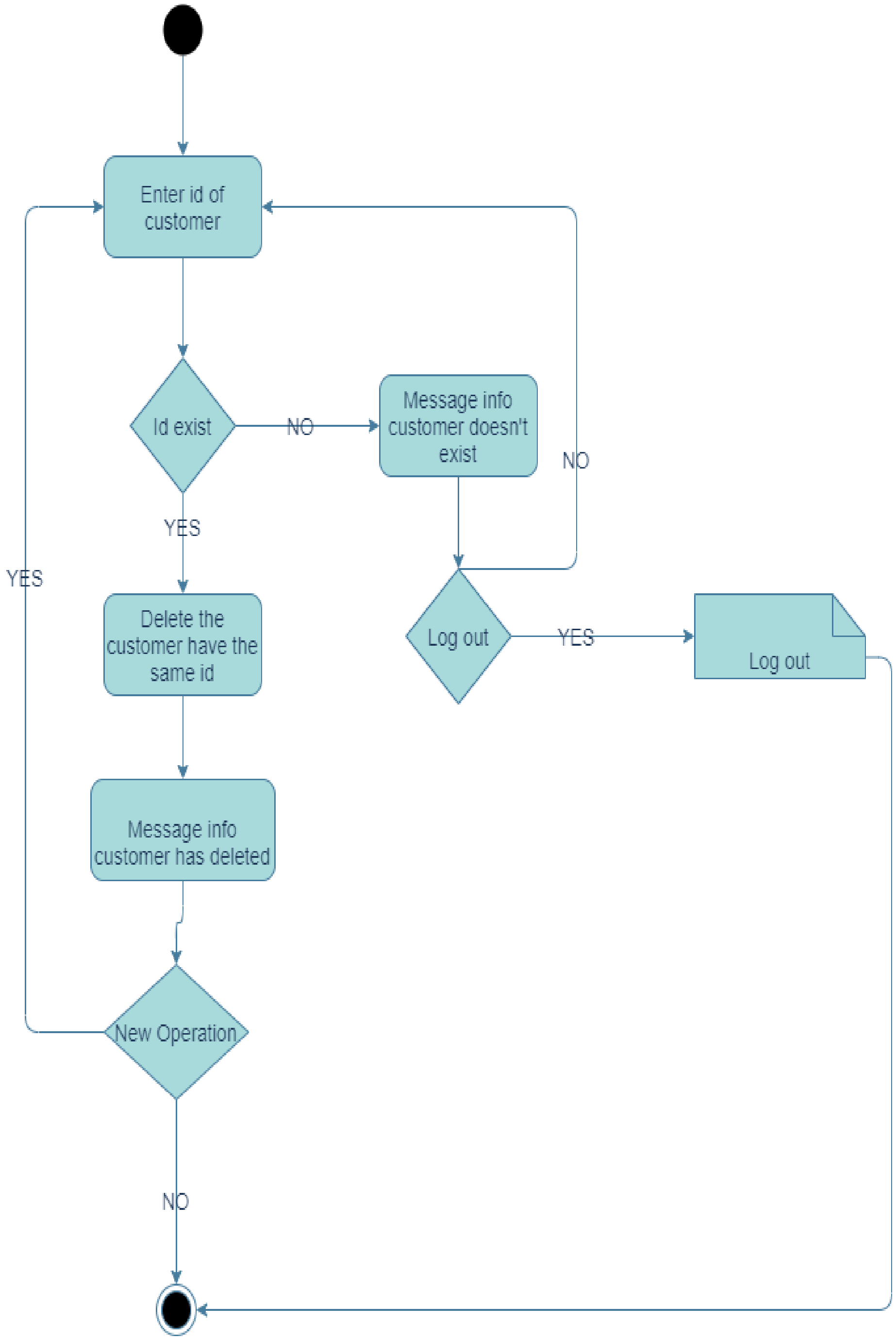
Customer perform this process ,enter id of order if id is exist on the system choose address and enter number of credit must be true and enter notes and make order


order
customer make payment
order has been comfirmd
choose payment method
customer select by credit?
NO
customer select when receiving?
NO
YES
YES
OK
check credit validate
NO
display "not valid anymore"
YES
display "successfully payment"

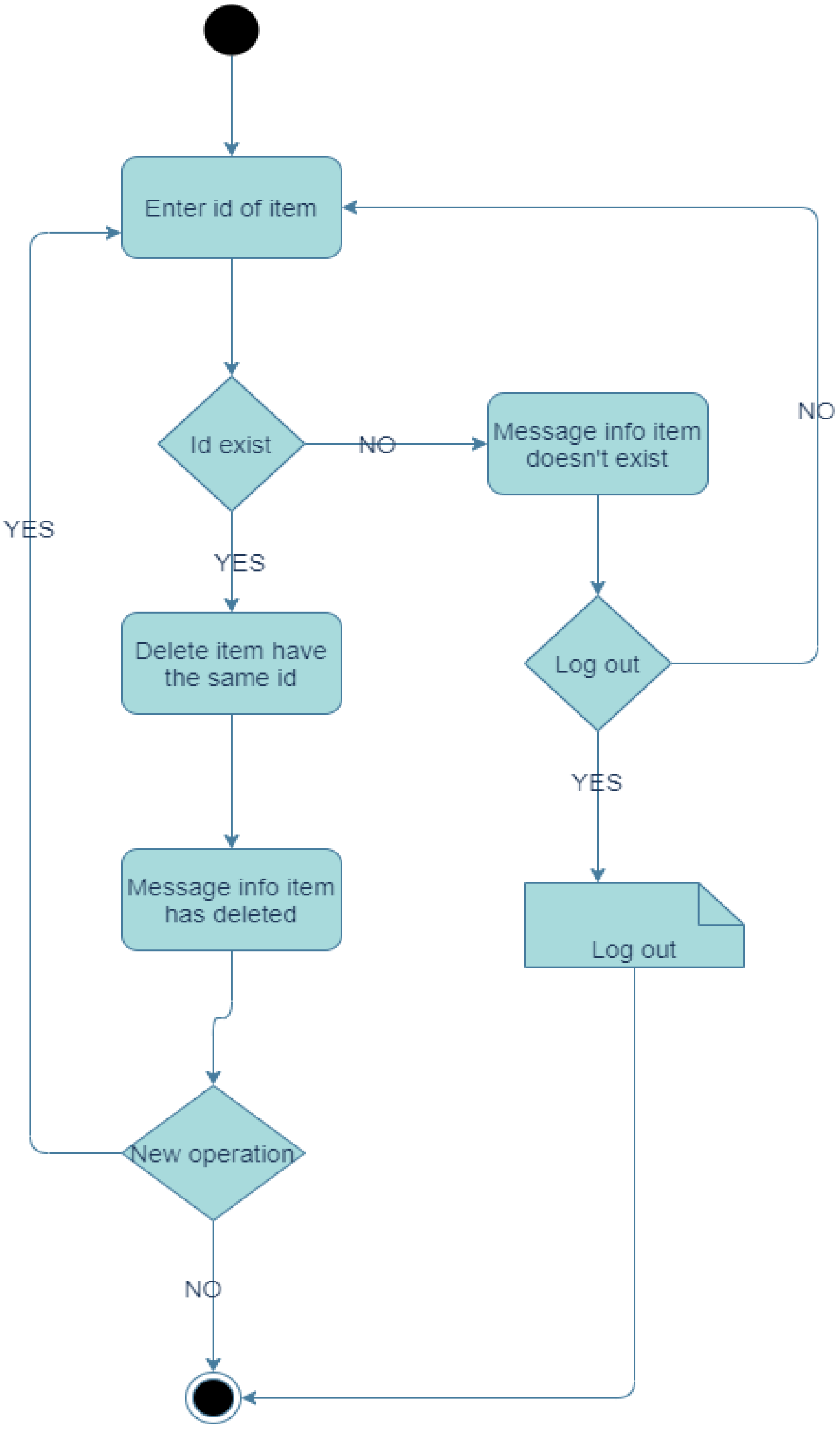
Make payment Customer perform this process after ordering choose payment method if he select credit method system check if credit available or not and display it else customer detect time to go to supermarket and take order



View item status Admin and staff perform this process and view quantity ,expire_date and date of item and display on the screen

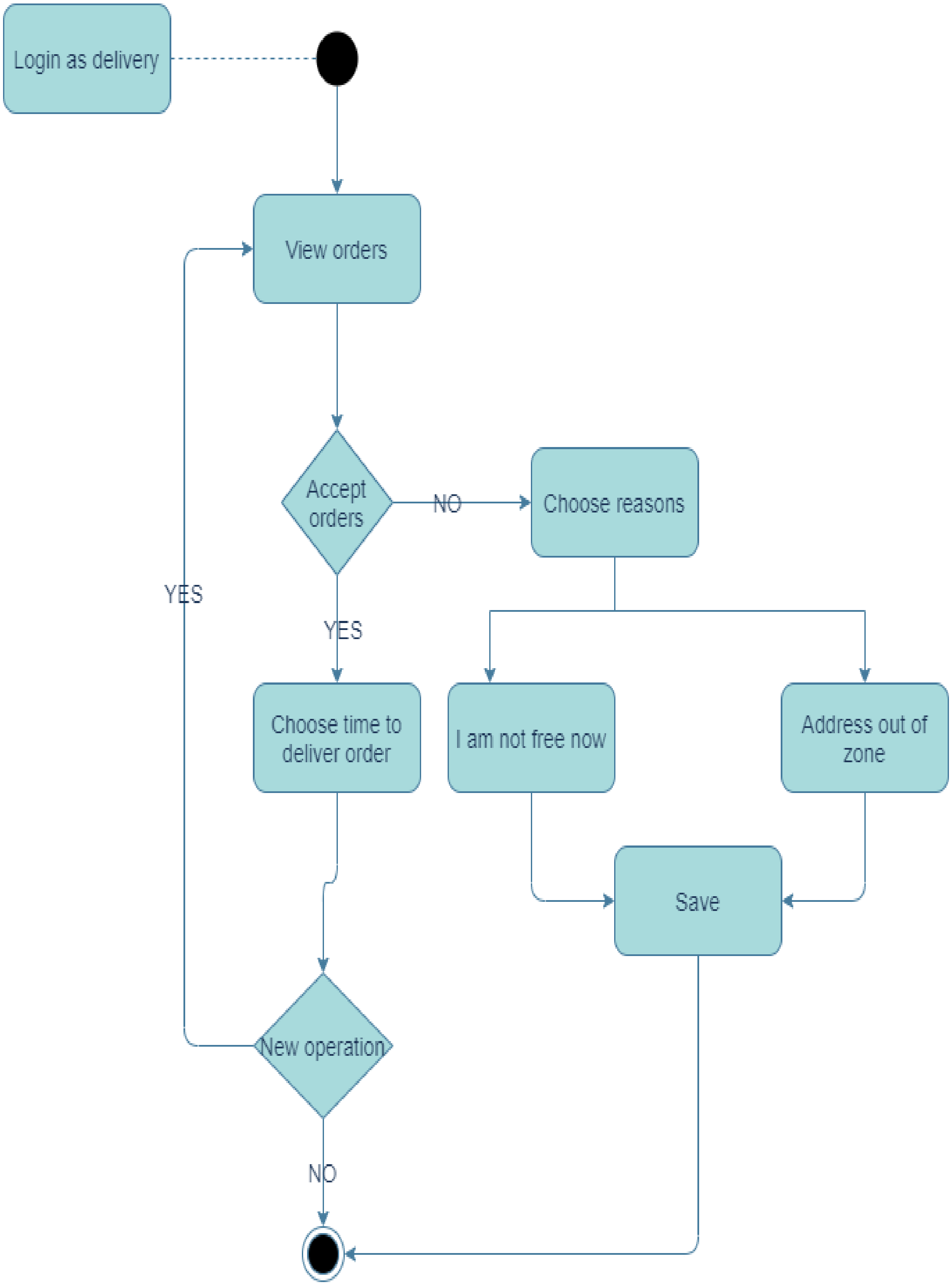

Enter id of customer
Id exist
NO
Message info customer doesn't exist
NO
YES
YES
Delete the customer have the same id
Log out
YES
Log out
Message info customer has deleted
New Operation
NO

Delete customer Enter id of customer and if id is exist delete this customer from the system else logout from the system


Enter id of item
Id exist
NO
Message info item doesn't exist
NO
YES
YES
Delete item have the same id
Log out
YES
Message info item has deleted
Log out
New operation
NO

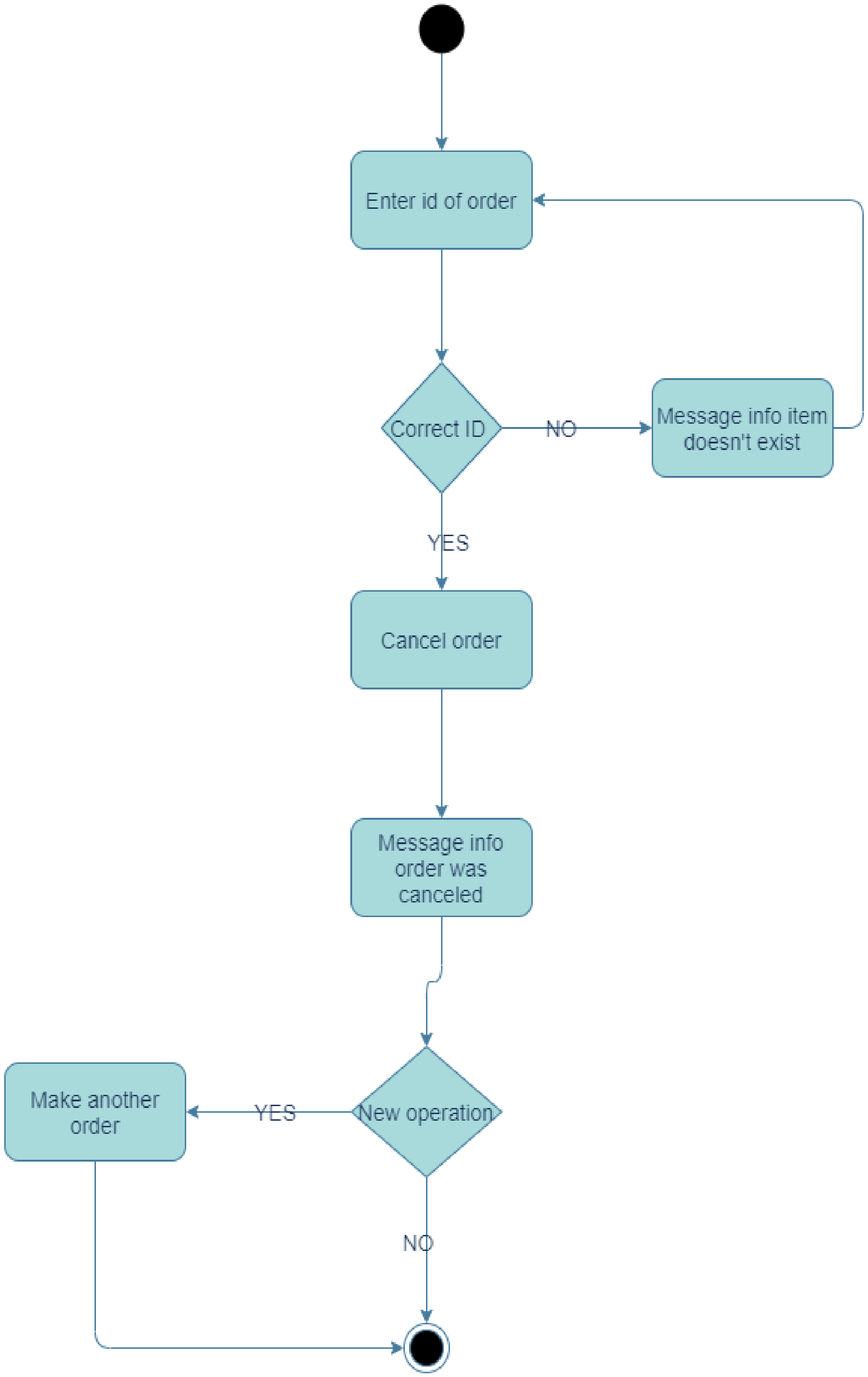
Delete item

Enter id of item and detect if id is exist perform deletion process else logout from the system



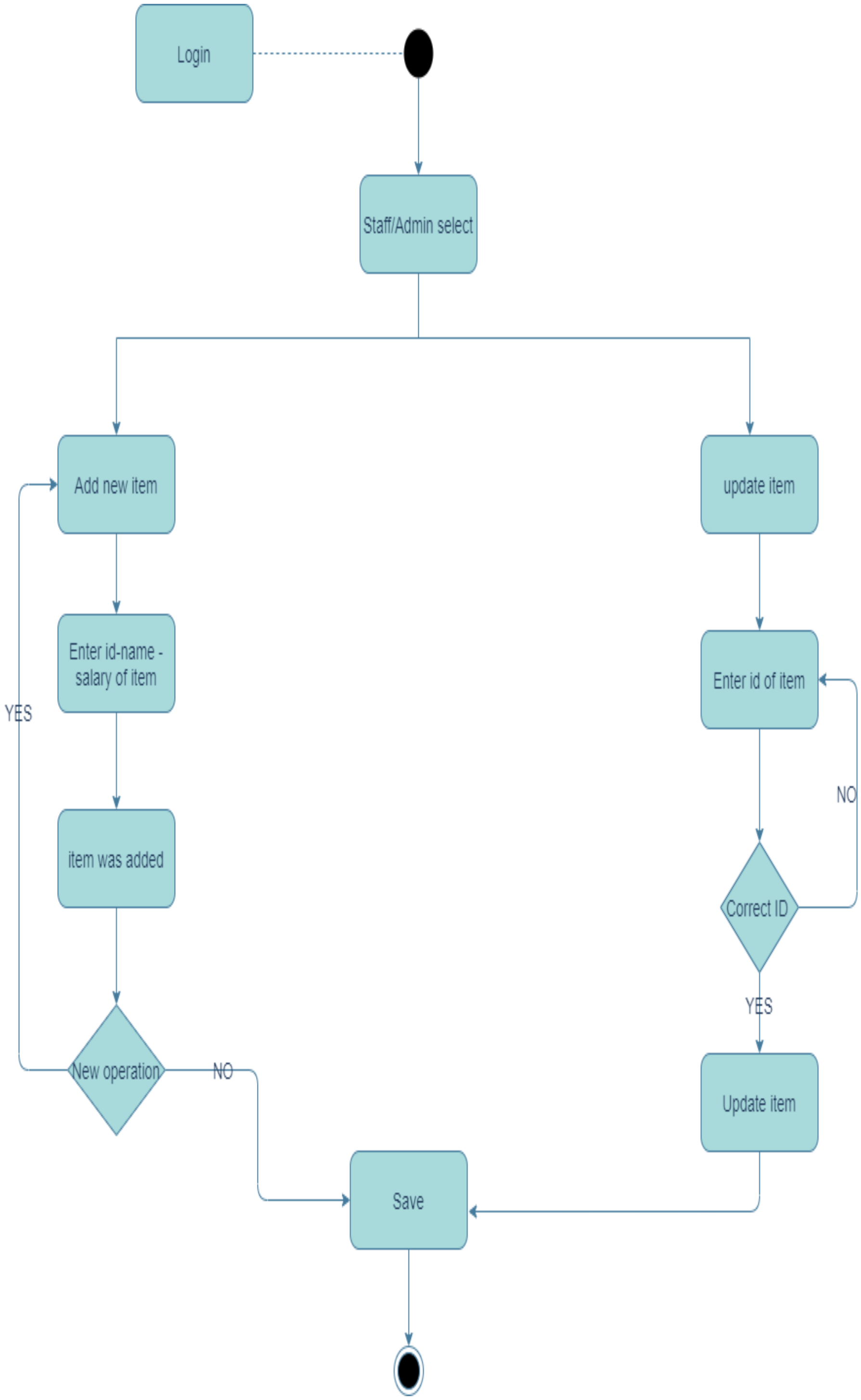
View orders

A man of delivery view orders comes from customers and if order in the zone and free now accept this order

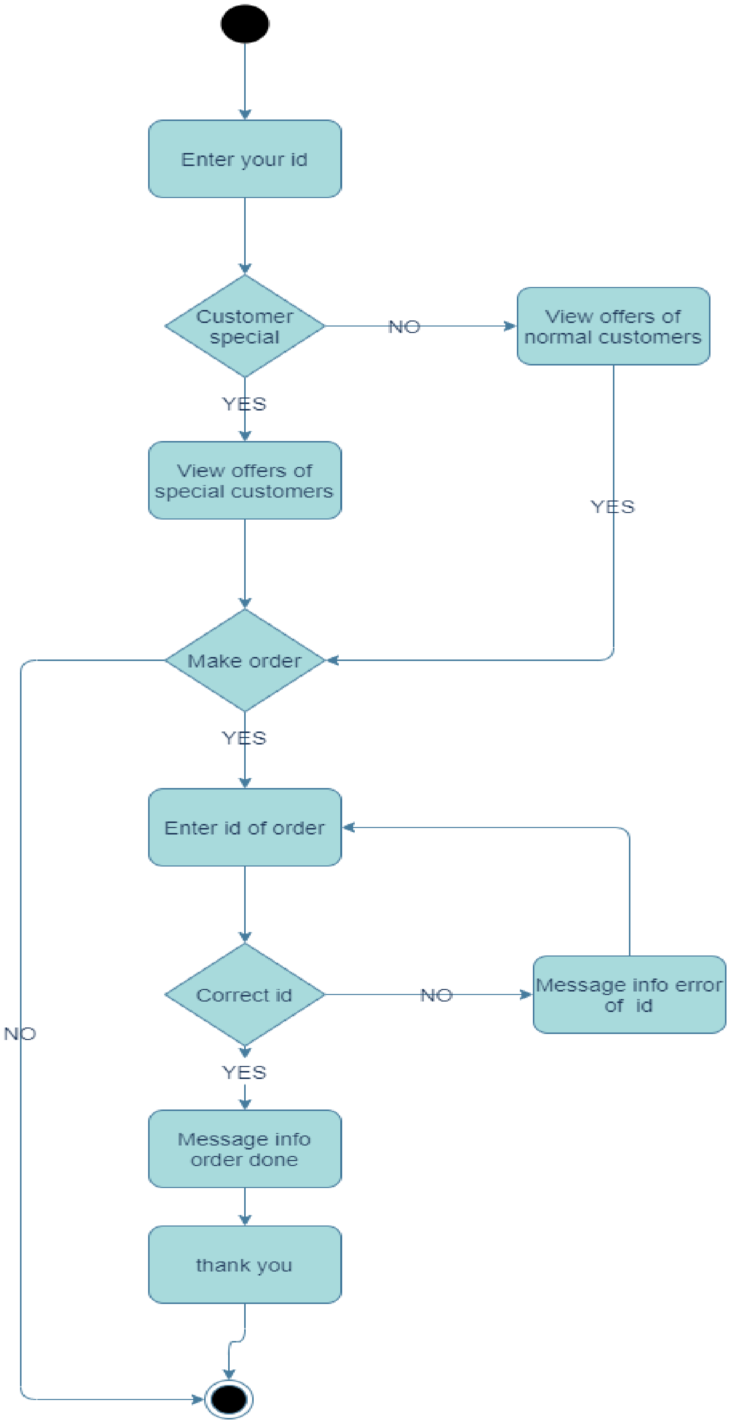


Cancel order

Enter id of order and if id of order is correct choose cancel order and if he want perform another order this option available

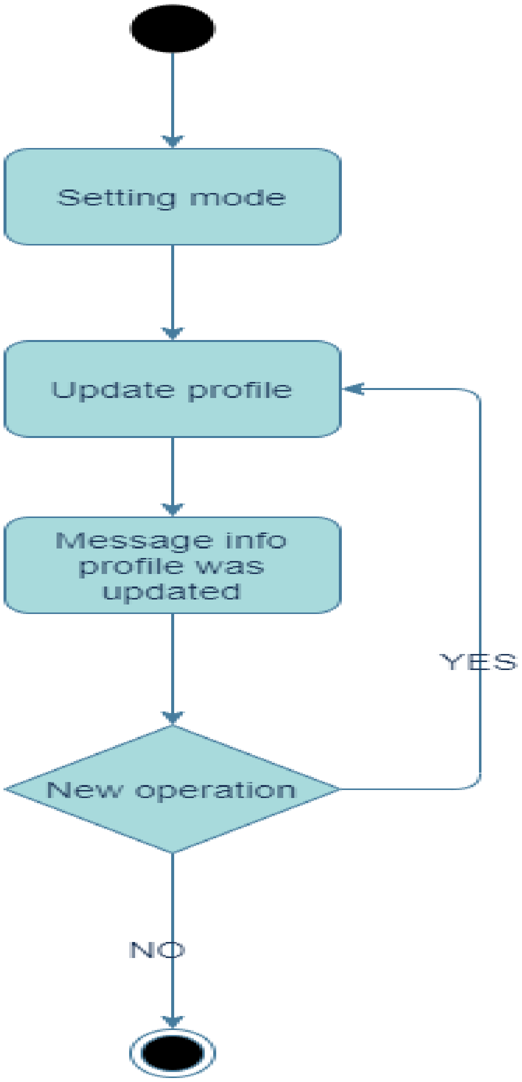


Add/update item Admin and staff perform this case detect if add new item or update item in each of them enter id of item and add or update item



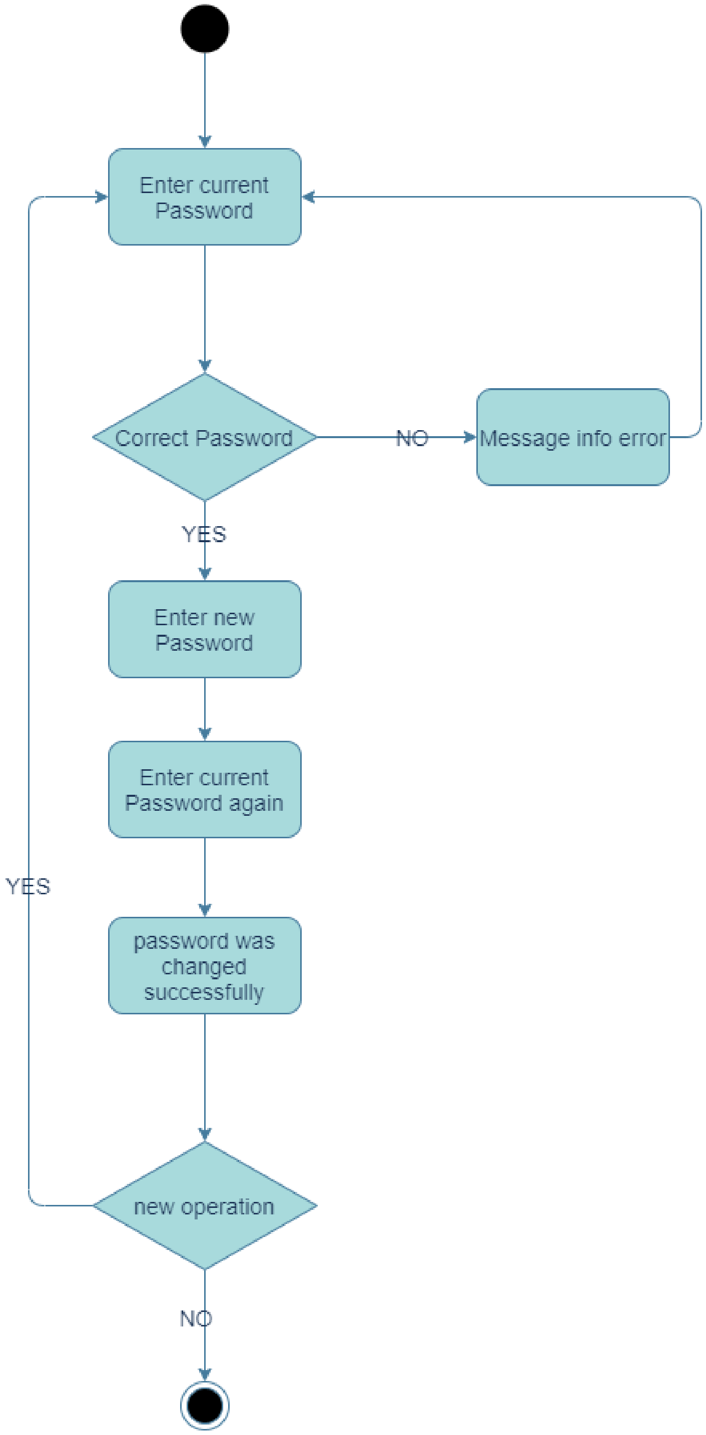
View offers

That customer enter id and system detect if customer is normal or special and each of them has offers and after of viewing detect if he order or not

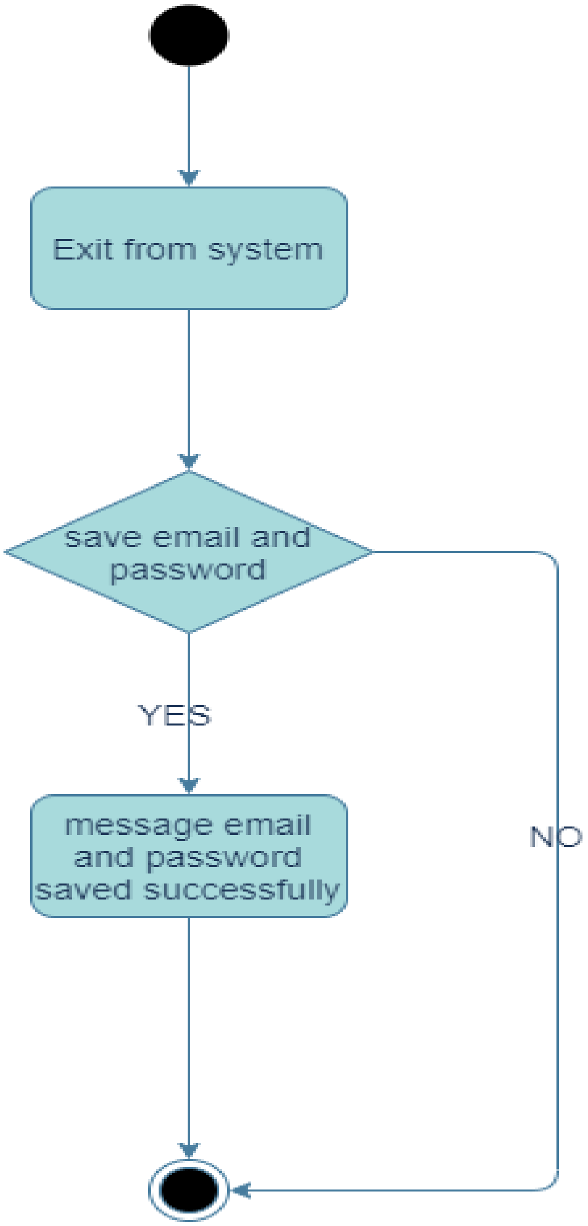


Update profile

Person enter to setting and update profile and display message that profile was updated



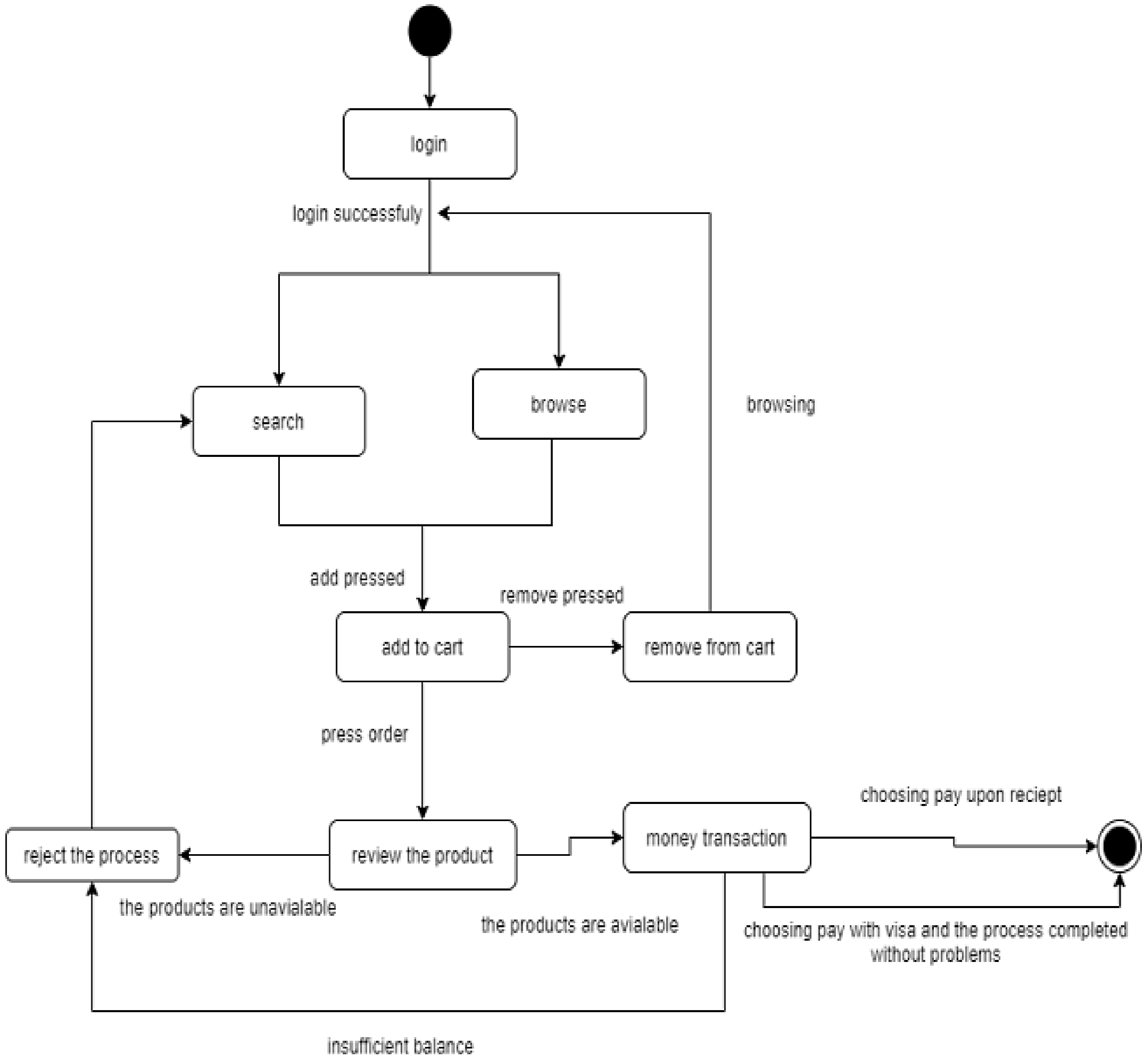
Change account password Person enter current password and enter new password twice and save changes



Logout

Person logout from the system and detect if he want save email and password or not

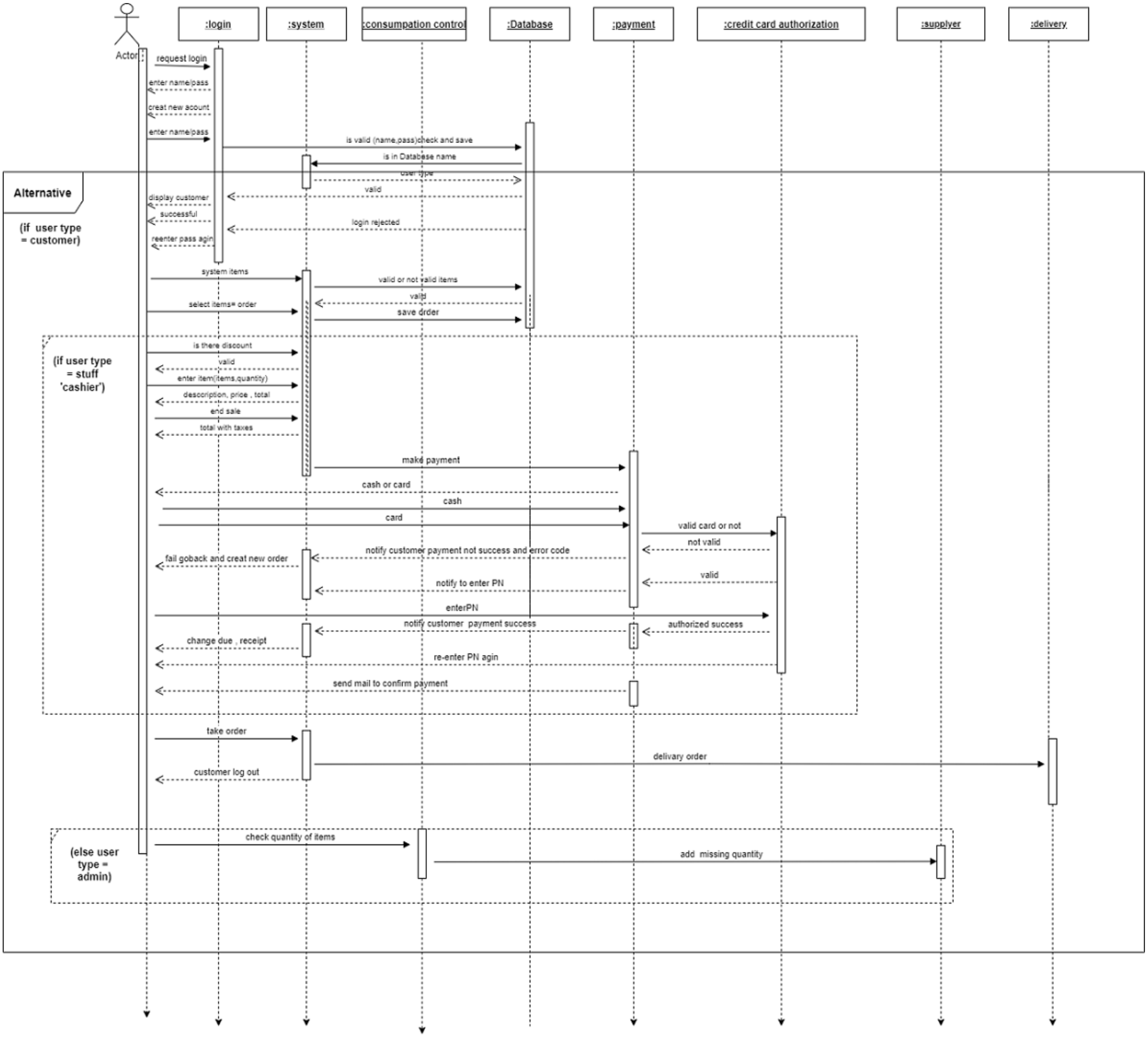
14-State diagram



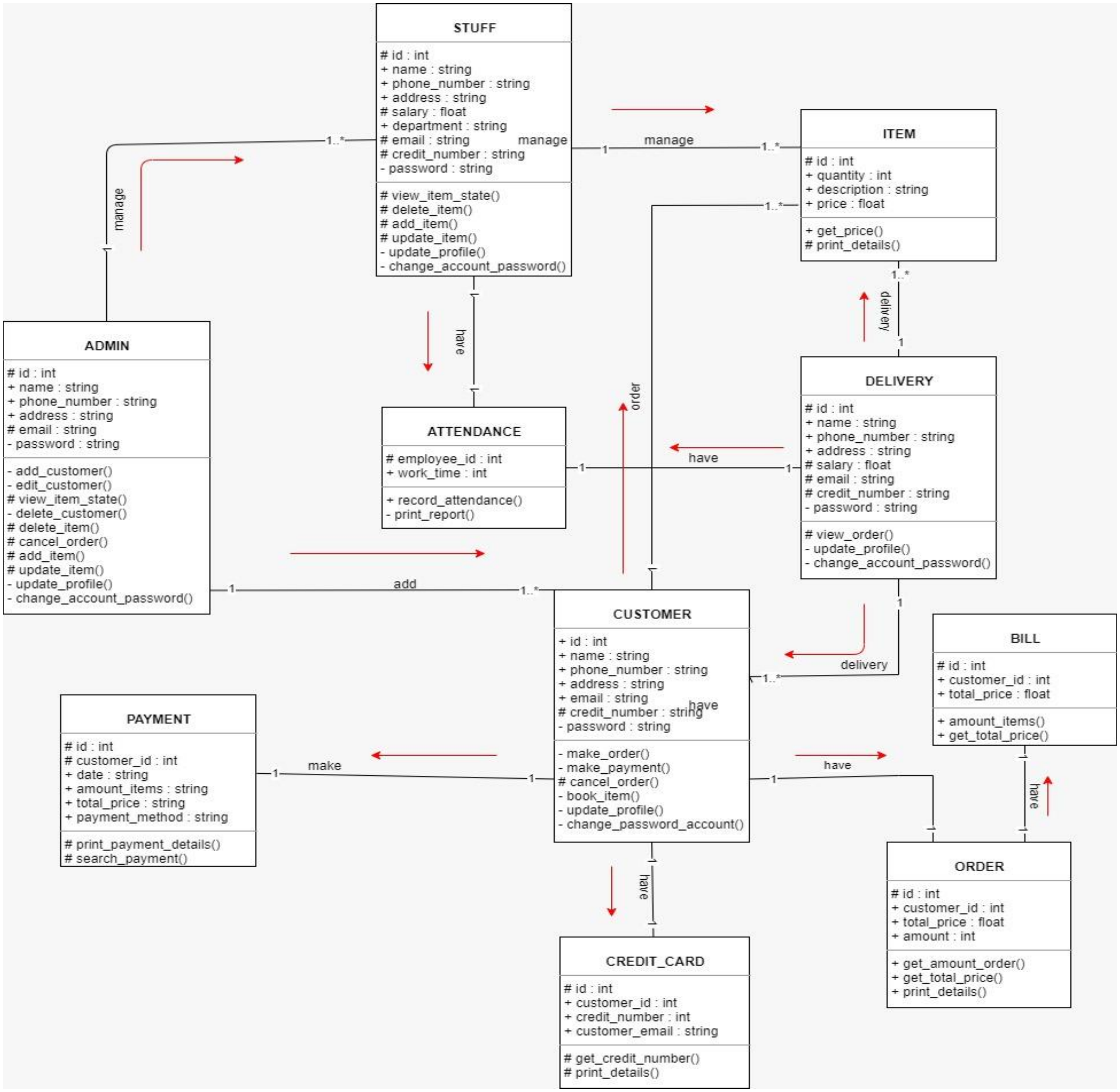
State diagram

- First the client has to login, once he login in he will be able to go the main page to browse and search for what he need.
- After founding what he need, he can to add it to the cart.
- If he made a mistake by choosing undesirable product for example, he can remove it from the cart.
- When he assure what he want to buy and ordered it the system will search and checks if the product are available at the store.
- If it’s not available the system will inform you and send you to the main page again.
- If it’s available the system will send you to pay and he will choose whether he will pay by visa or upon receipt.
- By choosing visa the system will check if there any problem with the visa such as insufficient balance or problem the number of the visa and inform you and send you to main page and remove the product from your cart.

15-Sequence diagram

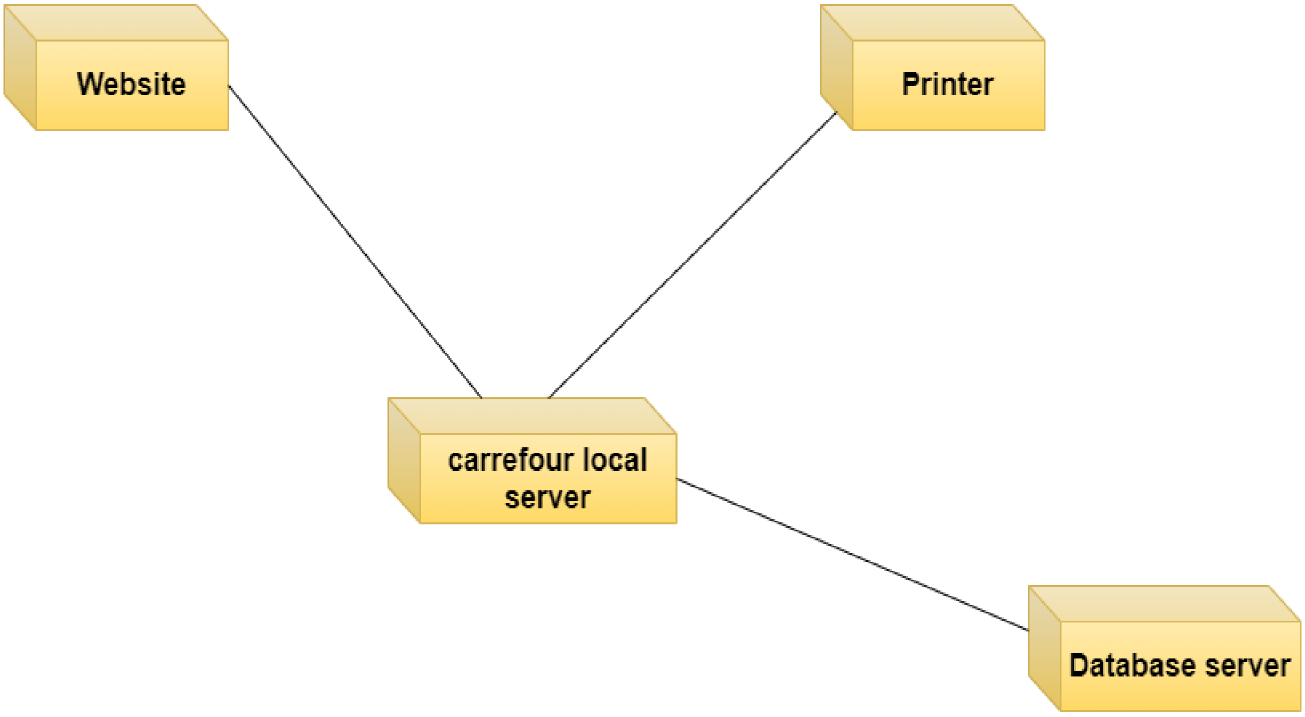


16-Class diagram



17- Deployment diagram

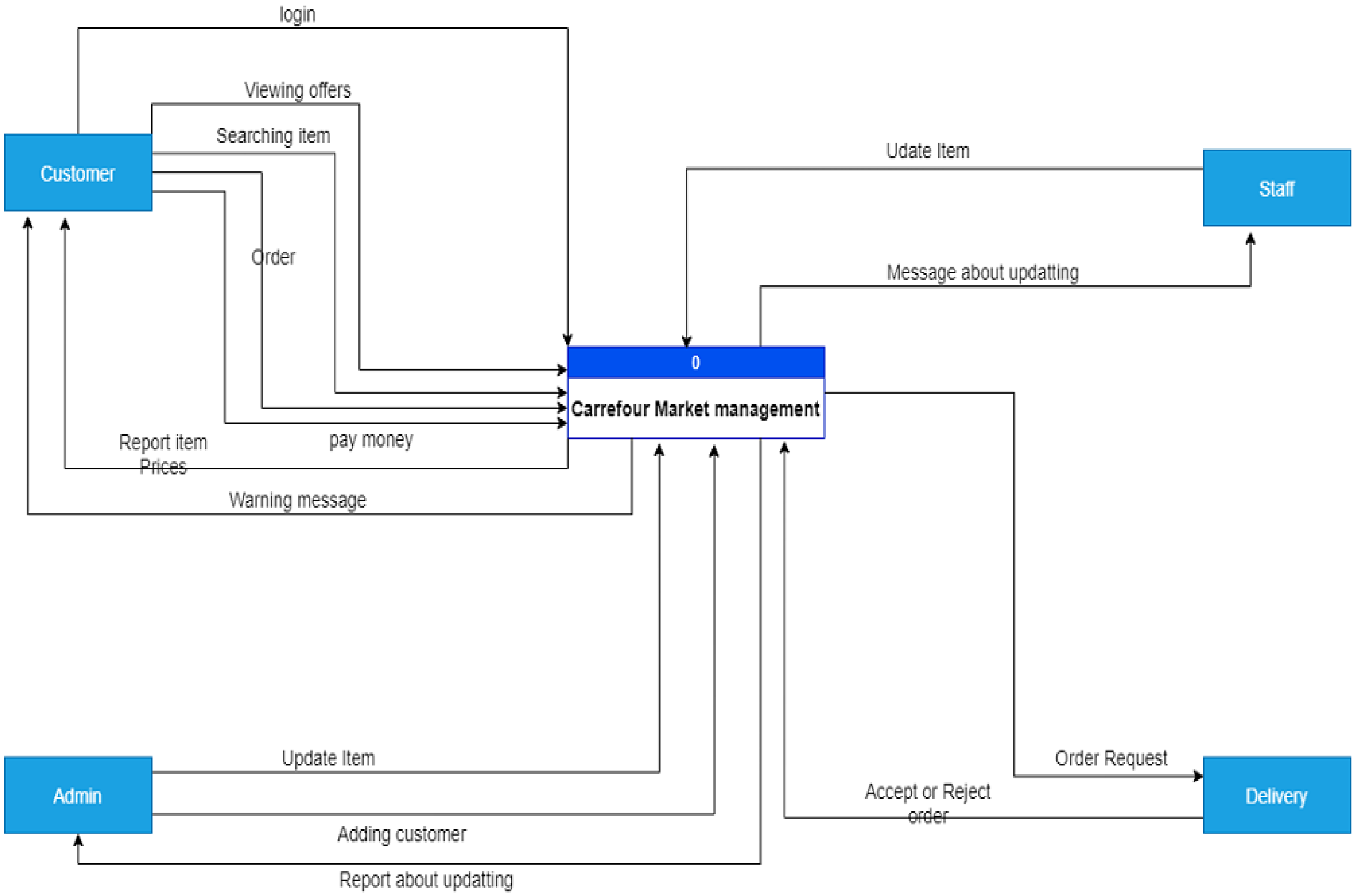
A Deployment Diagram shows the configuration of run-time processing nodes and the that live on components them. Deployment diagrams address the static deployment view of architecture. They are related to component diagrams in that a node typically encloses one or more components.



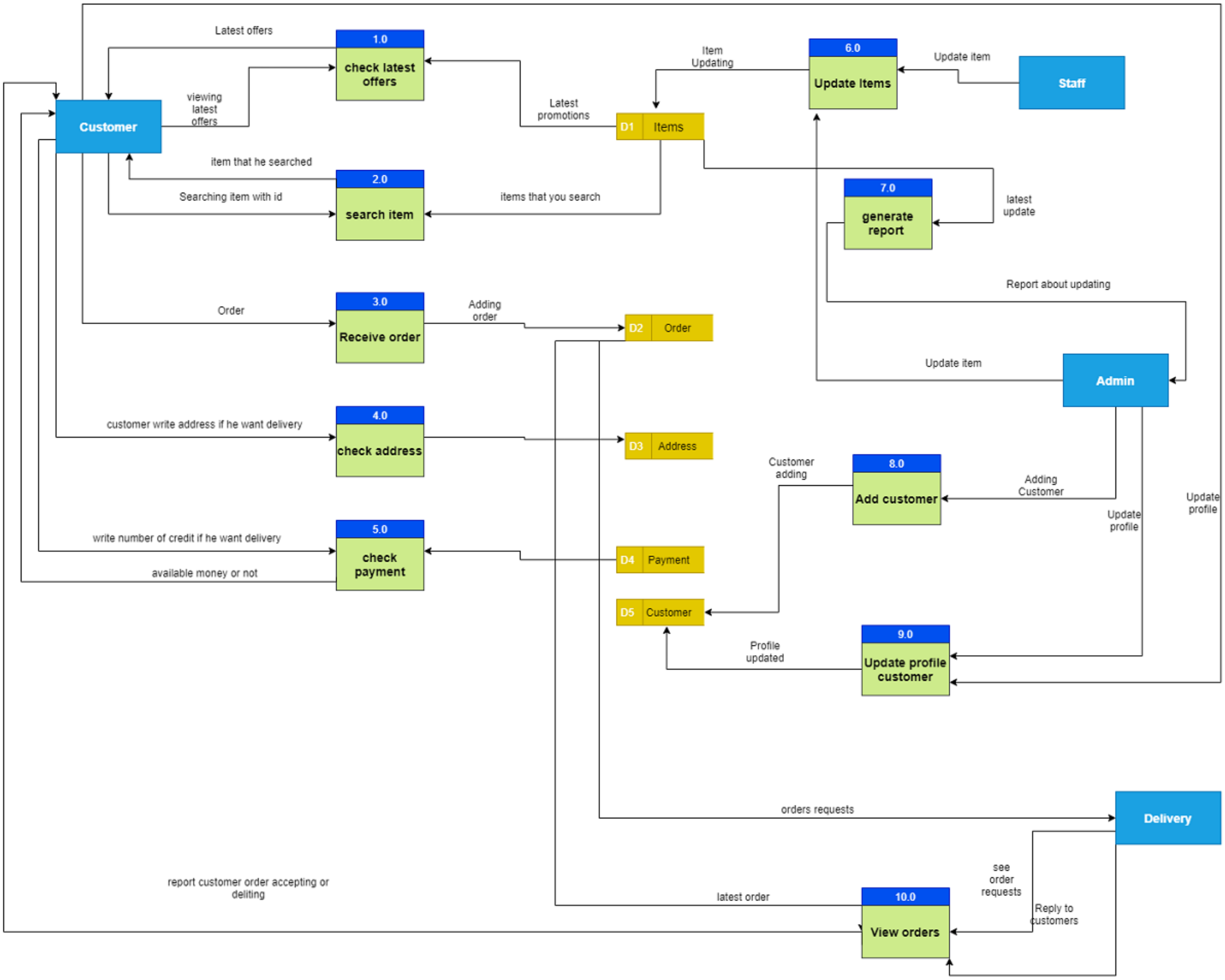
18-Data-Flow diagram(DFD)

A graphic that illustrates the movement of data between external entities and the processes and Data stores within a system.

*Context diagram

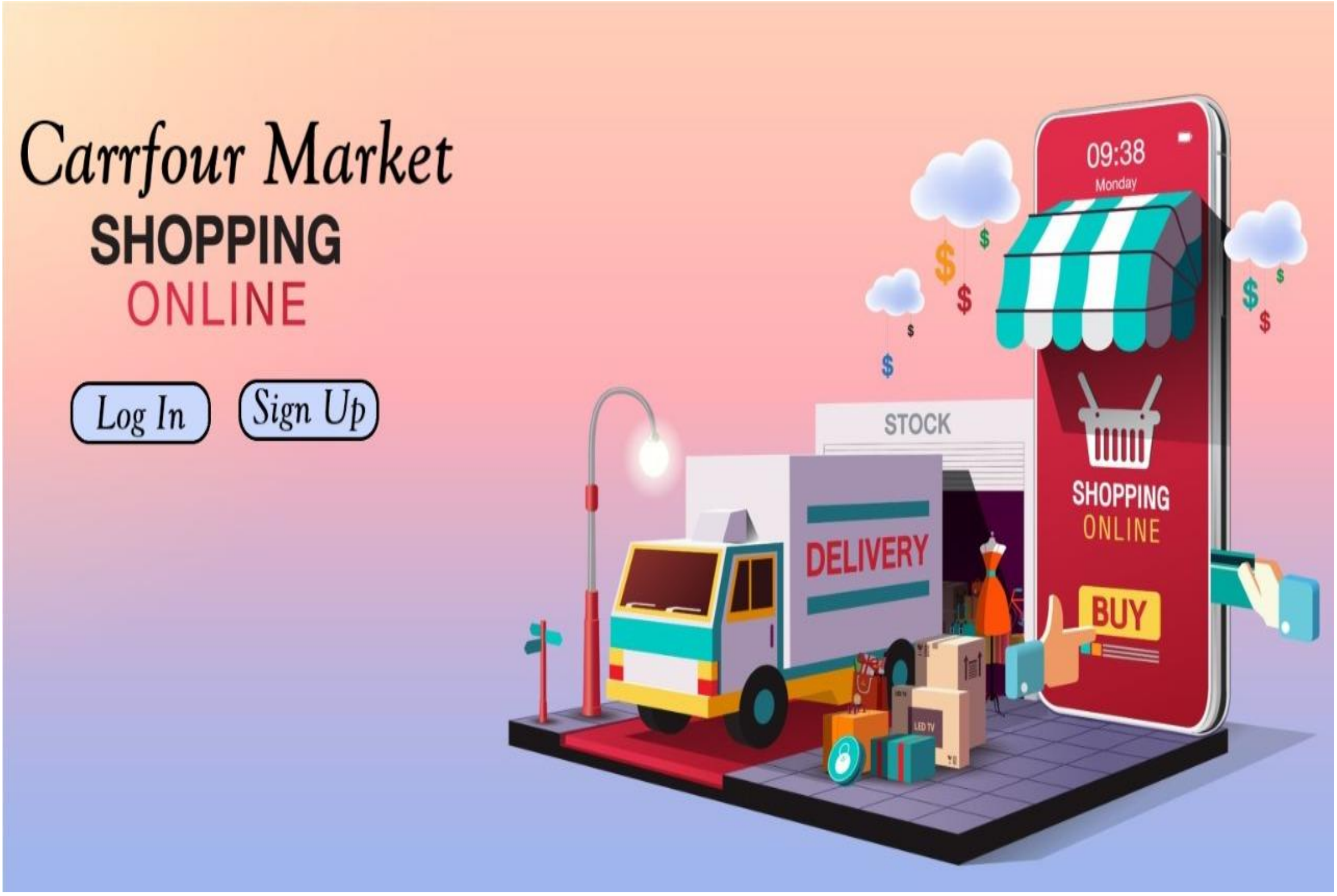


*Level 0 diagram

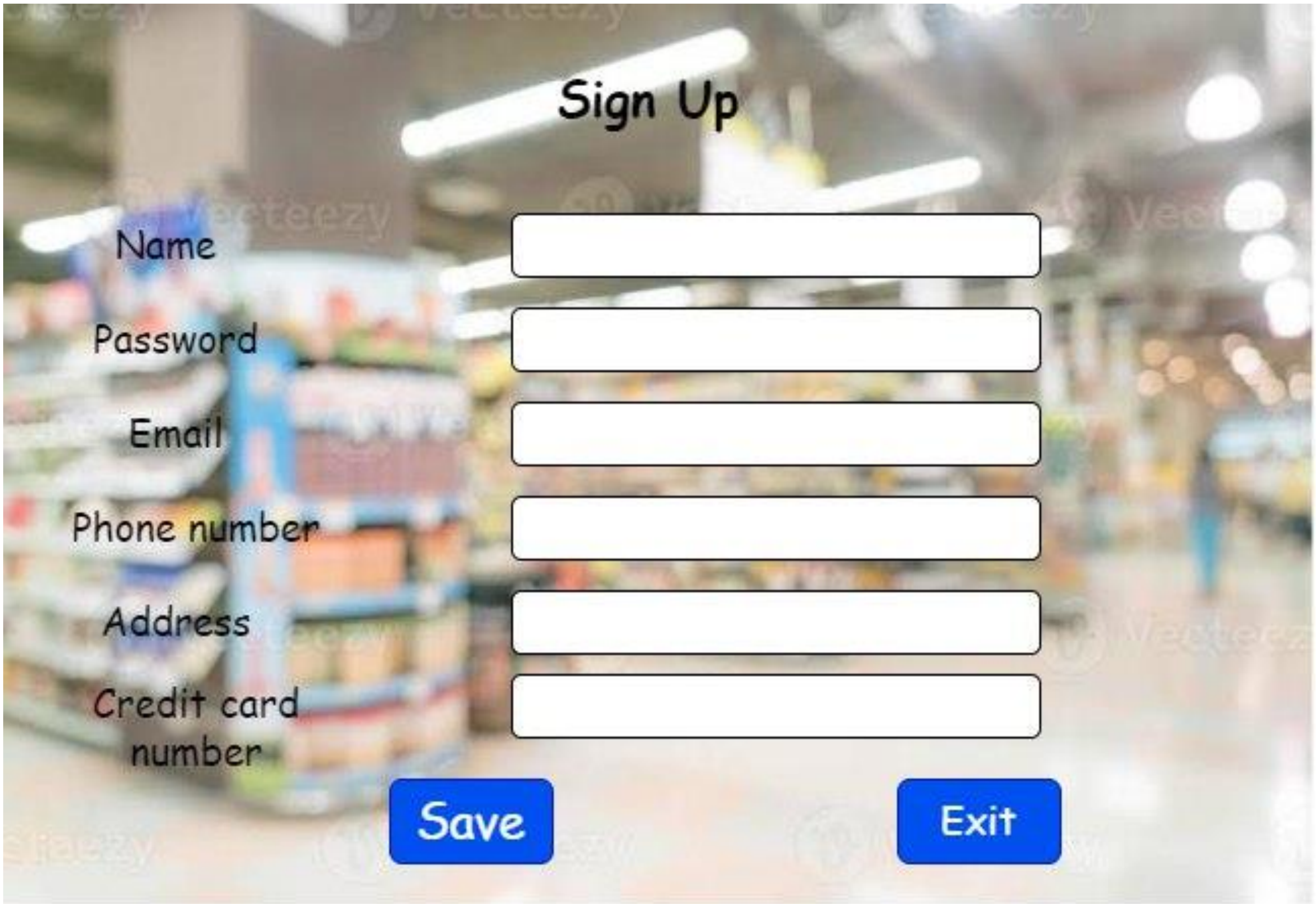


19-GUI(Graphical User Interface)

Index Page



Register Page



Sign Up

Name

Password

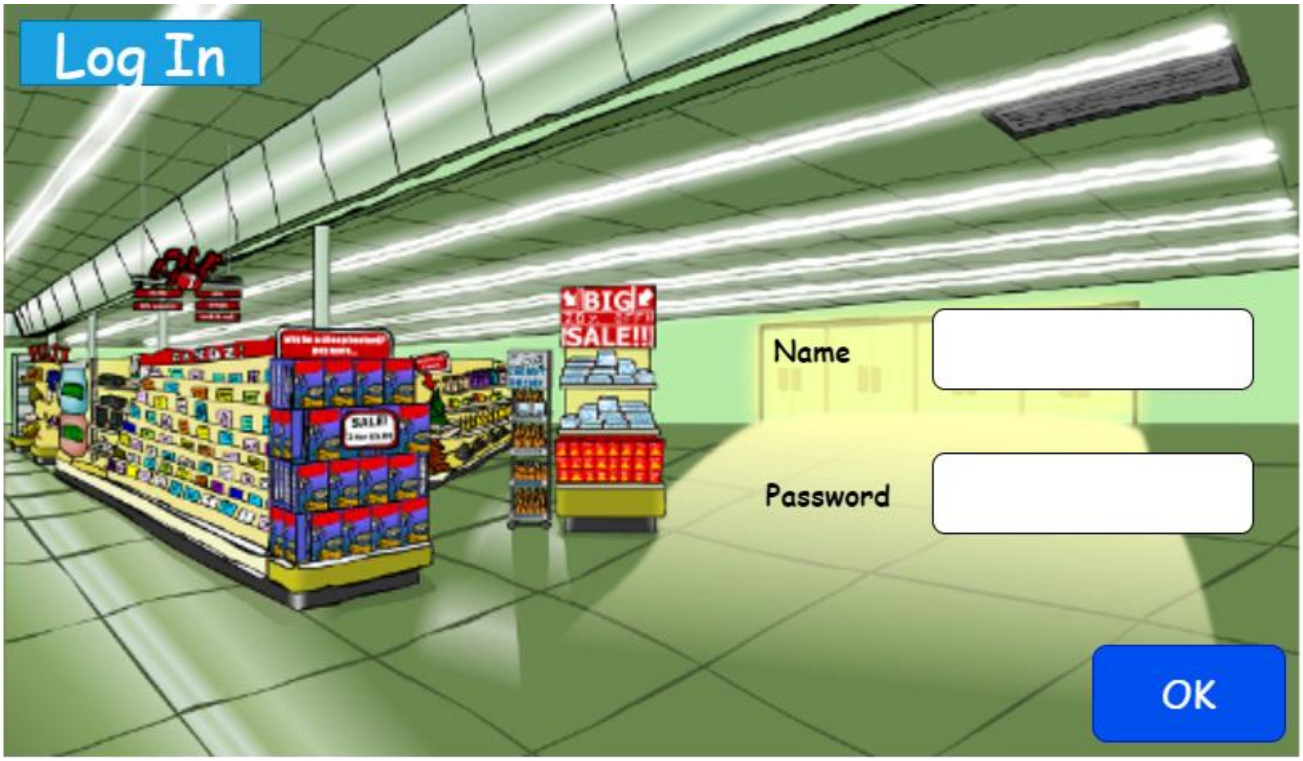
Email

Phone number

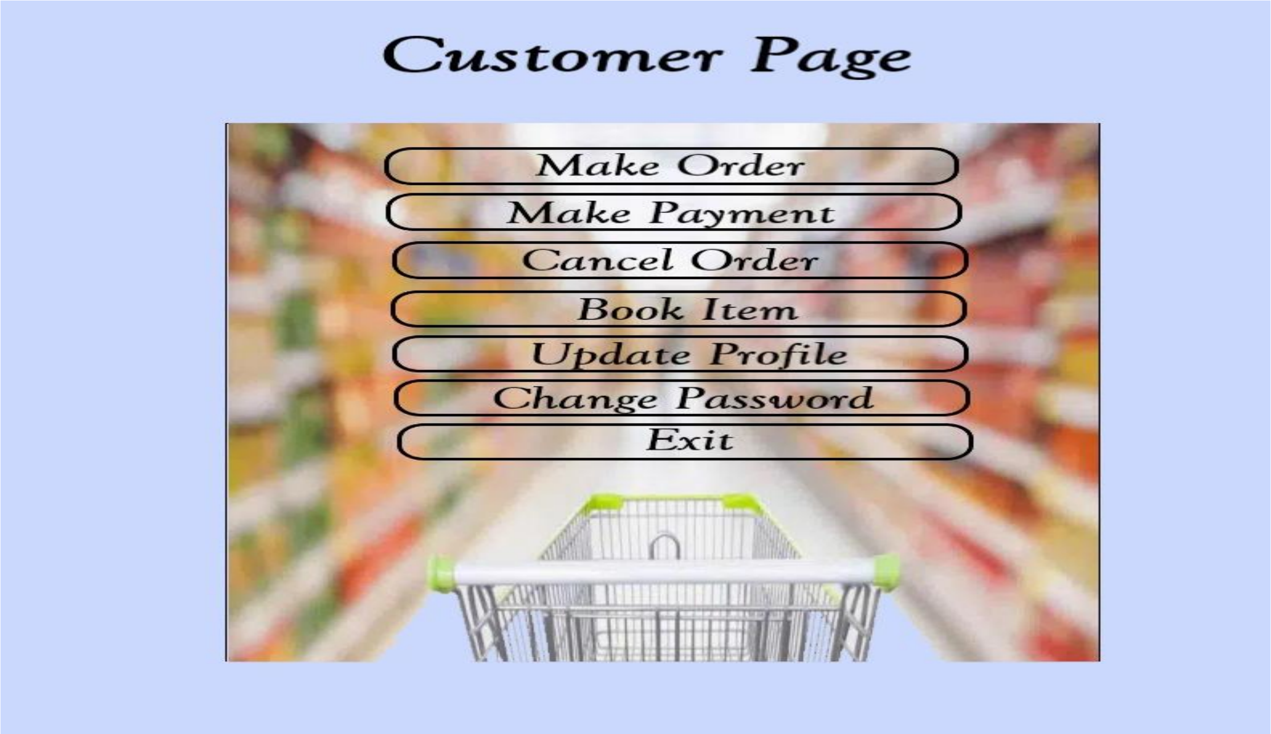
Address

Credit card number

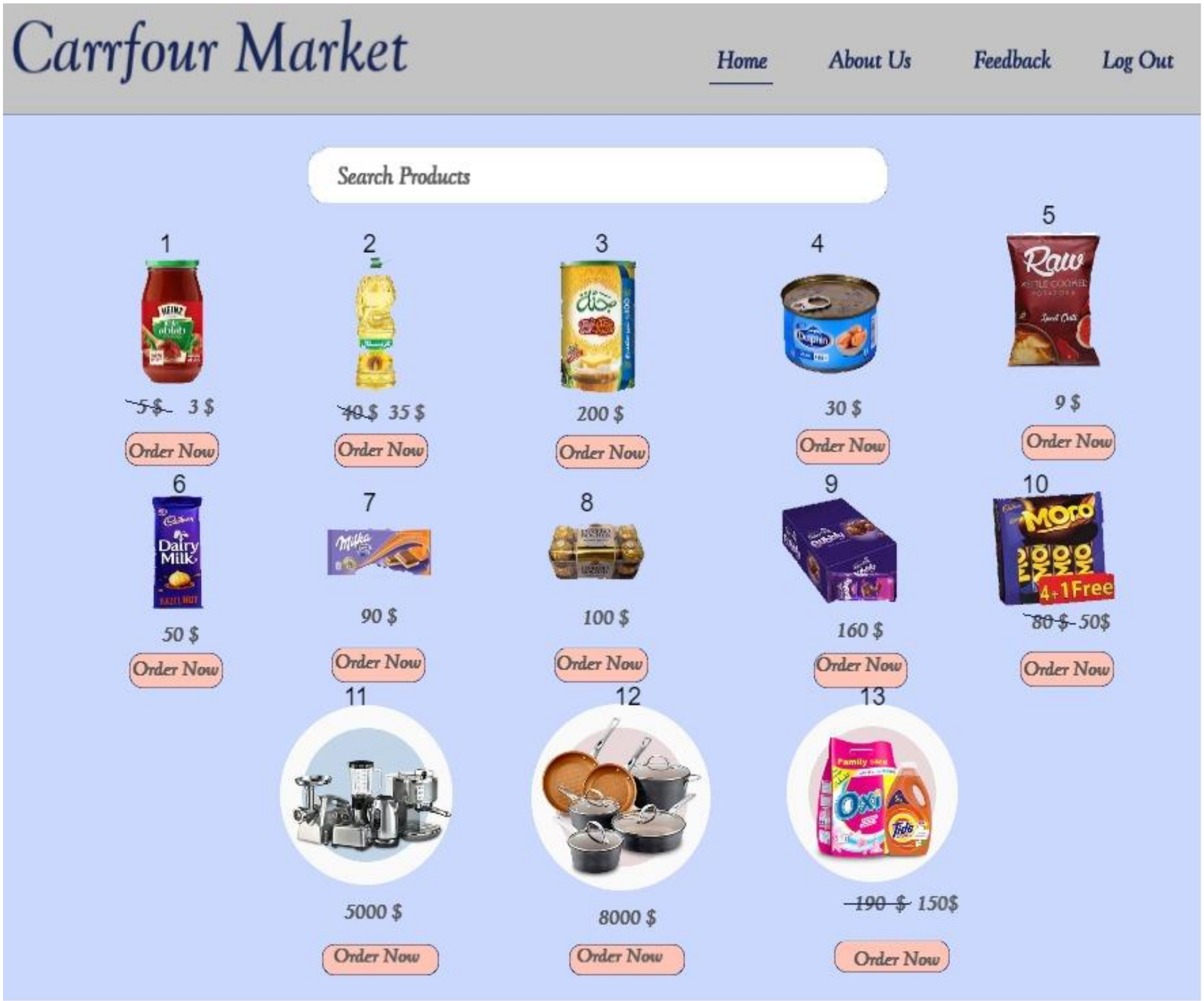
Login Page



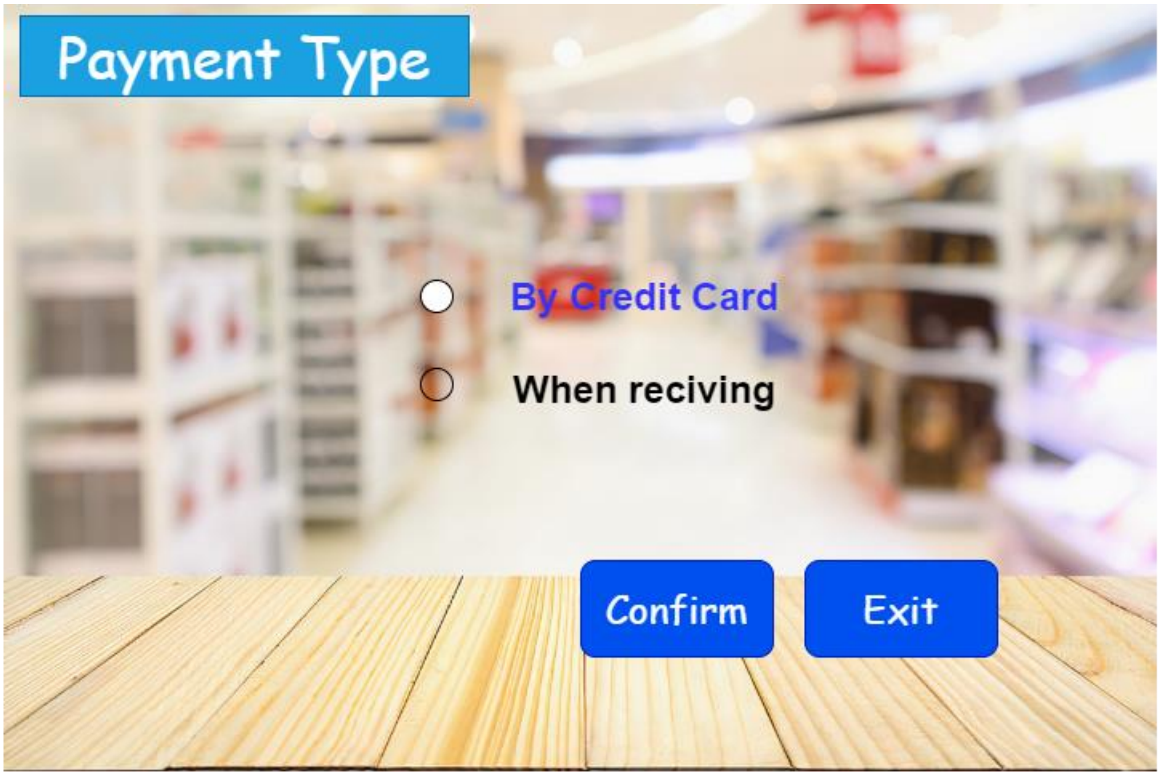
Customer Page



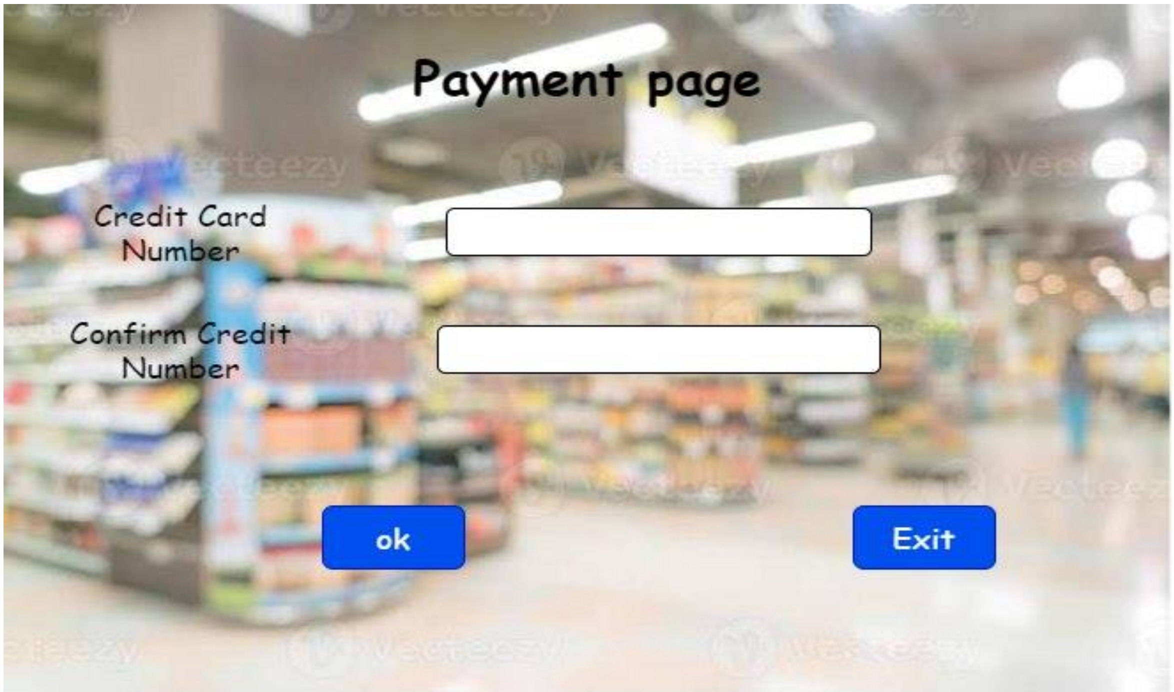
Make Order


Carrfour Market
Home
About Us
Feedback
Log Out
Search Products
1
~~5 $~~ 3 $
Order Now
2
~~40 $~~ 35 $
Order Now
3
200 $
Order Now
4
30 $
Order Now
5
Raw
9 $
Order Now
6
Dairy Milk
50 $
Order Now
7
Milka
90 $
Order Now
8
100 $
Order Now
9
160 $
Order Now
10
Moro
4+1Free
~~80 $~~ 50$
Order Now
11
5000 $
Order Now
12
8000 $
Order Now
13
Oxi
Tide
~~190 $~~ 150$
Order Now

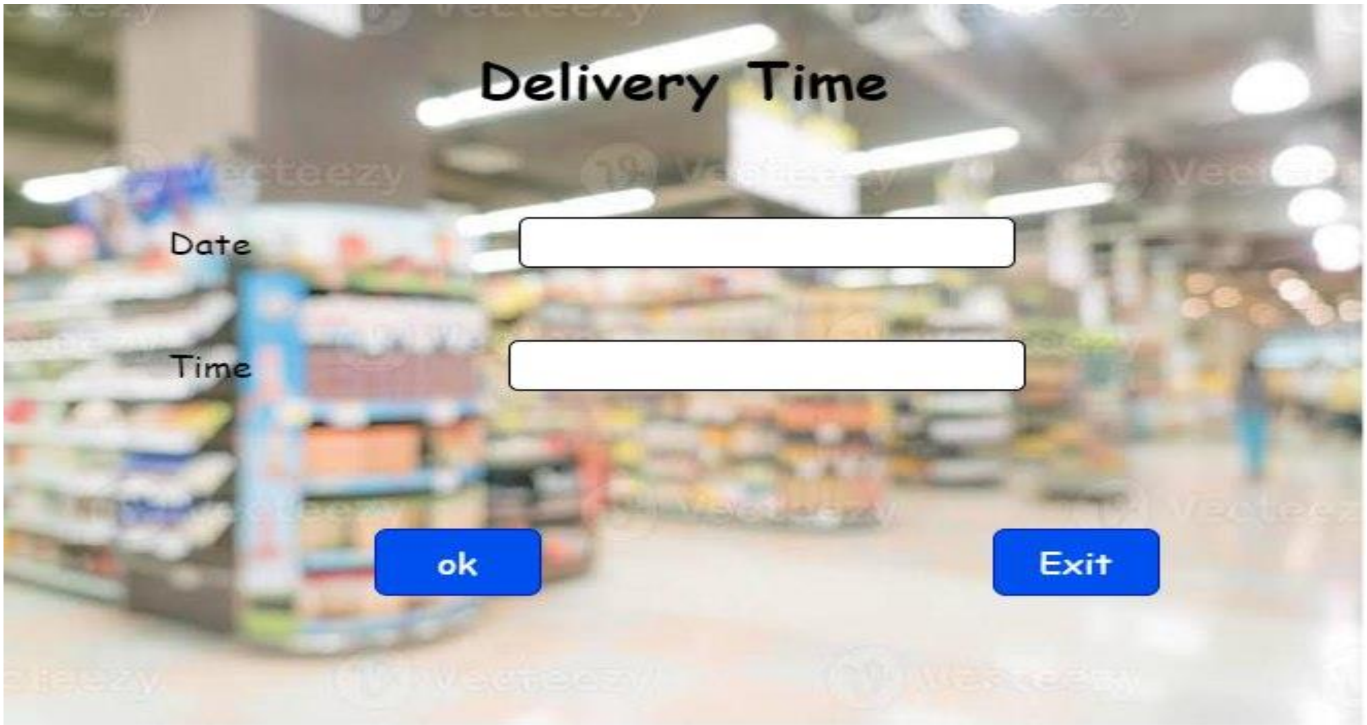
Payment Way



Payment with Credit



Delivery time

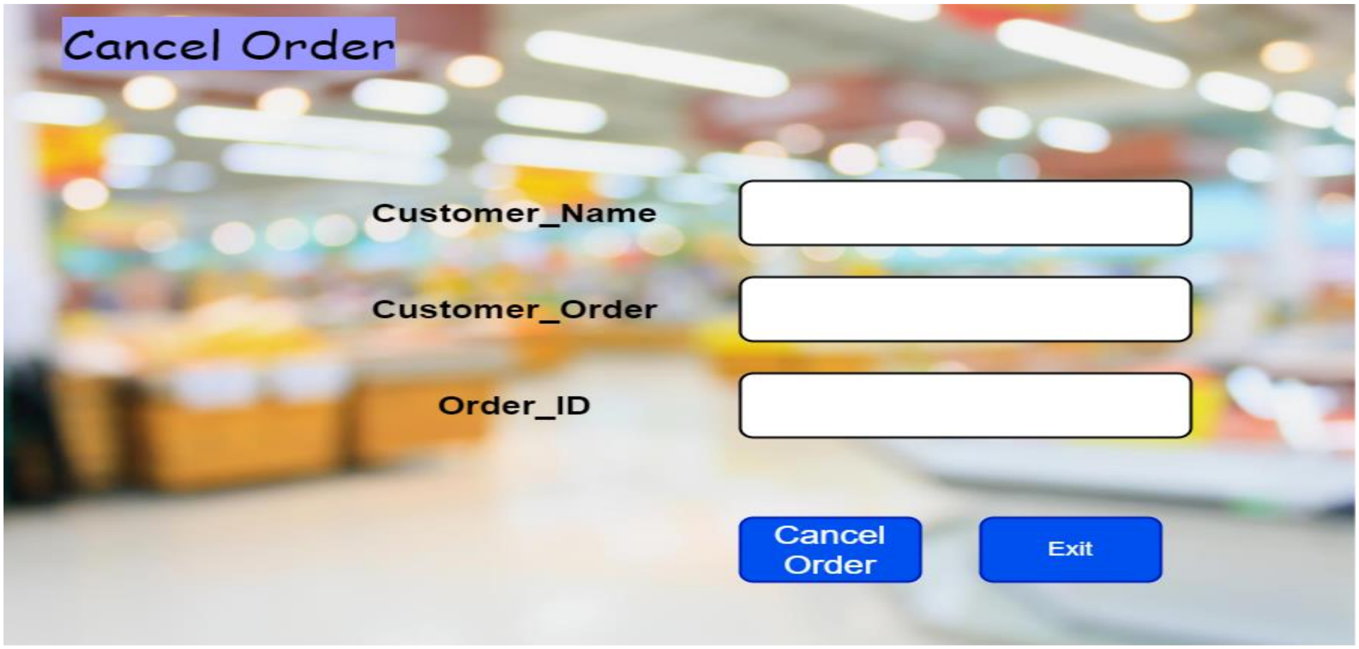


Delivery Time

Date

Time

Cancel order



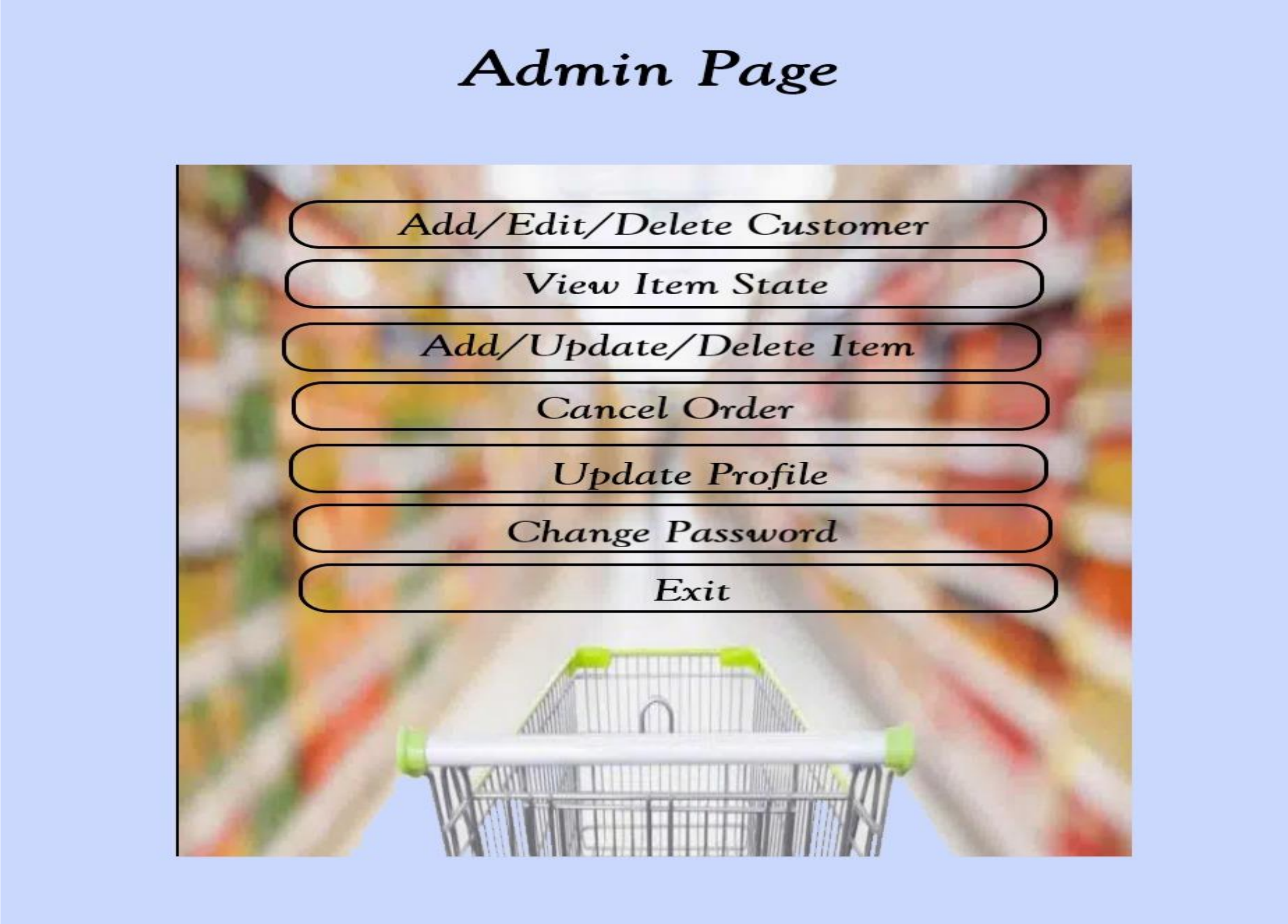
Cancel Order

Customer_Name

Customer_Order

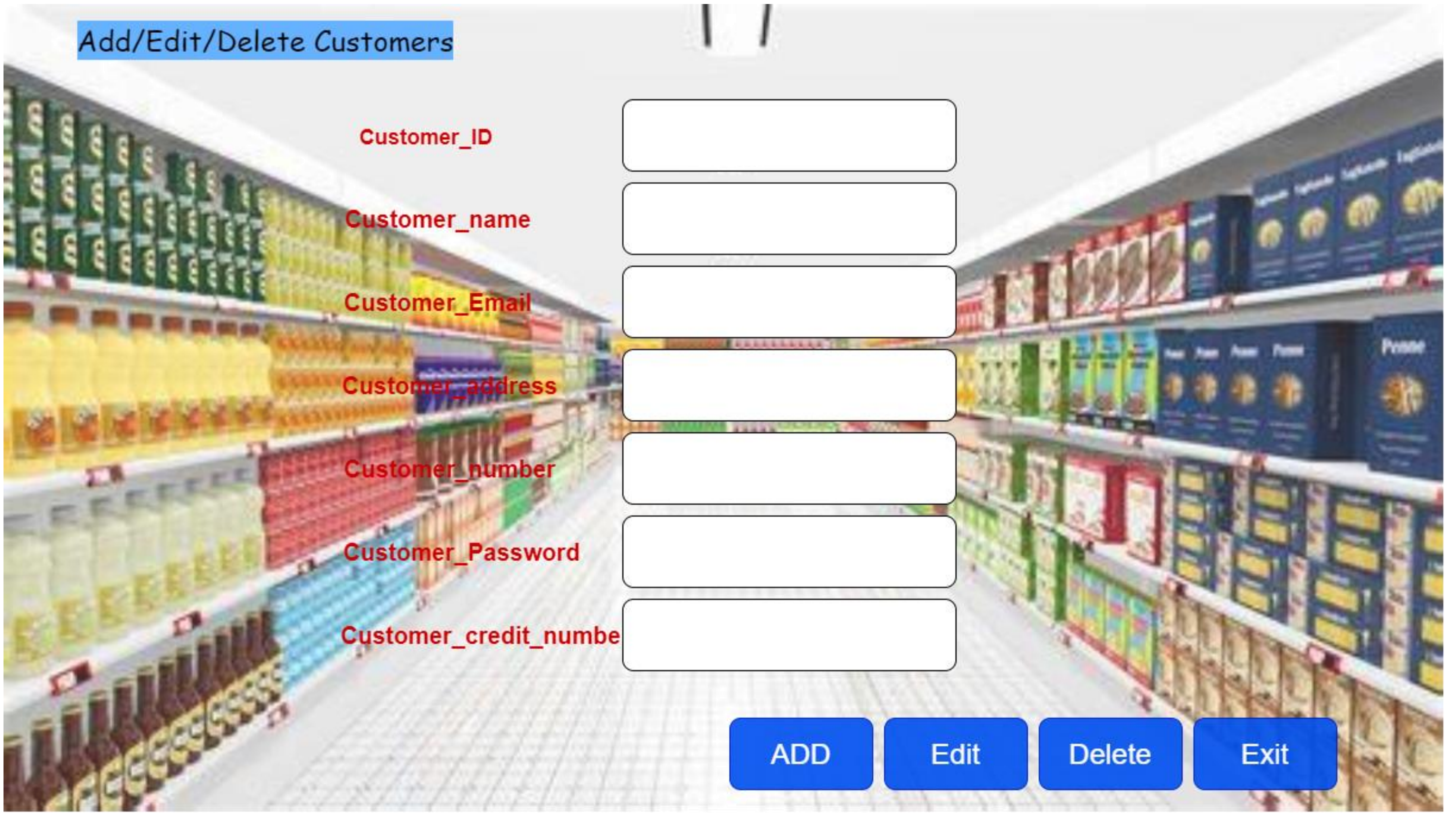
Order_ID

Admin page

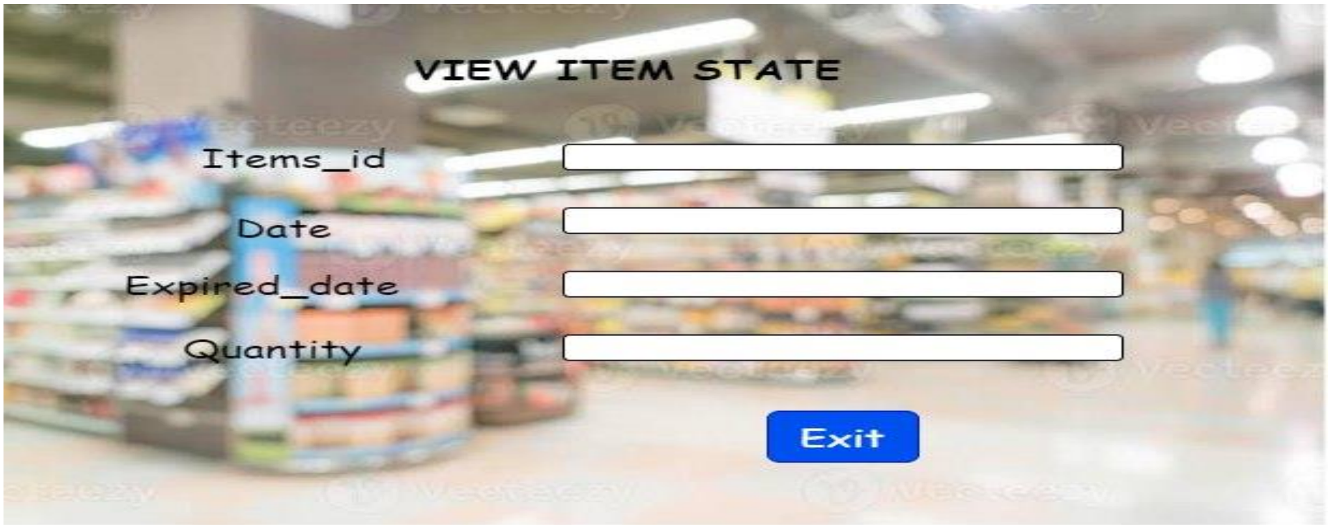


Admin Page
Add/Edit/Delete Customer
View Item State
Add/Update/Delete Item
Cancel Order
Update Profile
Change Password
Exit

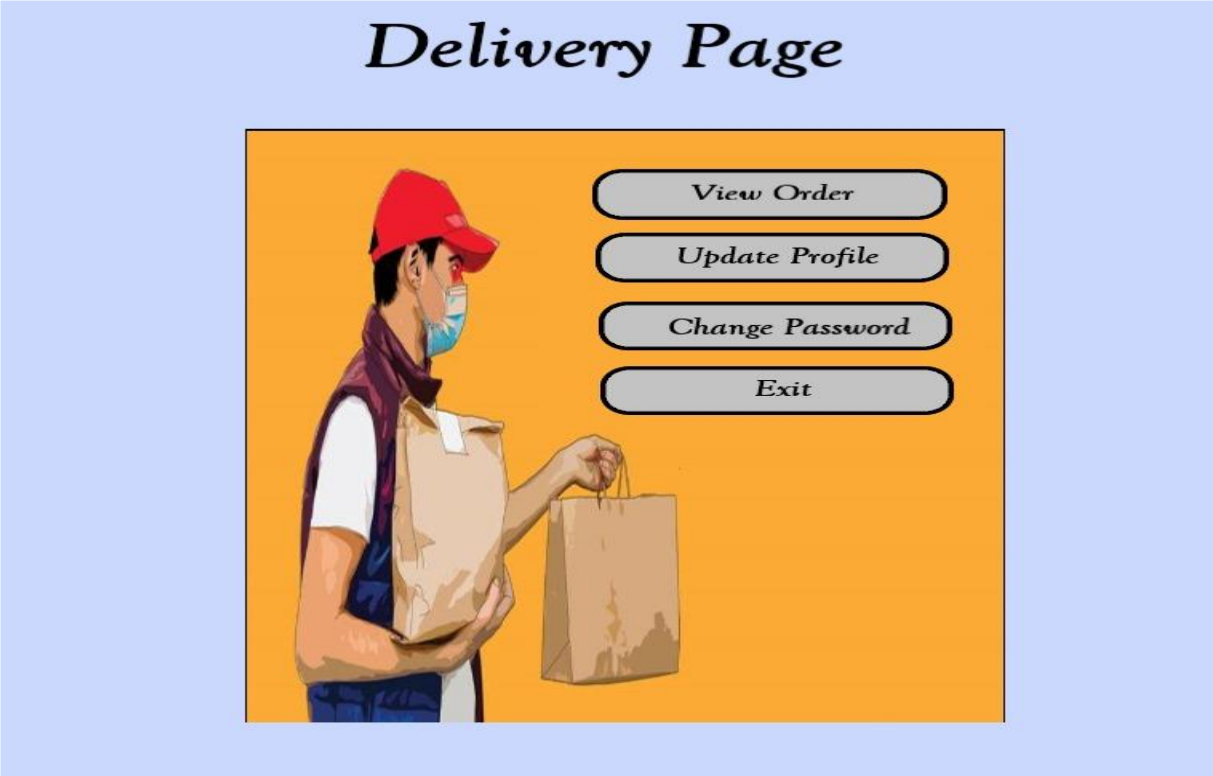
Add/Edit/Delete Customer



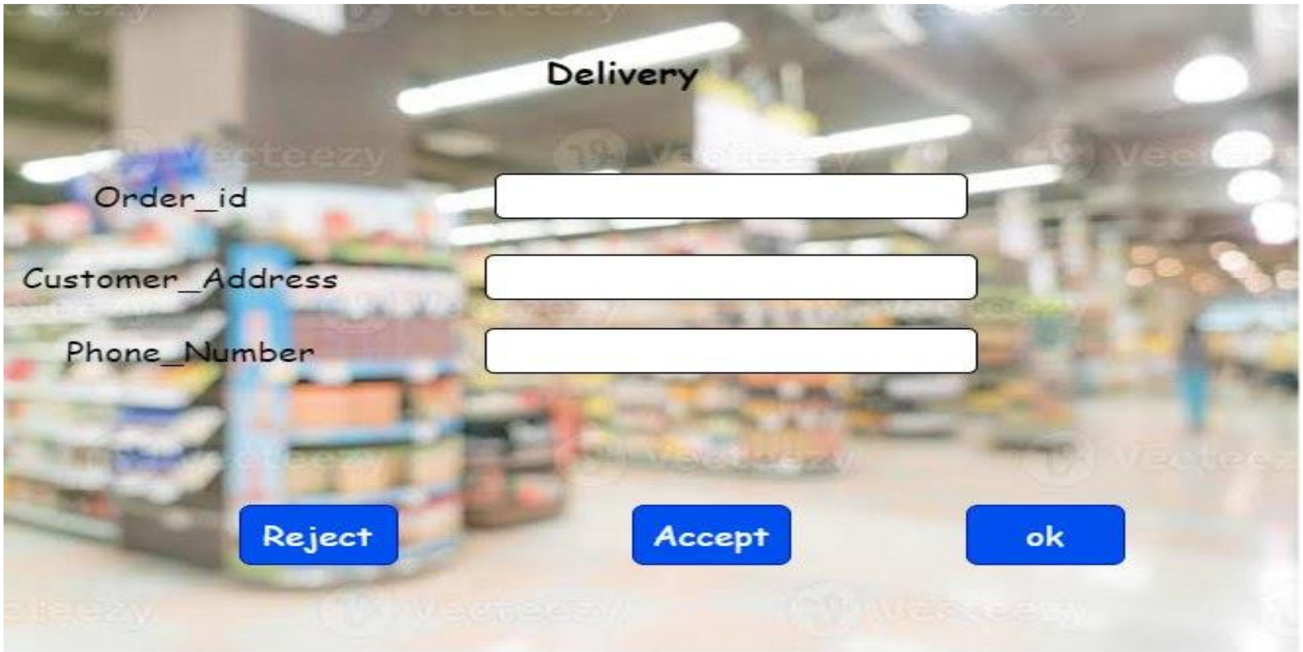
View item status



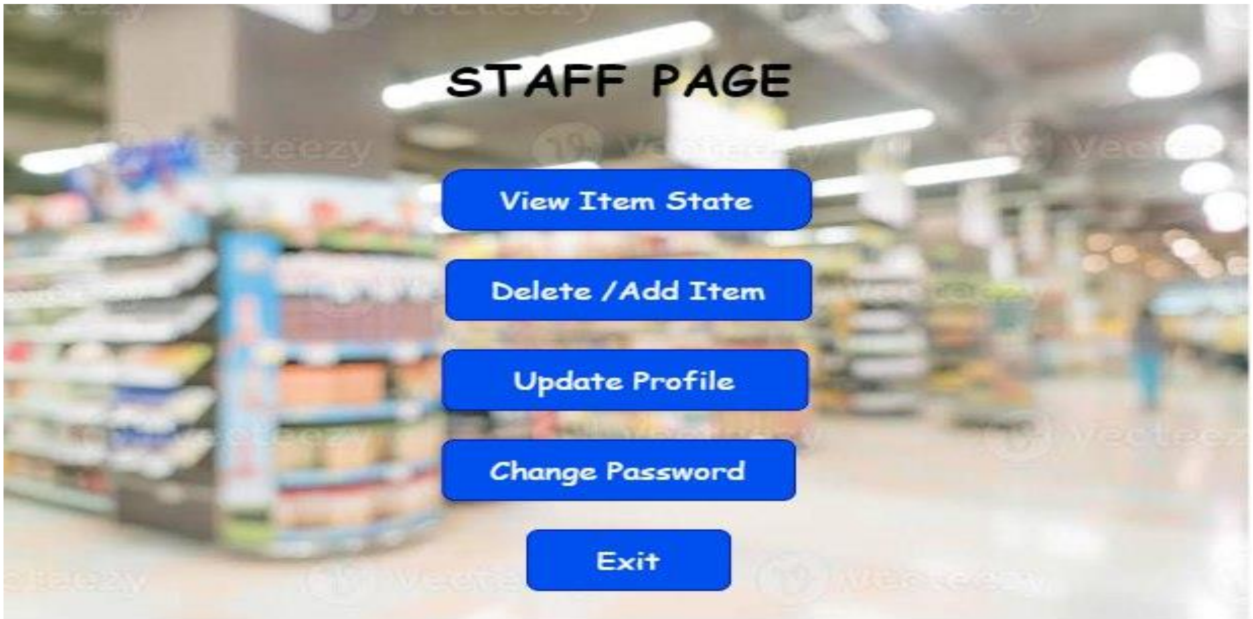
Delivery page



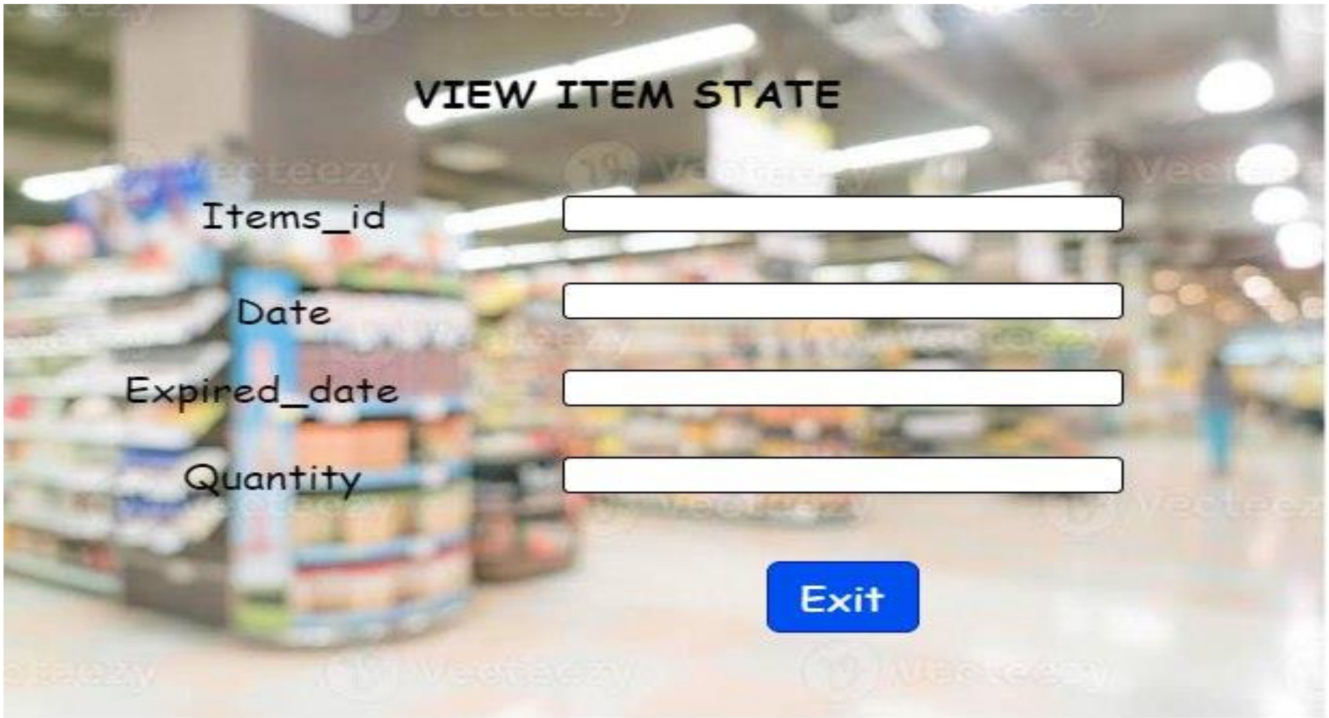
View orders



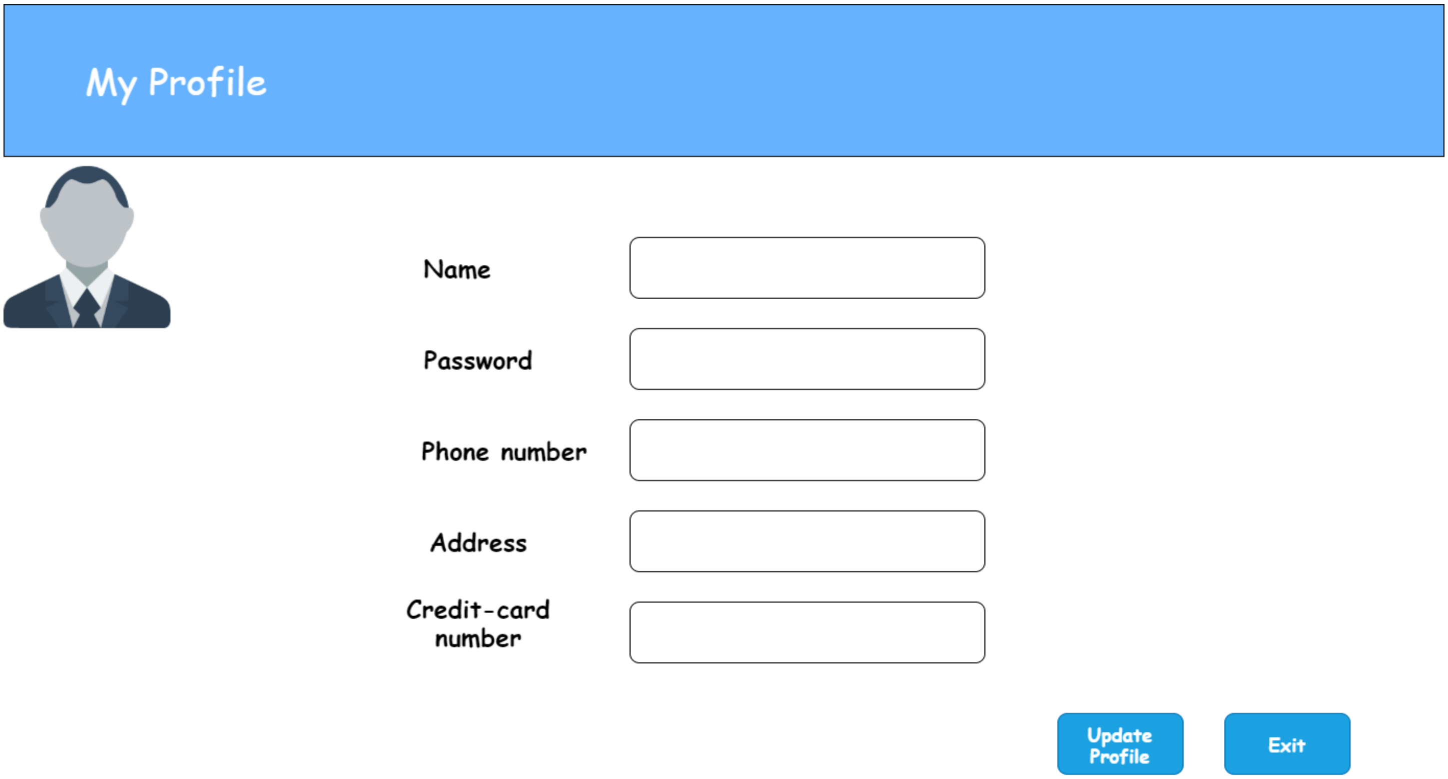
Staff Page



View item status



Update profile



My Profile

Name

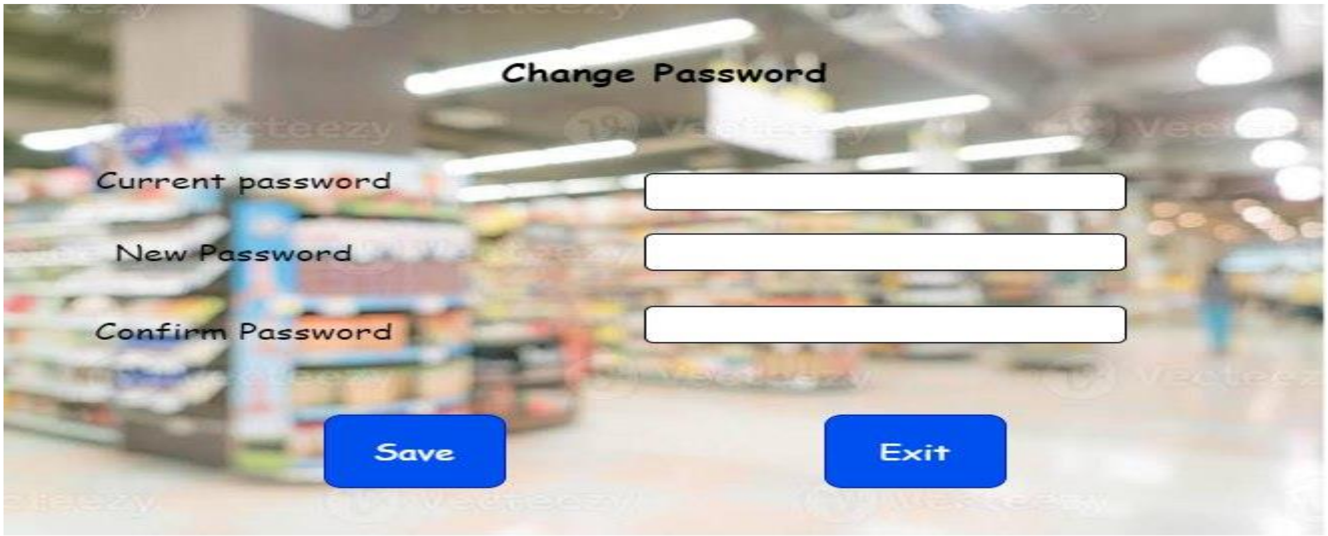
Password

Phone number

Address

Credit-card number

Change password



Change Password

Current password

New Password

Confirm Password