



Akademia Górniczo-Hutnicza

University of Science and Technology

Faculty of Computer Science, Electronics and Telecommunications

Subject: Software Engineering

FINAL PROYECT

Team members:

- Şafak Yıldız (Points 1,6,7)
- José Ignacio Valladares Ibarra (Point 2,3,6,7)
- Theodoros Kyriazis (Point 2,4,5,6)

Professor: Zabinska-Rakoczy
Dr. Tadeusz Dyduch

Date: 26/01/2017

Contenido

1. Project description	3
1.1 Area and subject of modeling	3
1.2 Modeling area	3
1.2.1 Description of organization structure	3
1.3 Activity areas' description	6
1.4. Verified system's responsibilities	6
1.5. Short problem name	6
1.6. Goals:.....	6
2. Requirements description	8
2.1 List of functions from the users' viewpoint	8
2.2 Data to be stored in the system	12
2.3 Input and output documents	12
2.4 Required special requirements and limitations	12
2.5 Functional requirements' analysis	12
2.6 Non-functional requirements	12
3. Functional analysis – DFD-s (<i>Data Flow Diagrams</i>).....	14
3.1 Context diagram	14
3.2 Top-down analysis.....	18
3.3 Processes' description.....	18
4. Working data dictionary.....	18
5. Analysis of data structures in data stores	21
6. Project of the interface	23
7. Summary	27
7.1 Assumptions for implementation	27
7.2 Verification of the Project	27
7.3Final remarks and conclusions	27
8. References.....	27

1. Project description

1.1 Area and subject of modeling

Aim of this project to show the model of a Sample Restaurant System which seems like a fast-food system such as McDonalds or Burger king. There will be multiple actors and tasks to increase the functionality of the system and to give various solutions to the system.

1.1.1 Problem domain

In our restaurant system, there are waiters who take the orders from the multiple customers and after that there will be a waiting time till the orders are prepared. So this happens for the group of customers but not the all customers. So there will be queue in the restaurant. For decreasing this number we provide a large number of staff, and enough space to wait in the Restaurant. There may also some complaints from customers about the restaurant. So we also get feedback from customers to increase the quality of the service in the Restaurant.

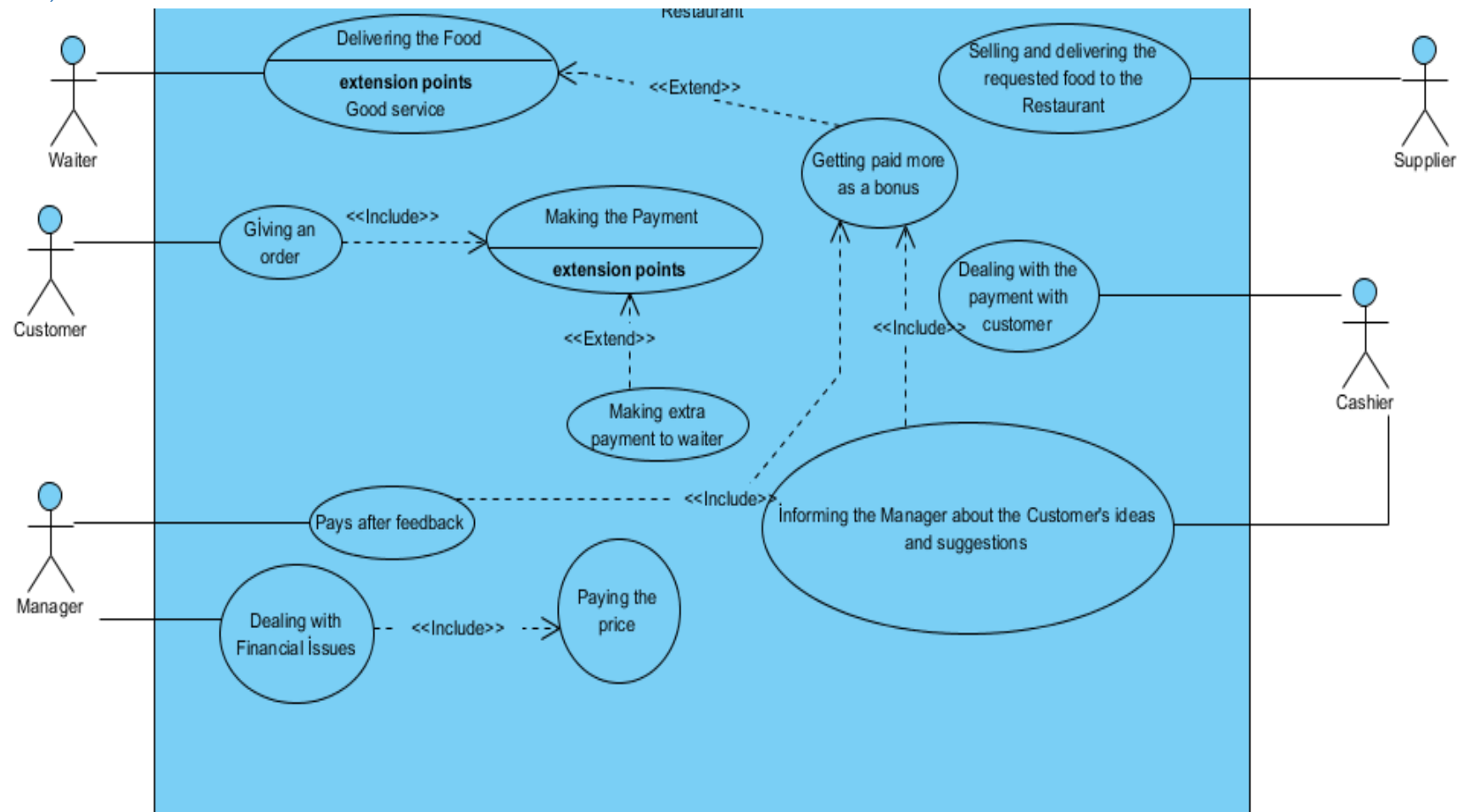
1.2 Modeling area

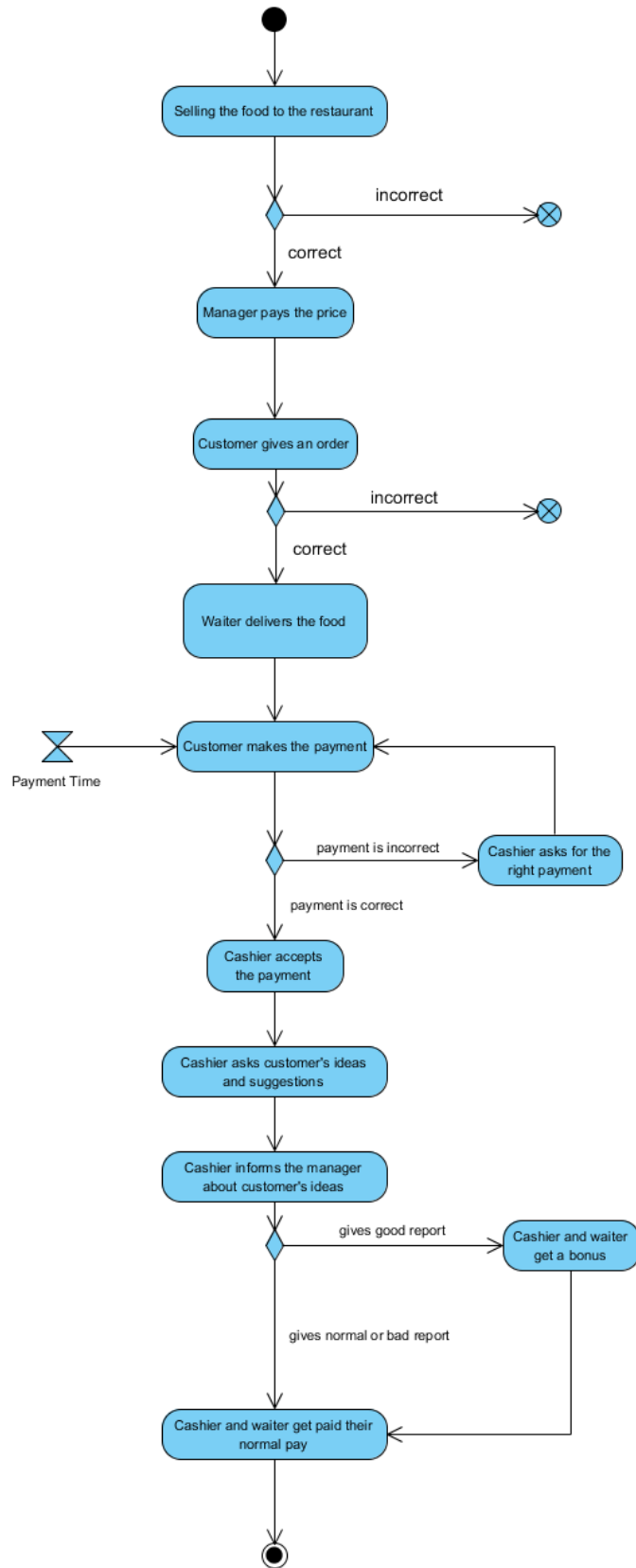
1.2.1 Description of organization structure

The customer will be able to edit his/her orders and he/she also be able to cancel the order. We also provide updated menu to increase the various tastes. After getting orders there is a waiting period for customer and this period finishes with delivery and validation of the orders. After delivery cashier gets the payment from customer and at the same time gets the feedback from customers.

After that, cashier is responsible for delivering these feedbacks to the manager by giving a report. Manager have the ability to look at the reports and edit the menu such as cancelling some orders that are taken by suppliers or adding some requested ingredients to the menu. So that the functionality and quality of the Restaurant may increase.

1.2.2 Activity areas





1.3 Activity areas' description

1.3.1 Posts description

This program will make partnership with the companies that have the same method to serve to the customers. When the customers enters to the Restaurant. Firstly they see the menu page and choose their order from this menu page. And they can also wait in the queue to give their orders and get them by hand. This is also another way of providing the orders. If they choose the way to order from website. They are taken a special ID from the website. This ID labels the customer. So that the order can be controlled. When one customer's order is announced he gets an alert which tells the order is ready. After that the customers has to go to the waiters and get their menu from there. So that the queue can be controlled and the customers don't get exhausted in the queues. From this menu, Customer may also edit the orders or cancel the orders. After choosing the menu, they are redirected to the payment menu which requests for the personal information's of the customers. We provide payment options either with cash or with credit card. Customers have to make the cash payment to the Kasa. But if they choose to pay with credit card, they can either pay from the website or they can pay to the Kasa as well.

After confirming the payment process,we request for feedback from customers.But of course this is optional.

1.3.2 Business procedures description

The business procedures will be handled between the payment company and the restaurant, similar to how finances are transfer within PayPal or a websites like Amazon

1.4. Verified system's responsibilities

This system is simply responsible for taking the orders from customers and bring it to the Restaurant staff.Restaurant staff which are waiter,cashier deals with this process.They prepare the requested menu after the payment process is verified.As it is told,we identify the customers by their ID's and their personal informations that are taken from the payment process.

1.5. Short problem name

- There's an inefficient system in the restaurant

1.6. Goals:

This product's goal is to increase the quality of service for our Customers.And also most importantly to save the Customer's time by providing them a choice from the website.Thanks to that property,they don't stand in the queues and get

exhausted. They sit and wait till they get an alert that announces their order is ready to be taken.

1.6.1 Product's aims

Our aim in designing is to give a variety of functional usage of the website to give a good service to the customers and moreover to give a good quality of service that meets the user's needs and feedbacks.

1.6.2 Design aims

Our design aims are to deliver a functional system where customers are able to order without any troubles always leaving a door open for changes.

2. Requirements description

2.1 List of functions from the users' viewpoint

Author	José Ignacio Valladares Ibarra				
Date	20 /01/2017 12:00 PM				
Brief Description	Customer enters to the Buffete restaurant and enters the webpage to order in the restaurant or before entering.				
Preconditions	<ul style="list-style-type: none">• The customer must know what he's going to order<ul style="list-style-type: none">• He must see the menu• He must be able to pay the meal				
Post-conditions					
Flow of events			Actor Input	System Output	
		1	Customer enters webpage		
		2	Customer ask order		
		3		System will generate an ID order	
		4	Client waits for system		
		5		System sends the order to Cashier	

Author	José Ignacio Valladares Ibarra				
Date	20 /01/2017 12:15 PM				
Brief Description	The system asks for payment via card or cash in the restaurant.				
Preconditions	<ul style="list-style-type: none"> The customer must have access to the page and internet The customer must be able to pay 				
Post-conditions					
Flow of events			Actor Input	System Output	
		1	Client decides method of payment and puts general info		
		2		Page will advise next step	
		3	If cash then he pays in cashier (Afterwards)		
		4	If card he inserts info		
		5		System will redirect	

Author	José Ignacio Valladares Ibarra				
Date	20 /01/2017 1:25 PM				
Brief Description	Client receives order ID.				
Preconditions	<ul style="list-style-type: none"> He must have done the previous steps 				
Post-conditions					
Flow of events			Actor Input	System Output	
		1	Customer clicks on next		
		2		System redirects to successful payment	
		3		System will give order ID	
		4		System asks for feedback	
		5	Customer gives feedback and clicks send		

Author	José Ignacio Valladares Ibarra				
Date	20 /01/2017 2:15 PM				
Brief Description	Customer receives Meal.				
Preconditions	<ul style="list-style-type: none"> He must have done the previous steps 				
Post-conditions					
Flow of events			Actor Input	System Output	
	1			System redirects to thank you page	
	2	Customer waits for meal			
	3			Waitress gets the meal and delivers	
	4	Client enjoys meal			
	5				

2.2 Data to be stored in the system

The only data stored in the system is the one that the restaurant manages such as feedback, inventory stock and financial records.

2.3 Input and output documents

	Customer View	Admin View
Inputs	<ul style="list-style-type: none">• Customers info into the webpage• Credit card information for payment• Order which the customer wants• Feedback at the end of the order	<ul style="list-style-type: none">• Updates of the menu• Redirect orders• Payment of supplies• Payment of workers
Outputs	<ul style="list-style-type: none">• Menu for the customer<ul style="list-style-type: none">• Order ID• Total amount of payment• Order (Meal ready)	<ul style="list-style-type: none">• Payment of costumer<ul style="list-style-type: none">• Supplies for stock• Records of stock• Monthly reports of gainings

2.4 Required special requirements and limitations

- System must be tested at least on a period of 1-2months.
- Customer must have access to Internet
- Customer must go into the restaurant
- System can run either by phone or computer since it's a webpage
- Credit card info shall not be stored in any database
- Interface is easy to use and quick
- There shall be a menu sotred in the webpage
- Depending on the system delays may occurr

2.5 Functional requirements' analysis

- The order to the supplier and the delivery shouldn't last for more than 1 week
- The payment transaction must happen immediately
- Menu should have all the products of the restaurant
- The customer can stay as long as he wants if he orders
- The payment by cash from the customer happens inside the store with the cashier
- If the feedback is good, the manager pays more to the employees

2.6 Non-functional requirements

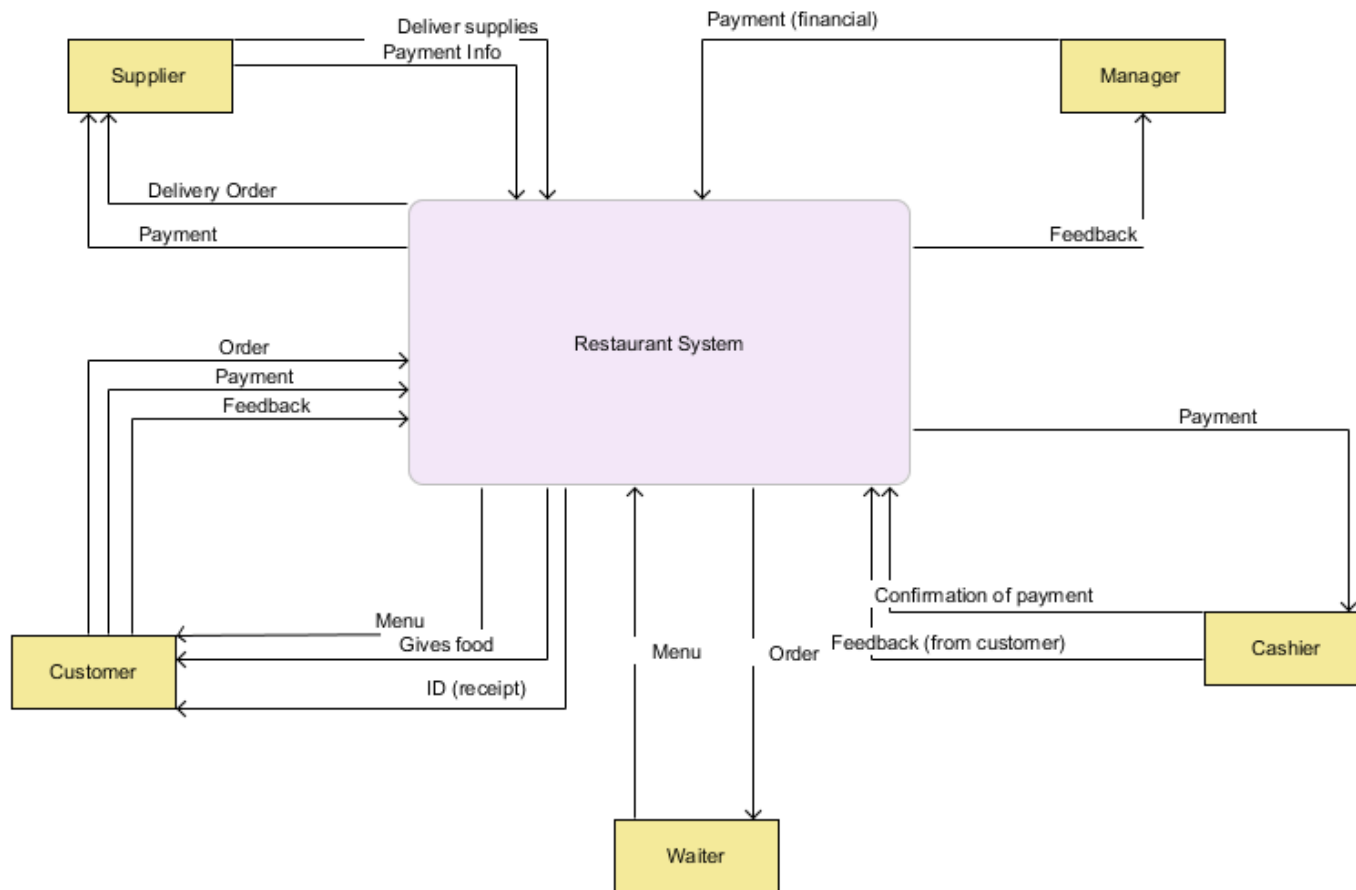
- The order to the supplier and the delivery shouldn't last for more than 1 week
- The payment transaction must happen immediately

- Menu should have all the products of the restaurant
- The customer can stay as long as he wants if he orders
- The payment by cash from the customer happens inside the store with the cashier
- If the feedback is good, the manager pays more to the employees

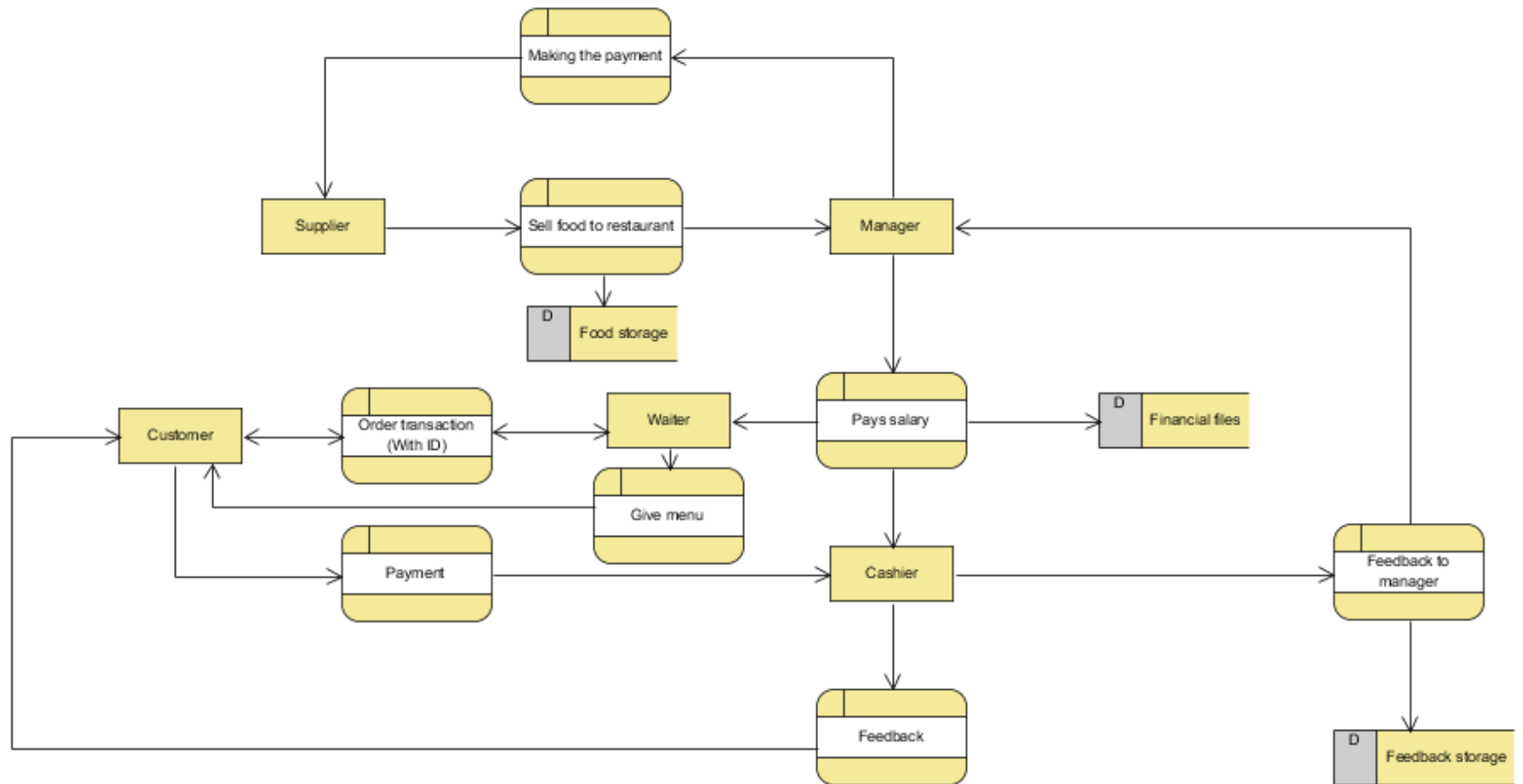
3. Functional analysis – DFD-s (*Data Flow Diagrams*)

3.1 Context diagram

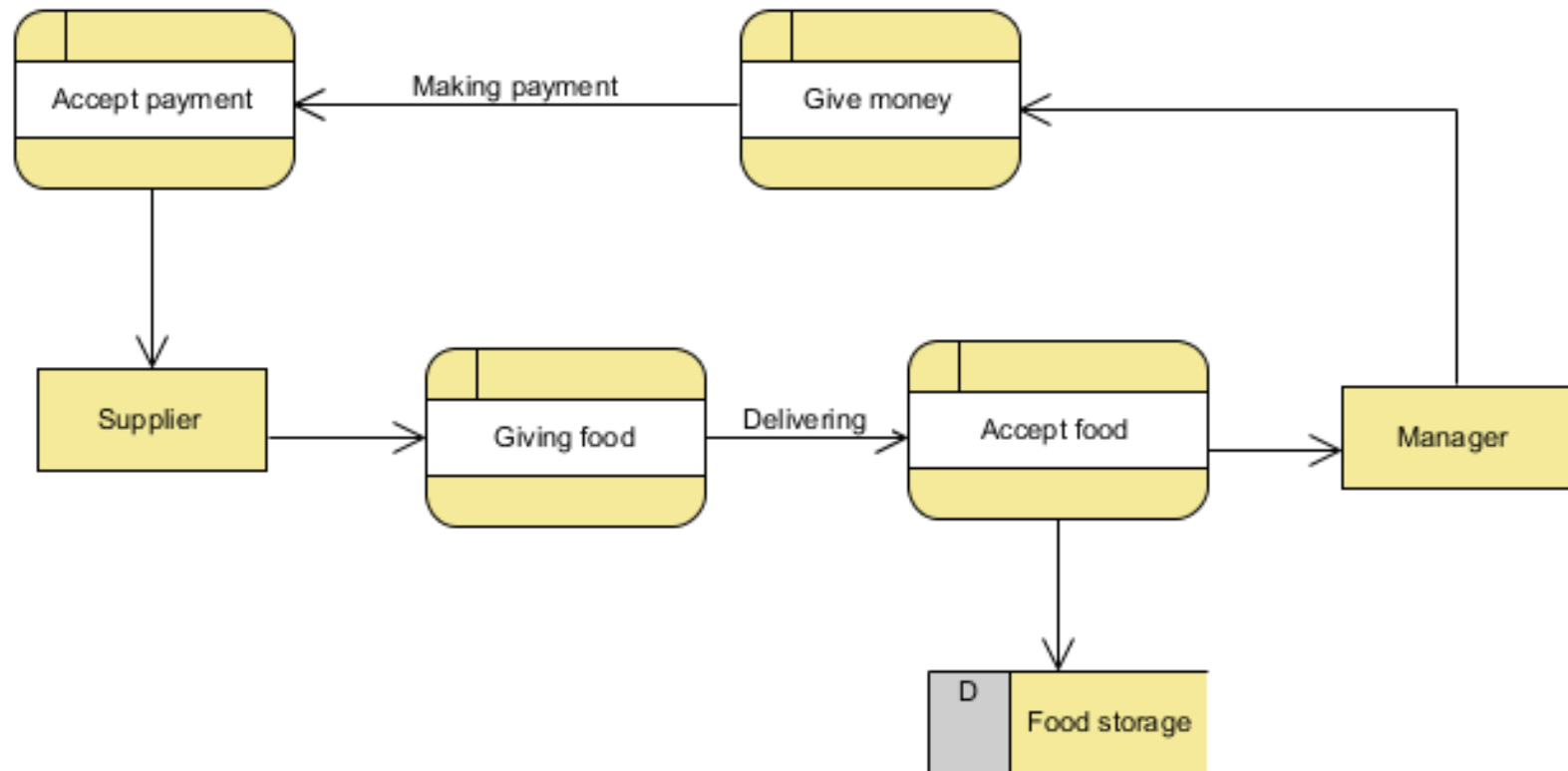
LEVEL 0 DATA FLOW DIAGRAM



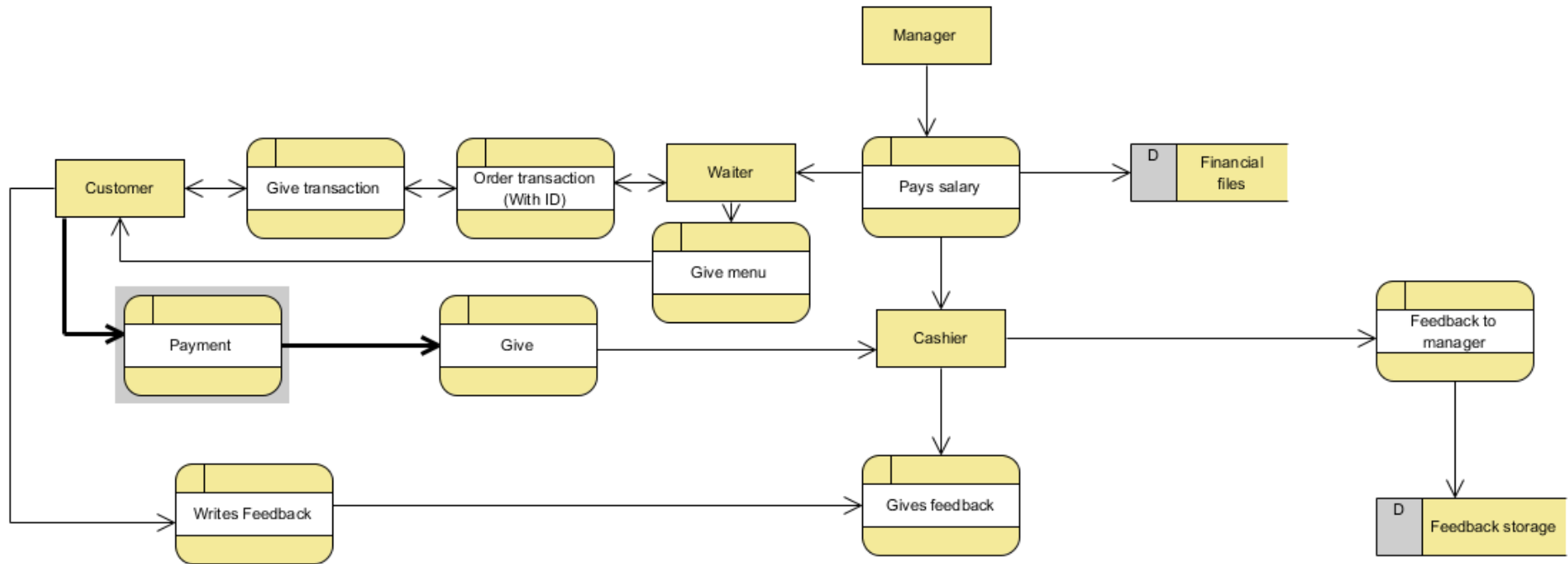
LEVEL 1 DATA FLOW DIAGRAM



LEVEL 2 DATA FLOW DIAGRAM



LEVEL 2 DATA FLOW DIAGRAM



3.2 Top-down analysis

The data flow requires at first, the supplier to get the order, deliver it, then ask for the payment by giving the payment information and at last, get paid.

The system interacts with the customer who after he gets the menu from the system, and orders. He gets the food from the system and then pays it gets the receipt and gives feedback back to it.

The waiter also interacts with the system by giving the menu and taking the order by it.

Also the cashier, gives the feedback to the system and confirms the payment.

Then he gets paid by it.

At last, the system gives the feedback to the manager and he pays the system.

3.3 Processes' description

4. Working data dictionary

SUPPLIER

Supplier entity has:

integer(10): SupplierID, Phone

varchar(255): Name, Company

SupplierID is the primary key, so it is unique for every supplier.

MANAGER

Manager entity has:

integer(10): ManagerID, Phone, Working hours

varchar(255): Name

ManagerID is the primary key, so it is unique too for every manager.

DELIVERY

Delivery entity has:

integer(10): DeliveryID, Date, Price, SupplierSupplierID, ManagerManagerID

varchar(255): Size

DeliveryID is the primary key, so it is unique for every delivery. Also, DELIVERY has two foreign keys. One from SUPPLIER and one from MANAGER since it is connected with both of them. They are both unique.

CUSTOMER

Customer entity has:

integer(10): CustomerID, Phone, FeedbackFeedbackID

varchar(255): Name, Address, Feedback

CustomerID is the primary key, so it is unique for every customer. Also, CUSTOMER has one foreign key. It comes from Feedback and it is unique.

ORDER

Order entity has:

integer(10): OrderID, WaiterWaiterID, CustomerCustomerID

varchar(255): Name, Food Type, Drink Type, Price

OrderID is the primary key, so it is unique for every order. ORDER has also two foreign keys. One from WAITER and one from CUSTOMER.

WAITER

Waiter entity has:

integer(10): WaiterID, Tables Number

varchar(255): Name

WaiterID is the primary key.

FEEDBACK

Feedback entity has:

integer(10): FeedbackID, CashierCashierID, ManagerManagerID

varchar(255): Type

FeedbackID is the unique primary key. FEEDBACK has two foreign keys from CASHIER and MANAGER.

PAYMENT

Payment entity has:

integer(10): PaymentID, Size,

varchar(255): Name, Food Type, Drink Type, Price, CashierCashierID,

CustomerCustomerID

PaymentID is the primary key and unique for every payment. PAYMENT has two foreign keys. One from CASHIER and one from CUSTOMER.

CASHIER

Cashier entity has:

integer(10): CashierID, Working hours

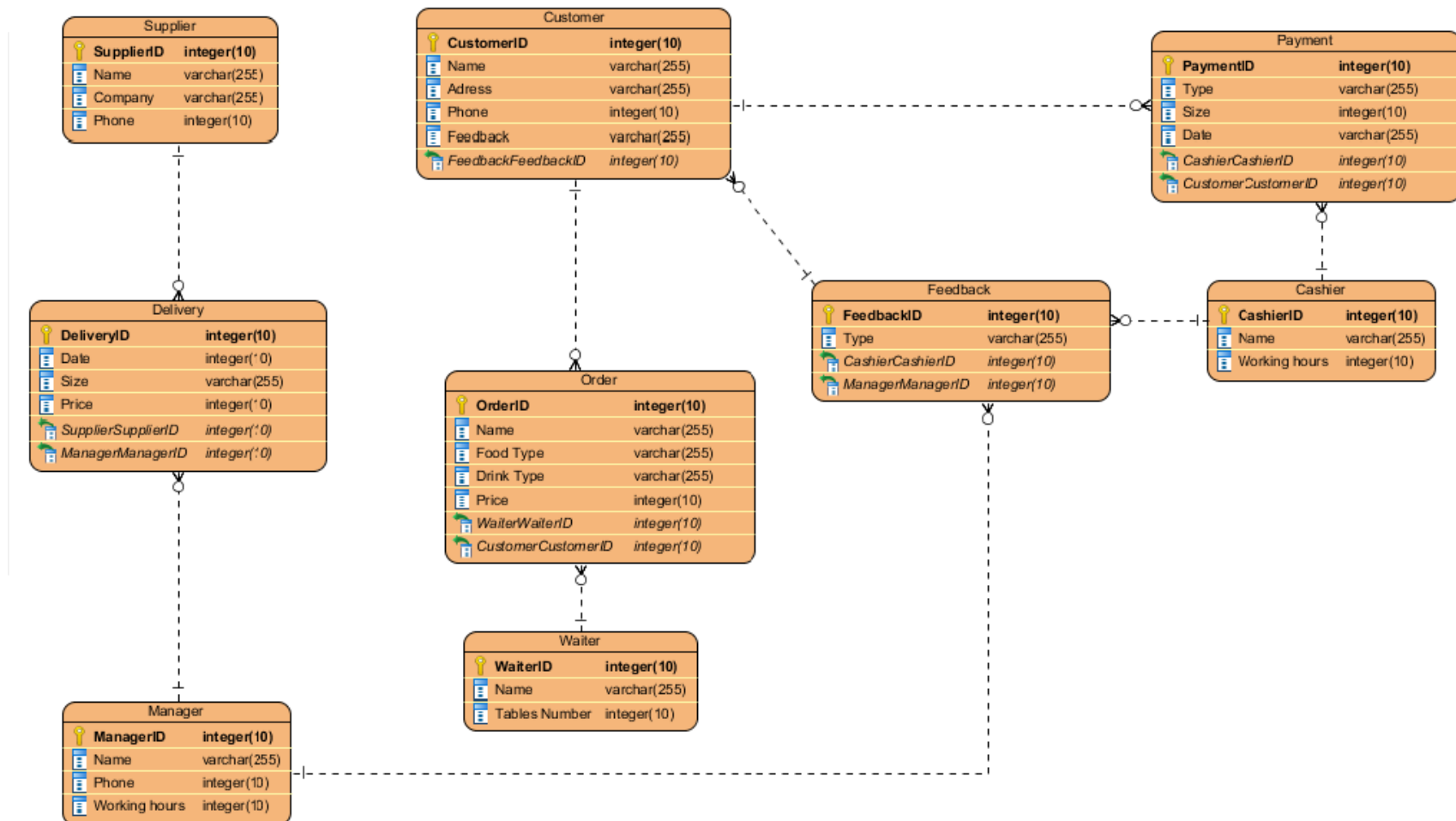
varchar(255): Name

CashierID is the primary key so it is unique.

RELATIONS

For every SUPPLIER there is going to be DELIVERY that is going to be accepted by the MANAGER. The CUSTOMER is going to give ORDER that is going to be taken by the WAITER. The CUSTOMER also gives the PAYMENT. The PAYMENT is accepted by the cashier who also gets FEEDBACK. The CUSTOMER is the one that gives FEEDBACK.

5. Analysis of data structures in data stores



The ERD shows the whole database of the restaurant with primary and foreign keys visible. This whole system has information for the process of the restaurant and the relationships between the entities.

6. Project of the interface

A Web Page

http://buffeterestaurant.com/menu

MENU

Food name	Size	Description	Checklist
Zurek	150ml	Polish soup	<input type="checkbox"/>
Pierogi	8 pieces	Polish dumplings	<input type="checkbox"/>
Kebab	100gr	Tortilla with meat	<input type="checkbox"/>
Hamburguer	150 gr	American style	<input type="checkbox"/>
Spagetti	200 gr	Italian style	<input type="checkbox"/>
Salad	200gr	Russian salad	<input type="checkbox"/>
Zupa grzybowa	150 ml	Polish soup	<input type="checkbox"/>

Drinks name	Size	Description	Checklist
Coca-cola	150ml	Polish soup	<input type="checkbox"/>
Water	250ml	Polish dumplings	<input type="checkbox"/>
Mineral Water	250ml	Tortilla with meat	<input type="checkbox"/>
Sprite	150ml	American style	<input type="checkbox"/>
Kompot	300ml	Italian style	<input type="checkbox"/>
Orange Juice	150ml	Russian salad	<input type="checkbox"/>
Lemonade	150 ml	Polish soup	<input type="checkbox"/>

>> Order -

A Web Page

http://buffeterestaurant.com/Payment

Payment

Name

Address

Email

Phone number

Way of payment

☐ Credit card

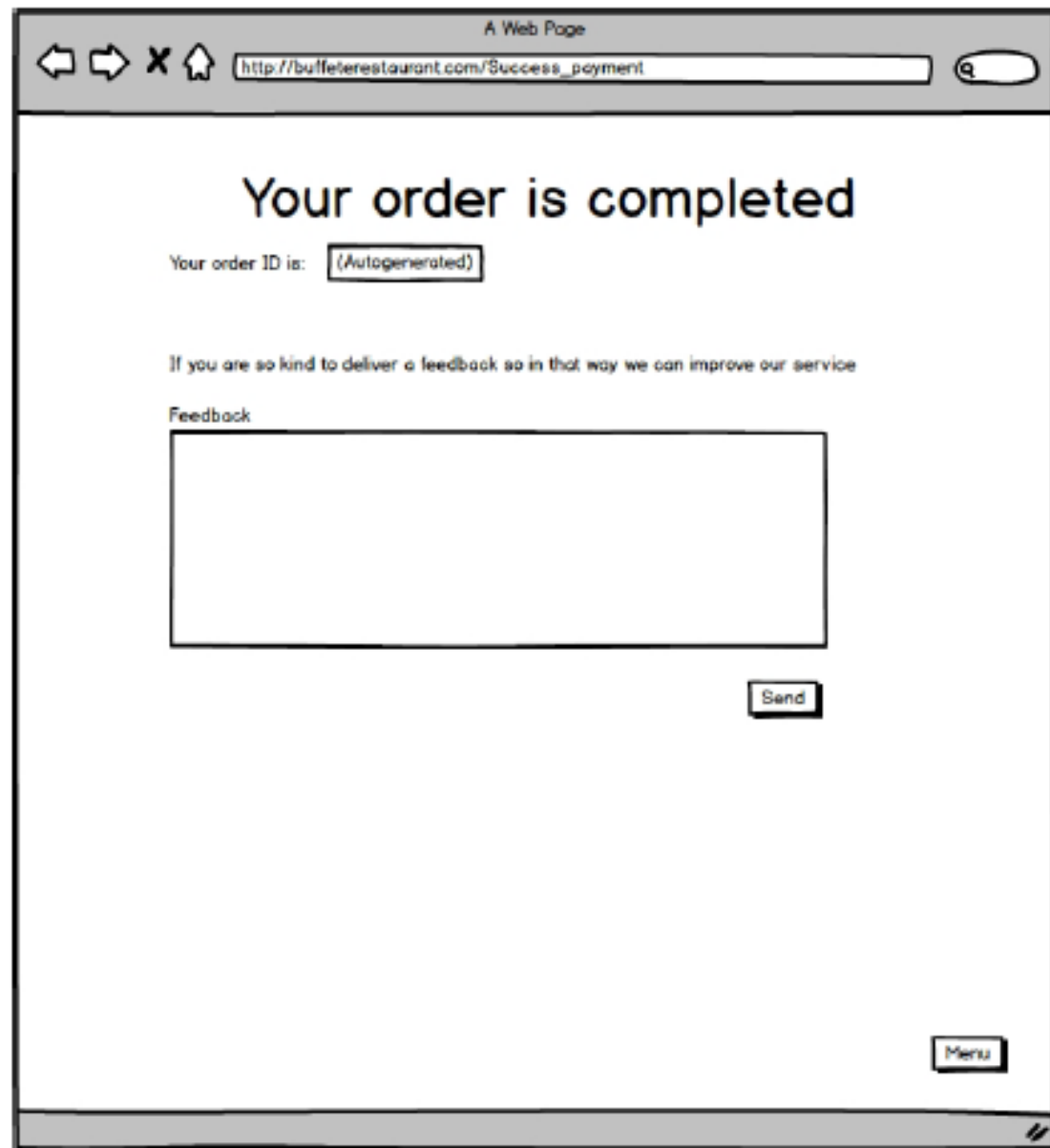
☐ Cash If you choose to pay by cash you have to pay directly to the cashier and mention your name

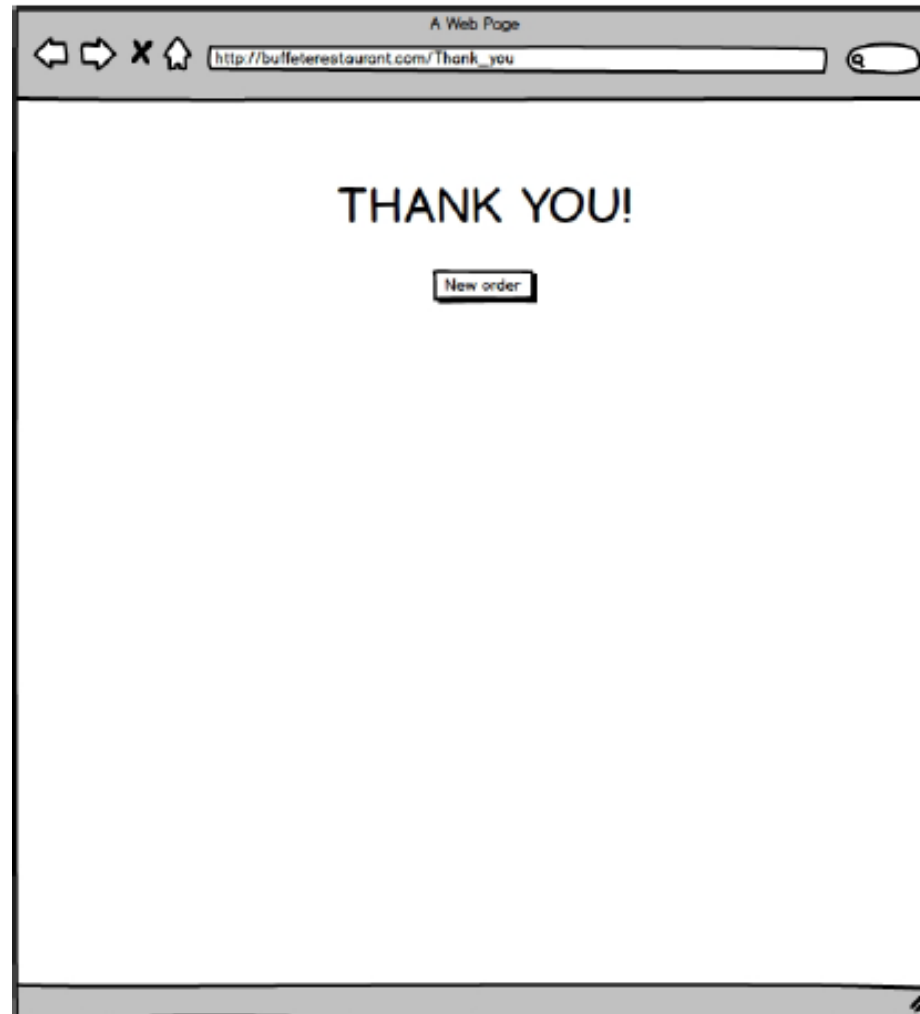
Credit card number

Expiration date

CVV

Next





7. Summary

7.1 Assumptions for implementation

The project is simple at simple glimpse but the quality of service that offers it's really good for the moment, it needs a verification of 1 to 2 months depending on the tests that the restaurant will experience.

Also another important thing to mention is that the system is designed so it can be modified to serve a better purpose.

7.2 Verification of the Project

This project is just theory based it means that it was not implemented in real life.

7.3 Final remarks and conclusions

The project from the perspective of theory successfully delivers all of the subjects seen in class although in real life it may be another story because we don't know for sure how much does it cost to really implement such system in a restaurant but besides that this project is made to satisfy the objectives which were mentioned at the beginning and so it does.

8. References

Sommerville, Ian (2011). Software engineering. Boston: Pearson. ISBN 0-13-705346-0.