

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY OF SOUSSE

المعهد العالي للإعلامية وتقنيات الاتصال بحمام سوسة



INSTITUTE SUPERIOR OF COMPUTERS

AND COMMUNICATION TECHNOLOGIES - HAMMAM SOUSSE

END OF STUDY INTERNSHIP MEMORY

Order N° 003/17

Presented for Master's Degree Professional Services Web and
Multimedia

Design and development of a mobile application for restaurant " The
Oasis " with loyalty techniques

Realized by:

Hajer BEN HASSEN

Supervised by:

Mrs. Hedia JEGHAM

Framed by:

M. Bessam BARKA

Host society:



Academic Year 2016-2017

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY OF SOUSSE

المعهد العالي للإعلامية وتقنيات الاتصال بحمام سوسة



INSTITUTE SUPERIOR OF COMPUTERS

AND COMMUNICATION TECHNOLOGIES - HAMMAM SOUSSE

END OF STUDY INTERNSHIP MEMORY

Order N° 003/17

Presented for Master's Degree Professional Services Web and
Multimedia

Design and development of a mobile application for restaurant " The
Oasis " with loyalty techniques

Realized by:

Hajer BEN HASSEN

Framer: Mr. Bessam BARKA

Date:.....

Signature:.....

Supervisor: Mrs. Hedia JEGHAM

Date:.....

Signature:.....

Academic Year 2016-2017

Dedications

To my dear parents,

*For their encouragement, advice and sacrifice, show
I hope that you will find in this work my deep appreciation and*

My great love for you

Your satisfaction will always remain my goal

To my brother,

*For your support and encouragement, you occupy a place
Peculiar in my heart. I dedicate this work to you wishing you
A bright future, full of happiness and success.*

To my dear friends,

*Thank you for the unforgettable moments we have shared
Together and for the pleasure I have enjoyed with you. Success and
joy are my Sincere wishes for you.*

To all who are dear to me

Acknowledgement

My thanks go first to the jury members who kindness to evaluate my project. I hope to find in their criticisms useful suggestions for the improvement of its content.

I also would like to thanks **Mrs. Hedia JEGHAM**, my Supervisor at Hammem Sousse's Higher Institute of Computer Science and Communication Technologies, his guidance, his precious help, his availability, his pertinent remarks, his continuous encouragement and sympathy. You have taught me so much about the meaning and the signification of research and your feedback on my work and especially my writing has been very valuable. Thanks a lot.

I would also like to sincerely thank my coach **Bessam BARKA** who spared no effort to be understanding and available.

My thanks go especially to my cousin **Imen JABALLAH** for his help and support. I would never be able to thank you enough. I am out of words to describe my deep respect and appreciation for your support through the busy and hard times.

Special thanks to all my colleagues for their collaboration and which helped me to realize this project.

Table of Contents

General introduction	1
Chapter 1: General Presentation	3
1. Introduction.....	4
2. Presentation of the host organization	4
3. Presentation of the project.....	4
4. Specifications	4
4.1. Expression of functional requirements	4
4.2. Expression of non-functional requirements.....	5
5. Background of the project	6
5.1. The e-commerce context.....	6
5.2. The e-loyalty context.....	6
5.2.1. Communication based programs	6
5.2.2. Compensation based programs	6
5.3. The study of existing system.....	7
5.3.1. The existing mobile-application catering	7
5.3.2. Weaknesses of the existing system.....	10
6. Methodology.....	11
7. Estimated cost of the project.....	11
8. Conclusion.....	12
Chapter 2: Requirements Specification	12
1. Introduction.....	13
2. Semi-formal functional requirements specification	13
2.1. Identification of actors in the system.....	14
2.1.1. Identification of the actors in the front office	14
2.1.1.1. The functions carried out by the mobinaute.....	14
2.1.1.2. The functions carried out by the prospect customer.....	14
2.1.1.3. The functions carried out by the customer	14
2.1.1.4. The functions carried by the loyal customer.....	15
2.1.2. Identification of the manager in the back office	15
2.2. Context diagram.....	15
2.3. The overall use case diagram	16
2.4. Refinement of the use case diagrams	18
2.4.1. Use case « Navigate »	18
2.4.2. Use case « Place order »	18
2.4.3. Use case « Manage loyalty points ».....	19
2.4.4. Use case « Manage dishes »	19
2.4.5. Use case « Manage loyalty techniques »	19
2.4.6. Use case « Manage Platform »	20
2.4.7. Use case « Manage booking »	20
2.5. Text description	20

Table of Contents

2.5.1.	Use case « Authenticate »	20
2.5.2.	Use case « Add dish»	22
2.5.3.	Use case « Validate reservation »	23
2.5.4.	Use case « Consult status order »	24
2.5.5.	Use case « Make reservation »	24
2.6.	Package of the use case diagram	26
3.	Specification of technical requirements.....	27
3.1.	Design of the architecture of the project.....	27
3.1.1.	The MVC pattern	27
3.1.2.	Architecture.....	27
3.1.2.1.	The model.....	27
3.1.2.2.	The views.....	27
3.1.2.3.	The Controller	27
3.1.3.	Behavior	28
3.1.4.	Advantages et disadvantages:	28
3.2.	Interface prototyping	29
3.2.1.	Front office part	29
3.2.1.1.	« Home » interface	29
3.2.1.2.	« Menu » interface.....	30
3.2.2.	Back office part.....	30
3.2.2.1.	« Dishes » interface.....	30
3.2.2.2.	« Statistics » interface.....	31
4.	Conclusion.....	31
Chapter 3:	Conceptual Study	32
1.	Introduction.....	33
2.	Static study.....	33
2.1.	The class diagram	33
2.2.	Optimization of semantics by the l'OCL.....	35
2.2.1.	what is the OCL	35
2.2.2.	Optimized class diagram.....	35
2.3.	Package class diagram.....	37
2.4.	Deployment diagram	39
3.	Dynamics study	40
3.1.	Activity diagram	40
3.1.1.	Authentication.....	40
3.1.2.	Add reservation.....	41
3.1.3.	Manage order	42
3.2.	Sequence diagram.....	43
3.2.1.	Sequence diagram « Edit dish ».....	43
3.2.2.	Sequence diagram « Add dish »	44
3.2.3.	Sequence diagram « Manage order »	45
3.2.4.	Sequence diagram « Consult statistics »	46
3.3.	State transition diagram	46
4.	Conclusion.....	46

Table of Contents

Chapter 4 : Implementation.....	47
1. Introduction.....	47
2. Development environment.....	48
2.1. Hardware environment.....	48
2.2. Software environment.....	48
3. Principle of ergonomics	50
3.1. Template « Back-office » part	50
3.2. Choice of colors	50
3.3. The web page structure	51
4. Aspects of physical model.....	52
5. Main interfaces	59
5.1. Back-office part.....	59
5.1.1. Connection Interface	59
5.1.2. « Home page » interface	60
5.1.3. « order List » Interface.....	61
5.1.4. « Details order » Interface.....	61
5.1.5. « Bill » Interface	63
5.1.6. Statistics interface « turnover by zip code ».....	63
5.1.7. Statistics interface « number of orders per month »	64
5.1.8. Statistics interface « number of booking per day »	64
5.1.9. Statistics interface « Dishes ordered by month »	65
5.1.10. « dish list » Interface.....	65
5.1.11. « Details dish » Interface.....	66
5.1.12. « Add promotion » Interface	67
5.1.13. « loyalty system » Interface.....	68
5.2. Front-office part	68
5.2.1. Splach interface	68
5.2.2. Home interface	69
5.2.3. Menu interface	69
5.2.4. Details dish interface.....	70
5.2.5. Delivery interface	70
5.2.6. Sign in interface.....	71
5.2.7. Account interface	71
5.2.8. Order interface.....	72
5.2.9. “Manage loyalty points” interface	73
5.2.10. “Pay with PayPal” interfaces	74
6. Conclusion.....	74
General conclusion.....	75
Bibliography	77
Webography	77

List of figures

Figure 1: Logo of the «The Oasis » restaurant	4
Figure 2: Logo of the "West Coast Tavern" application	7
Figure 3: logo of the “Diwan Sultan" application.....	8
Figure 4: Print screen of the "Diwan Sultan " application.....	9
Figure 5: logo of the "Johnny Mac's" application	9
Figure 6: Print screen of the "Johnny Mac's " application.....	10
Figure 7: the V-model life cycle	11
Figure 8: Gantt chart of the project.....	12
Figure 9: Context diagram	16
Figure 10: the overall use case diagram	17
Figure 11: use case diagram " Navigate "	18
Figure 12: use case diagram " place order".....	18
Figure 13: use case diagram " manage loyalty points"	19
Figure 14: use case diagram " manage dishes ".....	19
Figure 15: use case diagram " loyalty techniques"	19
Figure 16: use case diagram "manage platform".....	20
Figure 17: Use case « Manage booking »	20
Figure 18: package of the use case diagram.....	26
Figure 19: MVC architecture.....	28
Figure 20: Prototyping of the interface "Home".....	29
Figure 21: Prototyping of the interface "Menu"	30
Figure 22: Prototyping of the interface "dishes"	30
Figure 23: Prototyping of the interface "Statistics"	31
Figure 24: the class diagram	34
Figure 25: Class diagram with "OCL.....	36
Figure 26: Package class diagram	38
Figure 27: Deployment diagram	39
Figure 28: Activity diagram "Authentication"	40
Figure 29: Activity diagram " Add reservation "	41
Figure 30: Activity diagram " Manage order "	42
Figure 31: Sequence diagram « Edit dish ».....	43
Figure 32: Sequence diagram « Add dish ».....	44
Figure 33: sequence diagram « manage order ».....	45
Figure 34: Sequence diagram « Consulter Statistics ».....	46
Figure 35: State transition diagram.....	46
Figure 36: Connection Interface	59
Figure 37: « Home page » interface	60
Figure 38: « order List » Interface	61
Figure 39: « Details order » Interface.....	61
Figure 40: « change status order » Interface.....	62

List of figures

Figure 41: « Bill » Interface	63
Figure 42: Statistics interface « turnover by zip code»	63
Figure 43: Statistics interface « number of orders per month »	64
Figure 44: Statistics interface « number of booking per day »	64
Figure 45: Statistics interface «Dishes ordered by month »	65
Figure 46: « dish list » Interface	65
Figure 47: « Details dish » Interface	66
Figure 48: « Add promotion » Interface	67
Figure 49: « loyalty system » Interface.....	68
Figure 50: Splach interface	68
Figure 51:Home interface	69
Figure 52: Menu interface	69
Figure 53: Details dish interface.....	70
Figure 54: Delivery interface	70
Figure 55: Sign in interface.....	71
Figure 56: Account interface	71
Figure 57: Order interface.....	72
Figure 58: "Manage loyalty points" interface	73
Figure 59:" Pay with PayPal" interfaces.....	74

List of tables

Table 1: Text description of use cases “Authenticate”	21
Table 2: Text description of use case "Add dish"	22
Table 3: Text description of use cases “validate reservation”	23
Table 4: Text description of use cases “Check order status”	24
Table 5: Text description of use cases “Make reservation”	25
Table 6: The characteristics of the used machine	48
Table 7: table assortment	52
Table 8: Bill table	52
Table 9: Category table.....	52
Table 10: Configuration table	52
Table 11: Customer table	52
Table 12: Prospect customer	53
Table 13: Delivery table	53
Table 14: Dish table	53
Table 15: Dish assortment table	53
Table 16: event organized table	54
Table 17: event requested table	54
Table 18: Favorites table	54
Table 19: Fos user table	55
Table 20: line order table	55
Table 21: Loyal customer table.....	55
Table 22: Manager table.....	55
Table 23: Method payment table.....	56
Table 24: Orders table	56
Table 25: Picture assortment table.....	56
Table 26: Picture dish table.....	56
Table 27: Picture event table.....	57
Table 28: Picture resto table	57
Table 29: Point loyalty table.....	57
Table 30: price period table	57
Table 31: Price period dish table	57
Table 32: Promo period table.....	58
Table 33: Reservation table.....	58
Table 34: Review table	58
Table 35: Type event table.....	58

General introduction

General introduction

The Internet has revolutionized the world like nothing else before. Making its appearance other technologies based on the Internet were able to see the light. E-commerce, one of these technologies with strong potential came to revolutionize the way to make purchases and sales, several companies have progressively invested this network in order to develop their activities.

The concept of e-commerce covers possible applications of computing using information and communication « **ITC** » technology to treat permanent relationships of a business with external organizations or individuals. It offers a new way to trade by allowing the product search and collect advice from user, to make purchases and sales without having to reallocate.

For the success of e-commerce service, the consumer's trust must be earned by loyal techniques and a reliable delivery service must be available. It is within this context that we have designed and developed this project.

Our project is to develop a web and mobile application for a restaurant with the mentioned loyal techniques. We must take into account all the new aspects of e-commerce especially the issue of security where we discuss this subject with precision.

The manager of « The Oasis » restaurant suggested the development of a mobile application which provides the progress of various features needed for the automation processes which becomes a mandatory need to ensure good management of the restaurant. The application is concerned by organizing the passage of reservations, orders, meals, events. It allows the conservator to win customers by the loyalty points gained and assess the quality of its restaurant by consulting the statistics and the user's evaluation of the dishes.

This report focuses on four chapters namely:

The first chapter is devoted to **General presentation** of the project it gives a general idea on the frame of the project through the definition of the mission, the essential features, context and comparative study.

The second chapter titled « **Requirements specification** » describes a presentation of functional and non-functional requirements of the system to develop by specifying the different actors of the future system, the model for use with the text description and a prototyping of future interfaces.

The third chapter named « **Conceptual study** » will be dedicated to the exhibition of our conceptual approach for the development of the application by the construction of the class diagram, specification of scenarios through sequence diagrams, activities diagrams and transition state diagrams.

The fourth chapter « **Implementation** » will present the development tools that have been used as well as a the most important interfaces of the application.

Chapter 1: General Presentation

General introduction	1
Chapter 1: General Presentation	3
1. Introduction	4
2. Presentation of the host organization.....	4
3. Presentation of the project.....	4
4. Specifications.....	4
4.1. Expression of functional requirements.....	4
4.2. Expression of non-functional requirements	5
5. Background of the project	6
5.1. The e-commerce context.....	6
5.2. The e-loyalty context.....	6
5.2.1. Communication based programs.....	6
5.2.2. Compensation based programs	6
5.3. The study of existing system	7
5.3.1. The existing mobile-application catering	7
5.3.2. Weaknesses of the existing system	10
6. Methodology	10
7. Estimated cost of the project.....	11
8. Conclusion	11

1. Introduction

In this chapter, we first present the host organization and the specifications of the project. Then we put the project in its general and specific context to be able to criticize the existing and build the foundations of our future application. We finish by the choice of the methodology and the cost estimation.

2. Presentation of the host organization

« The Oasis St Albans » is a modern restaurant created in February 2016. It is located in St Albans and is distinguished by its menu that covers classical Mediterranean and Middle Eastern cuisine.

The labour of the organization is managed by a team of: 3 managers, 5 heads and 5 servers.



Figure 1: Logo of the «The Oasis » restaurant

3. Presentation of the project

The restaurant industry has evolved recently in a remarkable way. It is certainly due to the use of computerized information systems. It is in this sense that « The Oasis » restaurant manager suggested the development of a mobile application to improve the efficiency of the external and internal management of the restaurant, to achieve a better organization of the work and get a competitive advantage. A priori, the most significant features have been: management of orders, management of products, booking of tables, management of events, loyalty techniques as well as the consultation of statistics and the view of supporters, clients...etc.

4. Specifications

4.1. Expression of functional requirements

This project is aimed to design and develop an application in two parts: A Web part for the management of the Back Office and Android for the front part. On the web application, two big players appear spontaneously, these are: the restaurant manager and the mobile consumer. Roughly, from the expression of the needs of our interlocutor, they can do the following:

The restaurant manager can:

- ✓ Manage dishes.
- ✓ Manage orders.
- ✓ Manage bookings.
- ✓ Manage events.
- ✓ Manage promotions.

- ✓ Manage loyalty.
- ✓ Consult reviews.
- ✓ Consult statistics.
- ✓ Consult customers.

The user of the application can:

- ✓ Consult offers and promotions.
- ✓ Consult menus.
- ✓ Consult events.
- ✓ Create a user account.
- ✓ Contact the restaurant manager for a consultation, clarification or to report a case.
- ✓ Consult general information about the restaurant.
- ✓ Consult notifications.
- ✓ Share on social networks.
- ✓ Manage an online shopping basket.
- ✓ Add to favorite dishes.
- ✓ Place order.
- ✓ Make a reservation.
- ✓ Organize an event.
- ✓ Manage loyalty points.

4.2. Expression of non-functional requirements

Non-functional requirements specify the properties of the system such as the environmental and implementation constraints, ergonomics, performance, maintenance, scalability and flexibility. Some non-functional requirements are general and cannot be attached to a particular use case.

For our project the non-functional requirements include:

- ✓ **Performance:**
 - Response time: loading the application, opening the screen and delays of refreshment.
 - At the time of treatment: the receipt of notifications and notifications must be in real time.
- ✓ **Ergonomic:**

For our application, we must obey the following ergonomic constraints:

- Allow quick access to information.
- Simple and understandable interface.
- Organization of the headings, clear and attractive tabs.
- The application must be user-friendly: i.e. guide the customer or visitor to use the application in a simple and fast way; being written with language that is understandable by the user, present information in a clear and simple way, make choices or entries made by the customer appear clearly.

5. Background of the project

In this section we try to put our project within its framework trade in several contexts: the economic, commercial, gastronomic and application context i.e. which already exists as computerized applications or mobile in the field of restoration. This effort could be models, standards, recommendations and business practices that would help us to familiarize ourselves and quickly take ownership of specific features of the project. We will be exhibiting at this respect the context of e-commerce, the context of the e-loyalty and the existing in terms of applications of restoration.

5.1. The e-commerce context

It has been nearly a century now since humanity is moving toward innovation and technological progress in giant steps. Including the democratization of computing as well as the integration of the Internet with the broad public what has caused a major change in most of the commercial, economic and industrial sectors. These changes have emerged with astonishing rapidity. This led officials of the above sectors to review their trade policy to adapt to these unexpected changes. This is the reason why companies have begun to make huge investments in the creation of merchant sites, i.e. platforms allowing the electronic commerce act, in order to attract the new type of customers, at the level of its behavior in the face of the notion of purchase and sale. Indeed, commercial action occurs between two entities (buyers and sellers) electronically exchanging goods and services over the Internet. Here, we are talking particularly of the beginner electronics for restaurant business that requires transactional mechanisms, the model « B2C » (Business to Consumer) specifically on the no digitalisable ephemeral products and services requiring efficient distribution logistics and supply and high performance storage system [W1].

5.2. The e-loyalty context

Implementing a loyalty program allows both the customer to be informed daily on the updates of the restaurant; who is conducting its policy to establish and impose its brand name; on the special offers of the moment. A retention strategy by expanding e-loyalty which is dedicated to build a long-term relationship based on trust with its customers. There are two major loyalty programs that can be combined:

5.2.1. Communication based programs

These programs are intended to inform its customers daily, to provide advice through a business expertise, while providing a form of customer care throughout its life cycle.

5.2.2. Compensation based programs

Programs based on compensation allow customers to regularly accumulate points after a purchase, vouchers, discount or promotional offers. These forms of programs are also aimed to thank customers for their loyalty by offering gifts, invitations...

For my project I choose to use a system that accumulates reward points by following the next steps: when customers buy products, they earn a certain number of loyalty points. Points being credited to a customer only when he pays his order, he could use the gained loyalty points during future orders, and pay using these points if he had accumulated enough. Each product has a cost-related given points, specified by the restorer to retain customers and to each order of dishes the restaurant calculates and gives customers a sum of points. When the customer has collected a sufficient number of points he can order and pay through this medium. For the client to enjoy this service, it is necessary to create an account previously.

There are 3 different techniques of accounting for loyalty points:

Per unit of points: Loyalty points that a customer will win for each product purchased are fixed independently and in advance for each product. They are calculated on the basis of the actual price of the product. A more expensive dish brings more loyalty points than a cheaper dish.

By level of purchase: Loyalty points are acquired by price ranges of the total value of the food stored in the basket, in other words per total order if the basket is converted into actual order.

By purchase threshold: Loyalty points are acquired when the purchase threshold that you set is reached or exceeded [W2].

5.3. The study of existing system

5.3.1. The existing mobile-application catering

Currently you cannot deny the necessity of using mobile and web applications for managing restaurants. The automation processes of running a restaurant became a mandatory need to ensure good management. It is generally needed to organize the passage of orders as well as the management of dishes, reservations etc. So, it plays a vital role to the customer as well as the restaurant owner who always seeks to ensure good management by making this painful task computerized. It is in a dual role that we are presenting this paragraph, by providing examples of existing mobile applications on one hand and their criticism on the other.

We include existing applications abroad:

West Coast Tavern



Figure 2: Logo of the "West Coast Tavern" application

This mobile application is used for a restaurant situated in San Diego; it is beneficial to the point that it allows:

- ✓ Organize events.

Chapter 1: General Presentation

- ✓ Pass reservations.
- ✓ Share with friends.
- ✓ Enter the QR Code Scanner.

It is remarkable that this application lacks several features namely:

- ✓ Loyalty techniques.
- ✓ Booking page is still inaccessible.
- ✓ Making orders and deliveries.
- ✓ Adding to Favorites.
- ✓ Paying method.
- ✓ Customer reviews.

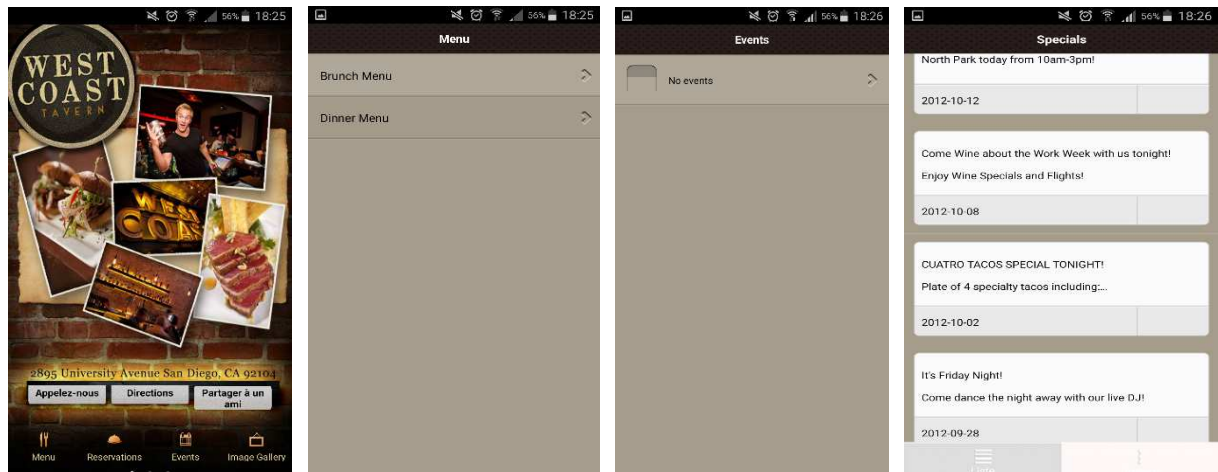


Figure3: Print screen of the "West Coast Tavern" application

Diwan Sultan



Figure 3: logo of the "Diwan Sultan" application

This Android application is owned by a restaurant located in Saudi Arabia. It provides the following features:

- ✓ Book.
- ✓ Consult menu.

Chapter 1: General Presentation

- ✓ Share on social networks.
- ✓ Send a message.
- ✓ Consult the restaurant locator (through a map).

But it has several shortcomings as it does not provide for the following:

- ✓ Organization of events.
- ✓ Place Order.
- ✓ Customer evaluation.
- ✓ Addition to favorites.
- ✓ Online payment.
- ✓ Loyalty techniques.

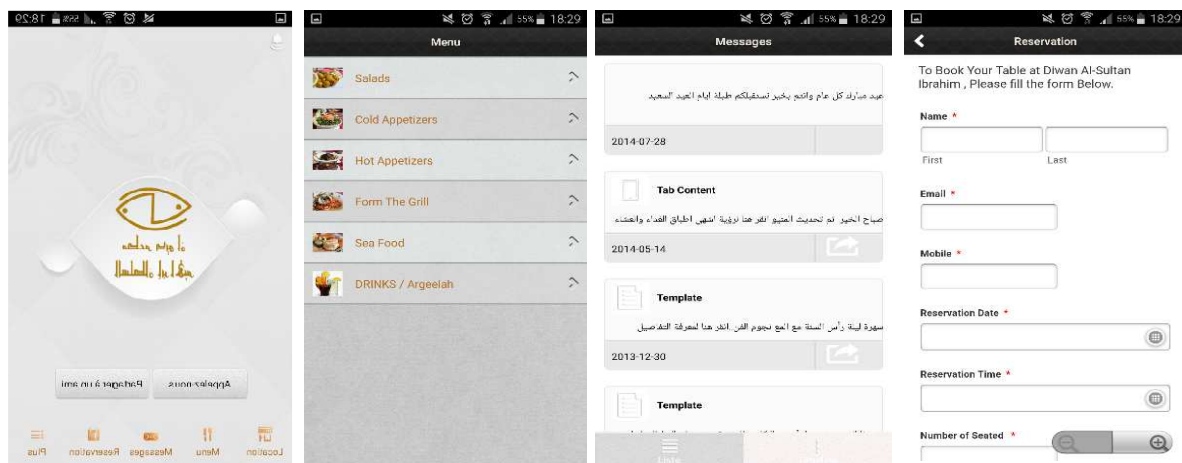


Figure 4: Print screen of the "Diwan Sultan " application

Johnny Mac's



Figure 5: logo of the "Johnny Mac's" application

Johnny Mac's is located in Henderson and through its application, the customer can:

- ✓ Print the menu.
- ✓ Take pictures.
- ✓ Share on social networks.
- ✓ Send messages.
- ✓ Fill out reviews.

Several failures are:

- ✓ Make reservation.
- ✓ Place an order.
- ✓ Organize events.
- ✓ Pay online.
- ✓ Add to favorites.
- ✓ Loyalty techniques.

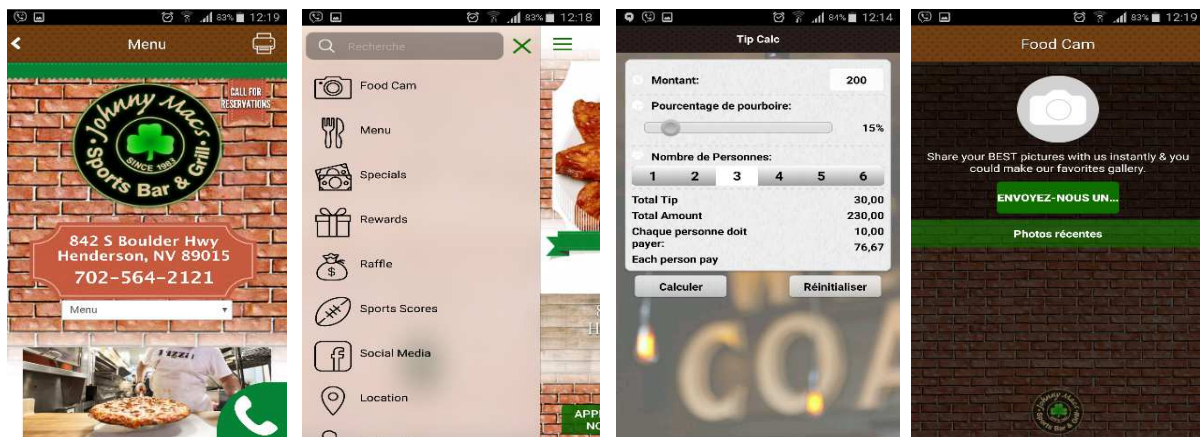


Figure 6: Print screen of the "Johnny Mac's " application

5.3.2. Weaknesses of the existing system

The current information system of 'The Oasis' restaurant suffers from a number of weaknesses and problems listed as follows:

- ✓ Inability to place orders for delivery, to organize events or evaluate dishes from the current web site.
- ✓ Inability for the manager to self-assess performance and work method.
- ✓ Unstructured and lack of organization in the workflow.
- ✓ The communication is only carried out by phone calls or is headed straight to the restaurant.

6. Methodology

The appropriate choice of software development process led to the successful completion of the project, it is for this reason we have chosen the « **V-model** ».

V-model means Verification and Validation model. Just like the waterfall model, the V-Shaped life cycle is a sequential path of execution of processes. Each phase must be completed before the next phase begins. **V-Model** is one of the many software development models. Testing of the product is planned in parallel with a corresponding phase of development in V-model [1].

The V model has several advantages which verifies our choice:

- ✓ It ensures a perfect product as the final delivery model has several phases of testing in relation to the description of the initial needs.
- ✓ Work goes quite naturally because the V model can predict the tests to be performed at the time when designing a feature or interface.
- ✓ Objectives, roles and steps are defined clearly from the beginning.

The V-cycle will consist of 9 steps:

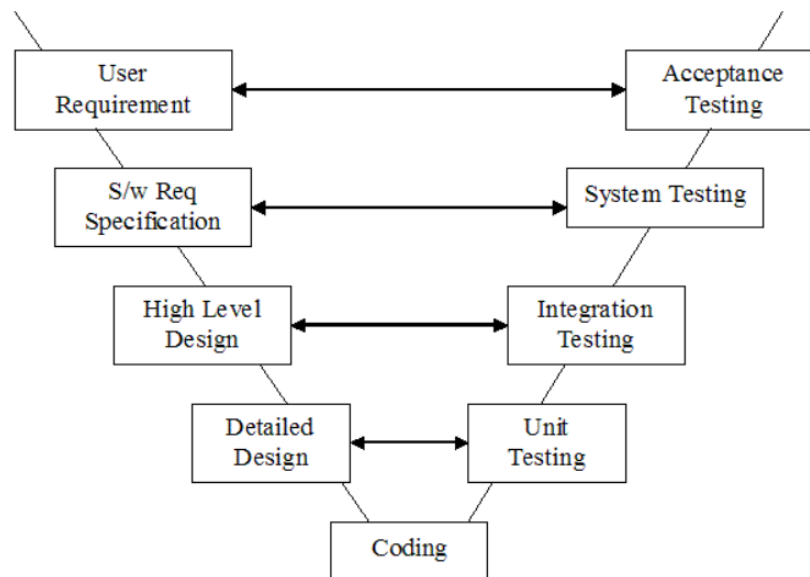


Figure 7: the V-model life cycle

7. Estimated cost of the project

“GanttProject” is a tool for managing projects by offering a model called the Gantt chart. GanttProject allows for the breaking down into tree time periods and assign resources to each of the duties provided for in the schedule.

The purpose of using Gantt Project was the determination of planning to manage our time and distribute it fairly, as it allows clarifying the duration of each task which is related to the whole process of the development of the project stages [W3].

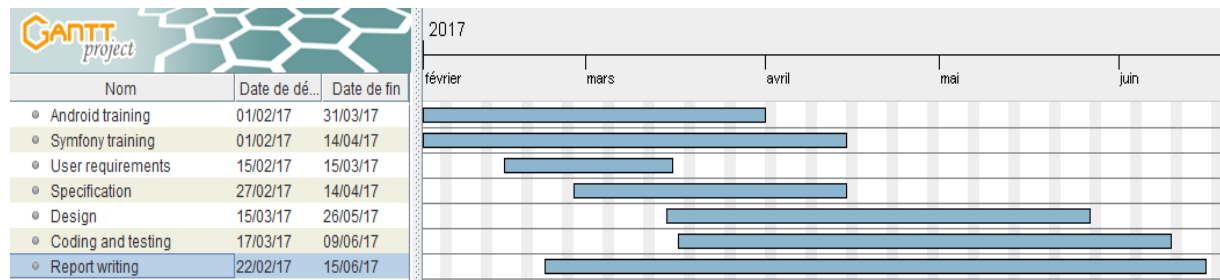


Figure 8: Gantt chart of the project

8. Conclusion

In this chapter, we presented the host organization, the specifications of our project. We have specified the nature of our project in these various contexts allowing for a comparative study of the existing application. The working methodology was chosen and stated. Now it is time to put the accomplished task of requirements specification in action.

Chapter 2: Requirements Specification

Chapter 2: Requirements Specification	12
1. Introduction	13
2. Semi-formal functional requirements specification	13
2.1. Identification of actors in the system	13
2.1.1. Identification of the actors in the front office	13
2.1.1.1. The functions carried out by the mobinaute	13
2.1.1.2. The functions carried out by the prospect customer	14
2.1.1.3. The functions carried out by the customer	14
2.1.1.4. The functions carried by the loyal customer	14
2.1.2. Identification of the manager in the back office	14
2.2. Context diagram	15
2.3. The overall use case diagram	15
2.4. Refinement of the use case diagrams	17
2.4.1. Use case « Navigate »	17
2.4.2. Use case « Place order »	17
2.4.3. Use case « Manage loyalty points »	18
2.4.4. Use case « Manage dishes »	18
2.4.5. Use case « Manage loyalty techniques »	18
2.4.6. Use case « Manage Platform »	19
2.4.7. Use case « Manage booking »	19
2.5. Text description	19
2.5.1. Use case « Authenticate »	19
2.5.2. Use case « Add dish »	21
2.5.3. Use case « Validate reservation »	22
2.5.4. Use case « Consult status order »	23
2.5.5. Use case « Make reservation »	23
2.6. Package of the use case diagram	25
3. Specification of technical requirements	26
3.1. Design of the architecture of the project	26
3.1.1. The MVC pattern	26
3.1.2. Architecture	26
3.1.2.1. The model	26
3.1.2.2. The views	26
3.1.2.3. The Controller	26
3.1.3. Behavior	27
3.1.4. Advantages et disadvantages:	27
3.2. Interface prototyping	27
3.2.1. Front office part	28
3.2.1.1. « Home » interface	28
3.2.1.2. « Menu » interface	28
3.2.2. Back office part	29
3.2.2.1. « Dishes » interface	29
3.2.2.2. « Statistics » interface	29
4. Conclusion	29

1. Introduction

This chapter is devoted to the first phase of the project where we attempt to go into details with the future functional system by a meticulous description of its features. For this fact we have first identified the actors, drawn up a diagram of the system context as a general view, proposed a global use case diagram and then refined it. We have also specified the technical needs by the method of prototyping.

2. Semi-formal functional requirements specification

The functional requirements are the main features required for a not yet existing system to designate a necessary step in the development of any computer system useful for the customer. They are considered to be vital needs expressed by the client at the completion of its business processes.

2.1. Identification of actors in the system

There is a distinction between the actors of the front-office and back-office.

2.1.1. Identification of the actors in the front office

The above features are indeed made by any user of the front-office of the application but to degrees of different States. A user can be a **mobinaute**, which can benefit from all the features that do not linger him to fill out a form and register, including navigate, share on social networks, complete and manage a basket. Adding to favourites for example transforms the State of a passenger user to another State where our platform requires a minimum of identification in order to book him a space dedicated to the storage of his favourites. As for the prospect customer and because we have some little information about him, including his electronic mail address, we can keep him prospect and target him by notifications.

The most engaging feature is the transformation of the basket in order, to state it as a **customer**. The client could do everything that any mobile user or prospect customer can, but it can make a reservation, opine, organize an event, see their order history and/or the status of his current command, pay online, view their history of reservations and manage their personal account. They will be proposed to the loyalty card and so we start to retain more.

2.1.1.1. The functions carried out by the mobinaute

The mobinaute can benefit from all the features that are not binding in terms of time and benefit from the information available.

- ✓ Consult menu.
- ✓ Consult offers and promotions.
- ✓ Manage a basket.
- ✓ Consult notifications.
- ✓ Consult events.
- ✓ Share on social networks
- ✓ Consult restaurant info.
- ✓ Contact.
- ✓ Create an account.

2.1.1.2. The functions carried out by the prospect customer

The prospect customer can benefit from all the features above in addition to others who linger to introduce some small information about him, including his electronic mail address. As a result, we can tag him as prospect and target it by notifications. It can also:

- ✓ Add to favorite dishes.
- ✓ Consult history of the favorites.
- ✓ Manage account.

2.1.1.3. The functions carried out by the customer

Since he also plays the role of a user and a prospect customer, he can do the following:

- ✓ Place Order.

- ✓ Make reservation.
- ✓ Consult current order.
- ✓ Consult order history.
- ✓ Consult reservation history.
- ✓ Organize an event.
- ✓ Vote
- ✓ Pay

2.1.1.4. The functions carried by the loyal customer

As he is already a client, he can:

- ✓ Manage his loyalty points by converting.

2.1.2. Identification of the manager in the back office

The back office is the party responsible by the manager of the restaurant. He plays the role of the administrator in our sense, therefore, he can:

- ✓ Manage dishes.
- ✓ Manage orders.
- ✓ Manage reservations.
- ✓ Manage events.
- ✓ Manage promotions.
- ✓ Manage loyalty techniques.
- ✓ Consult reviews.
- ✓ Consult statistics.
- ✓ Consult customers.

2.2. Context diagram

Using the formalism class diagrams, the context diagram allows to introduce the system to model, generally in the form of a 'black box' and actors that interact with this so-called system through associations with the cardinalities.

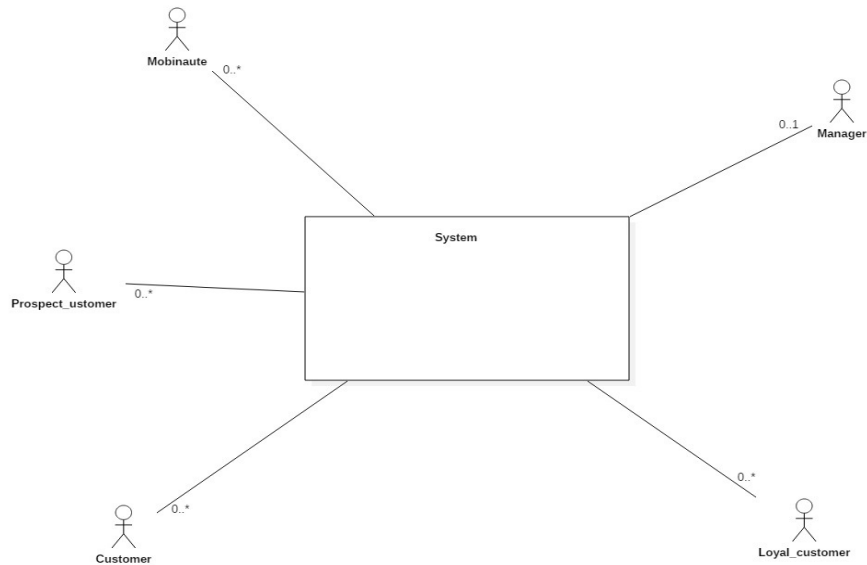


Figure 9: Context diagram

2.3. The overall use case diagram

A use case diagram captures the behaviour of a system, subsystem, a class or a component such as an outside user sees.

Use cases allow expressing the needs of the users of a system, so they are a user of this need instead of a computer vision-oriented.

The use case diagram also allows structuring the requirements of users and the goals of a system. It is used to:

- ✓ Identify the features of the system.
- ✓ Identify the system actors.
- ✓ Set the goals of the system.
- ✓ Define the interactions between the players and the system.

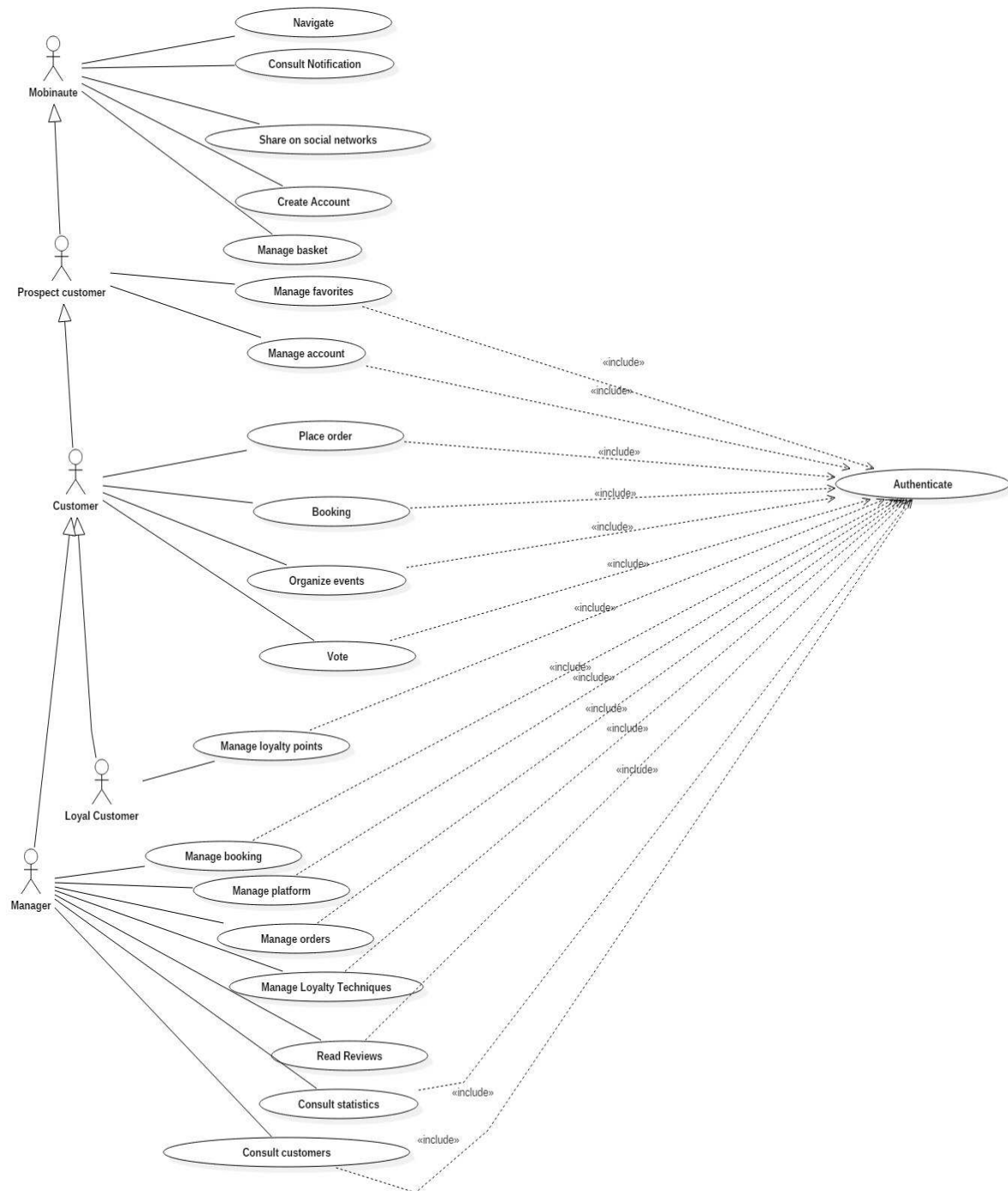


Figure 10: the overall use case diagram

2.4. Refinement of the use case diagrams

2.4.1. Use case « Navigate »

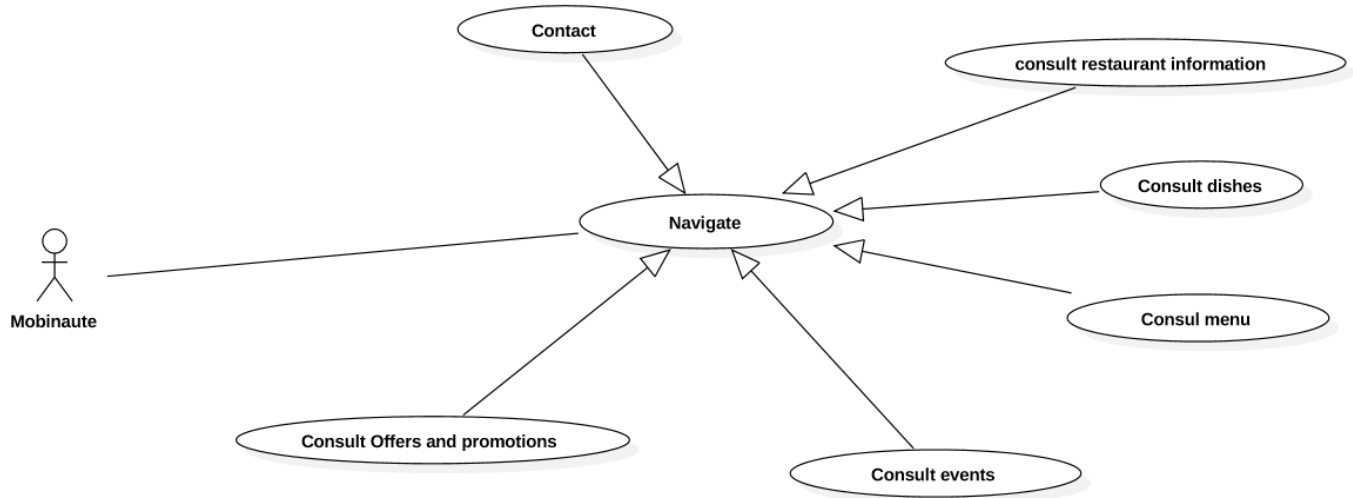


Figure 11: use case diagram " Navigate "

2.4.2. Use case « Place order »

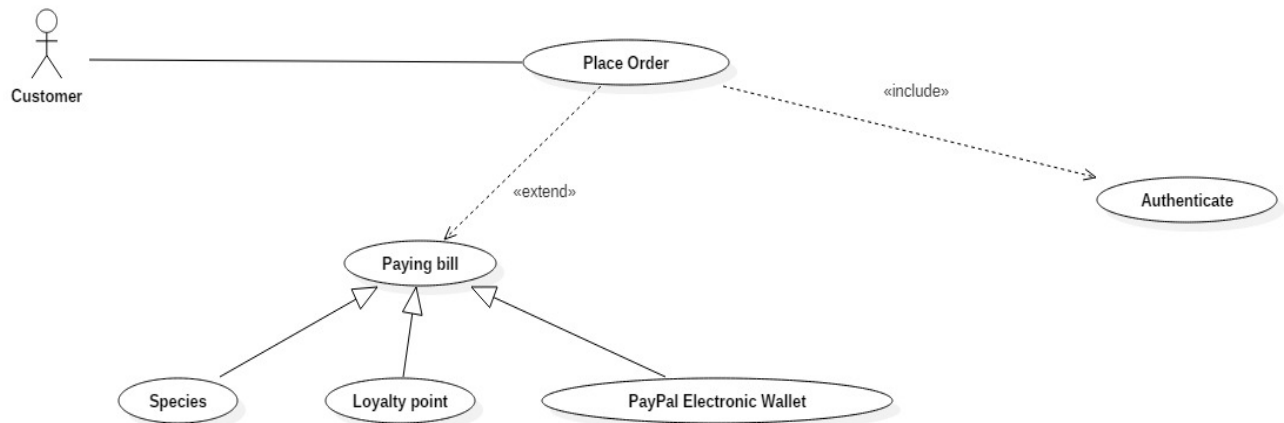


Figure 12: use case diagram " place order"

2.4.3. Use case « Manage loyalty points »

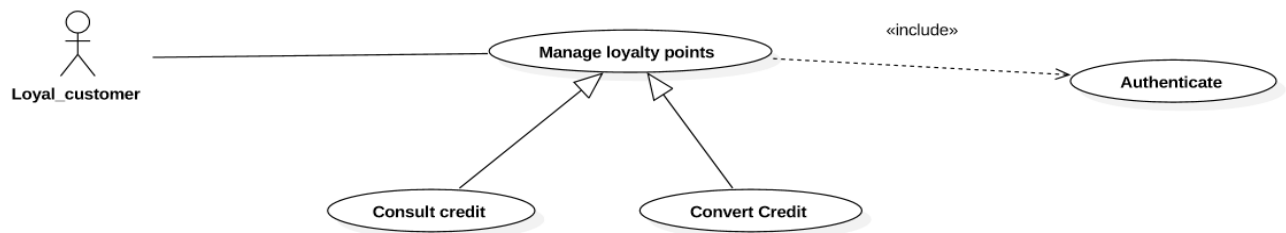


Figure 13: use case diagram " manage loyalty points"

2.4.4. Use case « Manage dishes »

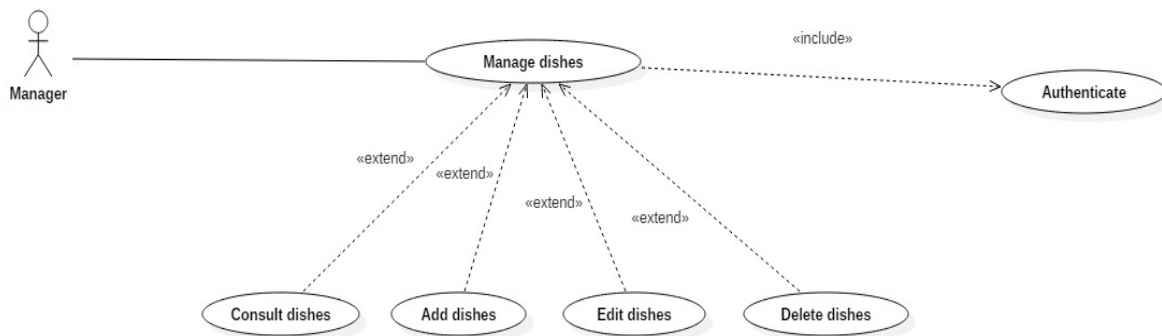


Figure 14: use case diagram " manage dishes "

2.4.5. Use case « Manage loyalty techniques »

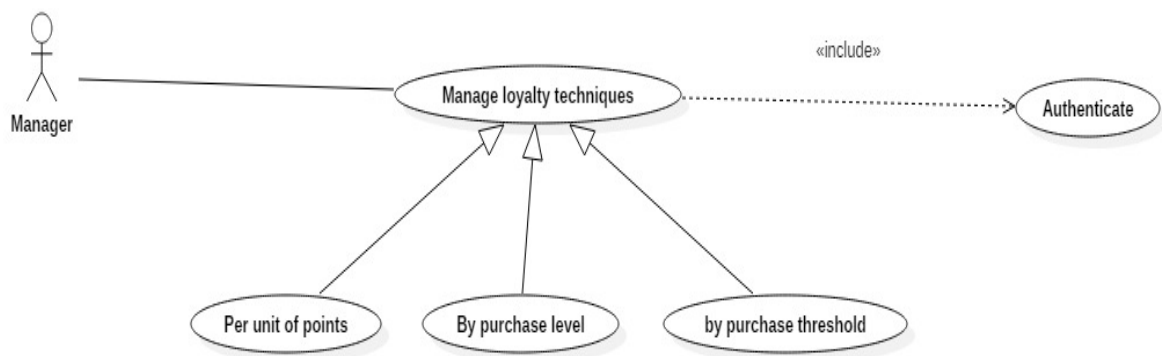


Figure 15: use case diagram " loyalty techniques"

2.4.6. Use case « Manage Platform »

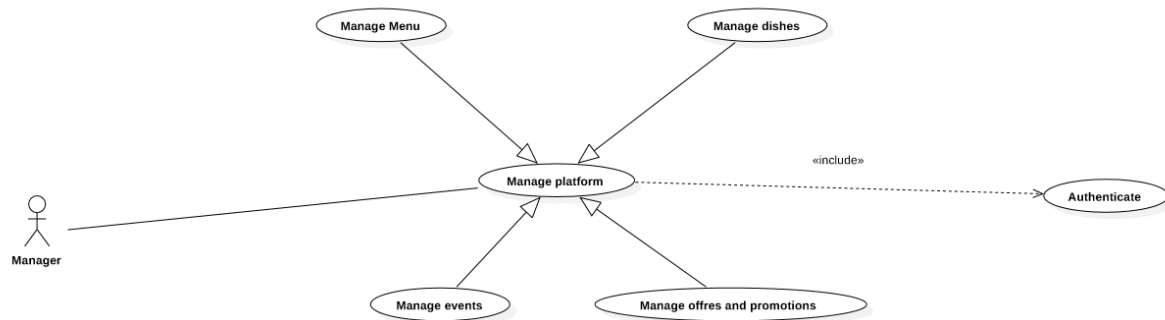


Figure 16: use case diagram "manage platform"

2.4.7. Use case « Manage booking »

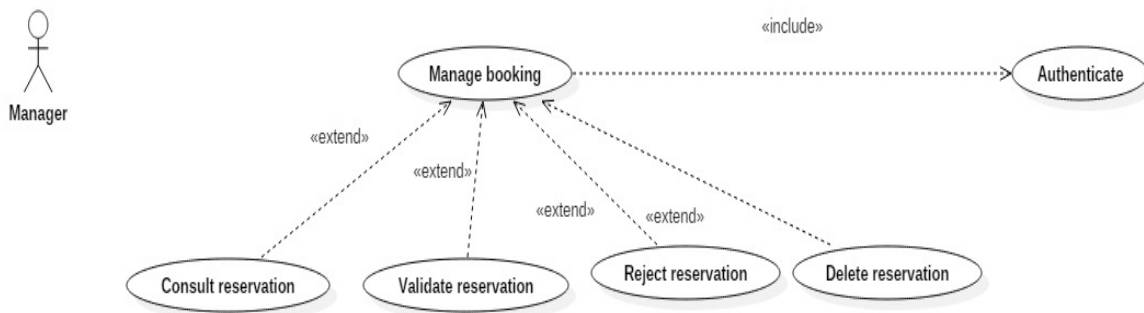


Figure 17: Use case « Manage booking »

2.5. Text description

2.5.1. Use case « Authenticate »

Title	Authenticate
Actors	Manager, customer, prospect, loyal customer.
Goal	This use case is used to connect to the application, and to ensure the security of the data.
Pre-condition	The user must have a user name and a password save in a good secure database.

Post conditions	The user accesses the application.
Scenario nominal	<p>This CPU starts when the user requests to access the application, the system displays an authentication window and the process continues as follows:</p> <ol style="list-style-type: none">1) The user enters his login.2) The user enters his password.3) The system checks the login of user.4) The system checks the password of user.5) The system displays a confirmation message.
Alternative scenario	<p>The alternative scenario AL A1 starts at the point 3 of the SN</p> <p>A1.1) the system informs the user that the login is incorrect</p> <p>A1.2) The user re-enter the login</p> <p>A1.3) the system will resume at point 4 of the SN</p> <p>The alternative scenario AL A2 starts at the point 4 of the SN</p> <p>A1.1) the system informs the user that the password is wrong</p> <p>A1.2) The user re-enter the password</p> <p>A1.3) the system resumes from the point 5 of the SN</p>
Exceptions	<p>The error scenario starts to point 3</p> <p>R1.1) the system blocks the user, the script ends in failure</p>

Table 1: Text description of use cases "Authenticate"

2.5.2. Use case « Add dish»

Title	Add dish
Actors	Manager.
Goal	This use case is used for adding dishes.
Pre-condition	the Manager is authenticated.
Post conditions	A dish is added.
Nominal scenario	<p>This CU begins when the Manager logs on to the application, the process continues as follows:</p> <ol style="list-style-type: none"> 1) The Manager clicks the button "« Add dish ». 2) The system displays the corresponding interface. 3) The Manager fill in the form for adding. 4) The Manager submits the form. 5) The system saves the entered information and a message of confirmation id displayed.
Alternative scenario	<p>A1: the wording of the dish already exists.</p> <p>The A1 sequence starts at the point 4 of the SN.</p> <ol style="list-style-type: none"> A1.1) the system informs the manager that the seized wording already exists. A1.2) the Manager pulled together a new wording of the dish. A1.3) the system resumes from the point 5 of the SN. <p>The A2 sequence starts at the point 4 of the SN.</p> <ol style="list-style-type: none"> A2.1) the system asks the Manager to fill in all fields. A2.2) Manager fill in all fields. A2.3) the system resumes from the point 5 of the SN.

Table 2: Text description of use case "Add dish"

2.5.3. Use case « Validate reservation »

Title	Validate reservation
Actors	Manager.
Goal	This use case is used for the confirmation of reservations.
Pre-condition	the Manager is authenticated.
Post conditions	A booking is confirmed.
Nominal scenario	<p>This CU begins when the Manager logs on to the application, the process continues as follows:</p> <ol style="list-style-type: none"> 1) the actor requesting access to the interface « reservations ». 2) The system displays the corresponding interface 3) The manager chooses the option « Show ». 4) The system displays the corresponding interface. 5) Manager valid booking.

Table 3: Text description of use cases “validate reservation”

2.5.4. Use case « Consult status order »

Title	View order status
Actors	Customer.
Goal	This use case is used to track the status of an order from a customer account.
Pre-condition	The client is already authenticated to place an order. An order is placed and the State value is assigned to one of the following: validated, rejected, pending, processing, dispatcher, completed.
Post conditions	The status of the order is displayed.
Scenario nominal	<p>This CU begins when the man client connects to the application, the process continues as follows:</p> <ol style="list-style-type: none"> 1) The client requests to access the interface « Orders history ». 2) The system displays the current status of the order.

Table 4: Text description of use cases “Check order status”

2.5.5. Use case « Make reservation »

Title	Make reservation
Actors	Customer.
Goal	to book a table at a given date and during a given time interval.
Pre-condition	The client is already authenticated to place an order that State one of these different values: validated, rejected, or pending.

Post conditions	A reservation is established.
Scenario nominal	<p>This CPU starts when the man client connects to the application, the process continues as follows:</p> <ol style="list-style-type: none">1) The client requests to access the interface « Booking ».2) The client allocates the desired booking date.3) The customer enters the desired booking time.4) The client allocates the number of people.5) The client validates this insertion.6) The system checks the date field booking7) The system checks the reservation time fields.8) The system checks if there is a field not filled.9) The system validates the booking.
Alternative scenario	<p>The alternative scenario A1 starts at point 6 of the SN</p> <p>A1.1) the system informs the customer that the date is not valid.</p> <p>A1.2) customer re-enter a correct date</p> <p>A1.3) the system will resume at point 7</p> <p>The alternative scenario A2 starts at point 7 of the SN</p> <p>A2.1), the system informs the customer that time is not valid.</p> <p>A2.2) customer re-enter time</p> <p>A2.3) the system will resume at point 8</p> <p>The alternative scenario A3 starts at point 8 of the SN</p> <p>A2.1) the system asks the customer to fill in all fields.</p> <p>A2.2) the customer fills all the fields</p> <p>A2.3) the system will resume at point 9</p>
Exceptions	<ol style="list-style-type: none">1) The system displays 'Required to enter empty field!'.2) The system displays "Invalid Date".3) The system displays "Invalid time".

Table 5: Text description of use cases “Make reservation”

2.6. Package of the use case diagram

The functional need of players for the future application seems quite complicated to us. To facilitate our work, we will cut the application into parts, so that it can be analyzed them separately. Each of these parts is a package.

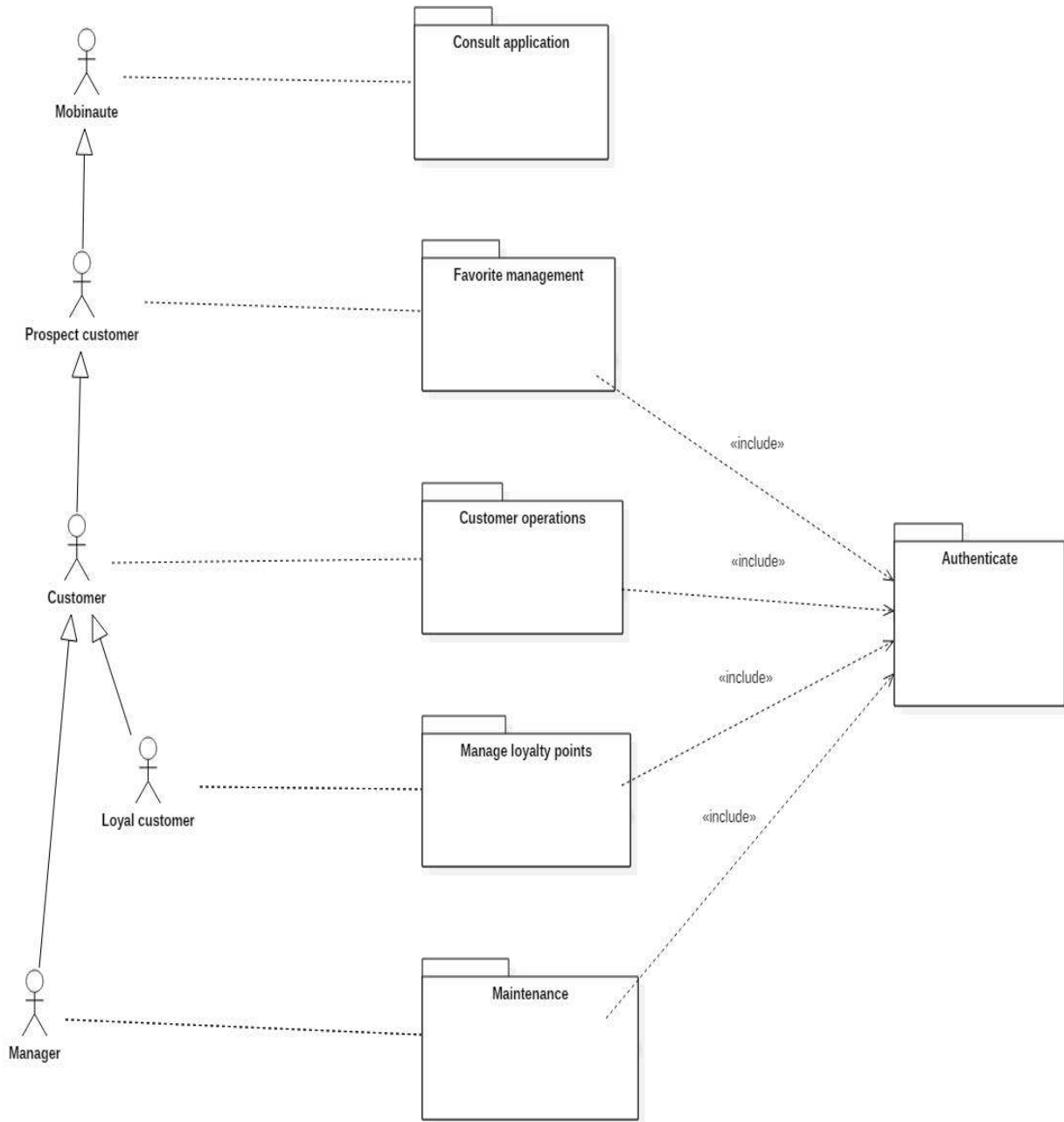


Figure 18: package of the use case diagram

3. Specification of technical requirements

In this part we will present the overall design of the architecture of our application, then we will present some prototyping of GUIs.

3.1. Design of the architecture of the project

3.1.1. The MVC pattern

The model-view-controller (abbreviated MVC, Model-View-Controller) is an architecture and a design method which organizes the interface man-machine (IHM) of a software application. This paradigm divides the IHM in a model (data model), a view (presentation, user interface) and a controller (control logic, event handling, synchronization), each with a specific role in the interface.

3.1.2. Architecture

3.1.2.1. The model

The model represents the behaviour of the application: processing of data, the database interactions, etc. It describes or contains the data manipulated by the application. It manages this data and ensures their integrity. In the typical case of a database, this is the model that is contained. The model provides methods to update these data (insertion, deletion, change of value). It also offers methods to retrieve the data. The results returned by the model are devoid of any presentation.

3.1.2.2. The views

The view is the interface with which the user interacts. Its first task is to present the results returned by the model. Its second task is to receive all the user actions (mouse click, selection of an entry, buttons, etc...). These events are sent to the controller. No processing is performed to the view, just display the results of the treatments performed by the model and interact with the user. It can be designed in HTML, or any other « language » of presentation.

3.1.2.3. The Controller

The controller supports the management of synchronization events to update the model or the view and synchronize them. The controller is the element that will use the data to send to the view. Its role is to retrieve information, process them according to the parameters required by the view (user, example: display the latest articles), and then to return to the view the data to be displayed. The controller can then instantiate objects (User class, class Articles) which will send queries to the database or any other form of storage. The transmission in full view of the retrieved data is usually done using a Template by inserting data into an HTML file.

3.1.3. Behavior

The processing of a request from a customer takes place according to the following steps:

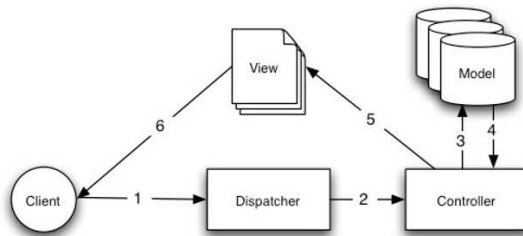


Figure 19: MVC architecture

- 1) The request arrives on the dispatcher, a script that is responsible for calling the right Controller.
- 2) Request the address "www.site.com/category/viewAll", it is directed to the "viewAll" of the "category" controller function.
- 3) The controller asks for the products list of the model "category".
- 4) The model returns this list to the Controller.
- 5) The controller sends this list to the view.
- 6) After you generate code for XHTML, the view returns to the client.

3.1.4. Advantages et disadvantages:

An advantage brought by this model is the clarity of the architecture that it imposes. This simplifies the task of the developer who would attempt to perform maintenance or an upgrade on the project. Indeed, the change of treatment does not alter the view. For example, you can go from a database of SQL to XML type by simply changing the processes of interaction with the base, and the views are not affected. The MVC approach brings real **benefits**:

- ✓ A clear and effective design due to the separation of the data from the view and the controller.
- ✓ Saving maintenance time and evolution of the site.
- ✓ More flexibility to organize the development of the site between different developers (independent of the data display (Web design) and actions).

Disadvantages:

The major drawback of the MVC pattern is visible in the realization of small projects, of small websites. Indeed, the separation of the different layers requires creating more files (one file per each layer), it is not very interesting to use this system in this case here [1].

3.2. Interface prototyping

Prototyping is to realize a document presenting the proposed structure of the future application, it represents a further step in the design of the application that allows you to graphically view the organization of information within the pages.

To realize these interfaces, we used "Balsamiq" as a tool, it makes possible to easily create prototypes of HMI electronic desktop, web, smartphone.

3.2.1. Front office part

3.2.1.1. « Home » interface

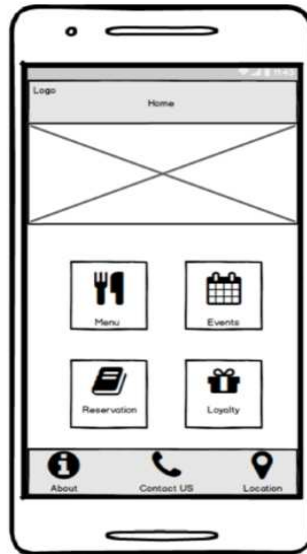


Figure 20: Prototyping of the interface "Home"

3.2.1.2. « Menu » interface



Figure 21: Prototyping of the interface "Menu"

3.2.2. Back office part

3.2.2.1. « Dishes » interface

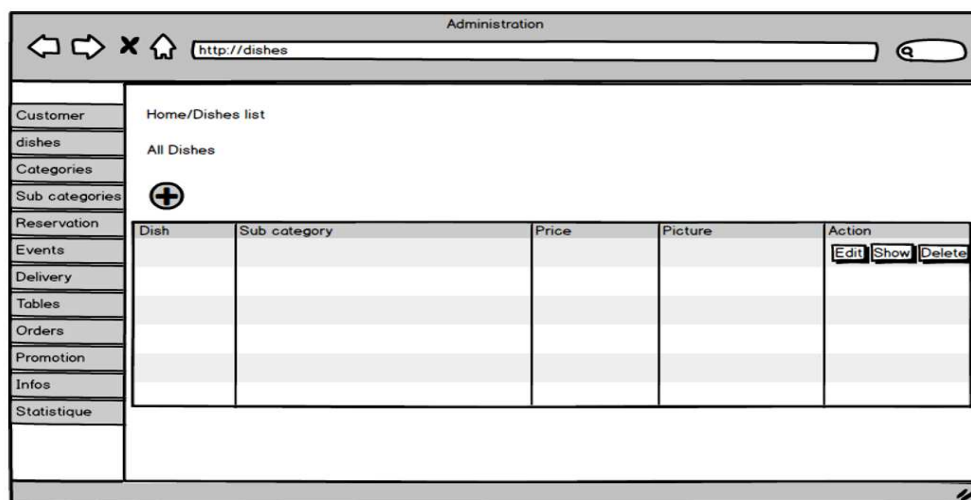


Figure 22: Prototyping of the interface "dishes"

3.2.2.2. « Statistics » interface

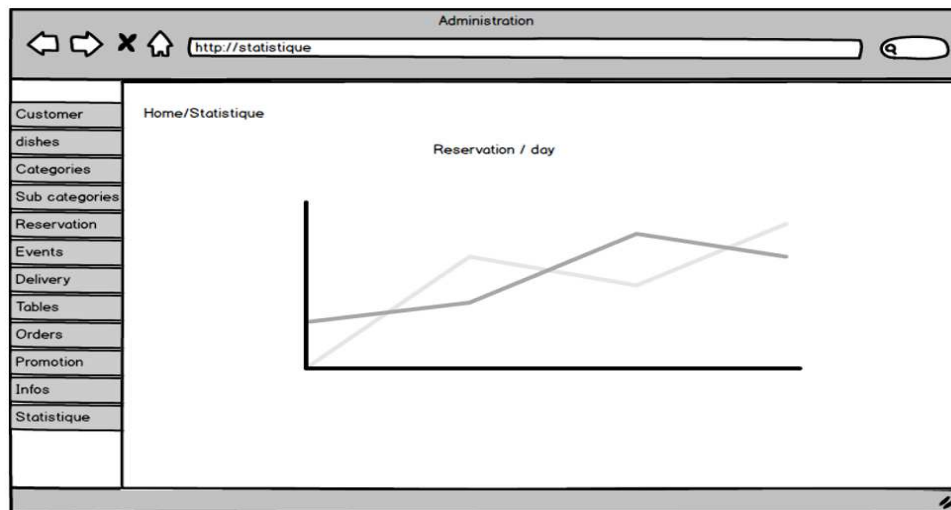


Figure 23: Prototyping of the interface "Statistics"

4. Conclusion

After presenting the functional requirements and technical requirements of the system to be developed through identification of actors, context of use case diagram model building and refinement with the corresponding textual descriptions. We are set and ready to start the design phase in the next chapter.

Chapter 3: Conceptual Study

Chapter 3: Conceptual Study	30
1. Introduction	31
2. Static study.....	31
2.1. The class diagram	31
2.2. Optimization of semantics by the l'OCL.....	33
2.2.1. what is the OCL	33
2.2.2. Optimized class diagram	33
2.3. Interpretation of the class diagram	35
2.4. Package class diagram	35
2.5. Deployment diagram	37
3. Dynamics study	38
3.1. Activity diagram	38
3.1.1. Authentication	38
3.1.2. Add reservation	39
3.1.3. Manage order	40
3.2. Sequence diagram	41
3.2.1. Sequence diagram « Edit dish »	41
3.2.2. Sequence diagram « Add dish »	42
3.2.3. Sequence diagram « Manage order »	43
3.2.4. Sequence diagram « Consult statistics »	44
3.3. State transition diagram	44
4. Conclusion	44

1. Introduction

The chapter will allow us to have a clear idea about our field of study through a detailed design. Indeed, we present a static model by the class diagram, and a dynamic model that focuses on the organization of the scenarios in sequence diagram, we refer to the description of the lifecycle of a business.

2. Static study

2.1. The class diagram

The class diagram represents the classes involved in the system, is a static representation of the elements of the system and the different relationships between them.

A class is an abstract concept that allows representing all entities of a system, it allows modelling a program and thus breaking down a complex task into several simple steps [W4].

This class diagram represents an essential part of the work dedicated to this project. We have the «**Event Type**» class that allows identifying the possible types of events: the "Value" attribute takes the value 1 if the event is organized by the customer and is set to 0 if it is organized by the restorer. An «**Event**» can be an 'organized Event' or a 'requested Event' which is organized by the client and is linked to the class «**booking**».

The user can be the «**Manager**», a «**lead customer**» that can add several «**Dish**» to his «**Favorite**» dishes, and if it makes an order, a reservation or it organizes an event it will become a «**Customer**» and it can earn points and become a «**Loyal customer**».

A «**Customer**» can pass one or more commands «**Order**» and an order is made through a single client. Each order results in an invoice which will be created and delivered.

The customer chooses the mode of payment used it can be with the PayPal «**PayPal**» portfolio, with Fidelity points «**Loyalty points**» or species at home «**Home**».

A «**Customer**» can pass one or more «**booking**» reservations and a reservation is booked through a single client.

Each «**dish**» belongs to a «**Category**» and a category contains one or more dishes.

The dish may have some additional «**Assortment**» dishes.

The customer can put a «**Review**» note on several dishes.

«**Caddy**» is a non-persistent class, that is, it does not exist in the database.

Each dish must have a picture.

Each assortment must have a picture.

The event organized by the Restorer «**Event Organized**» has one or more images.

The class «**Resto Picture**» is created to facilitate the creation of animations slider.

Chapter 3 : Conceptual Study

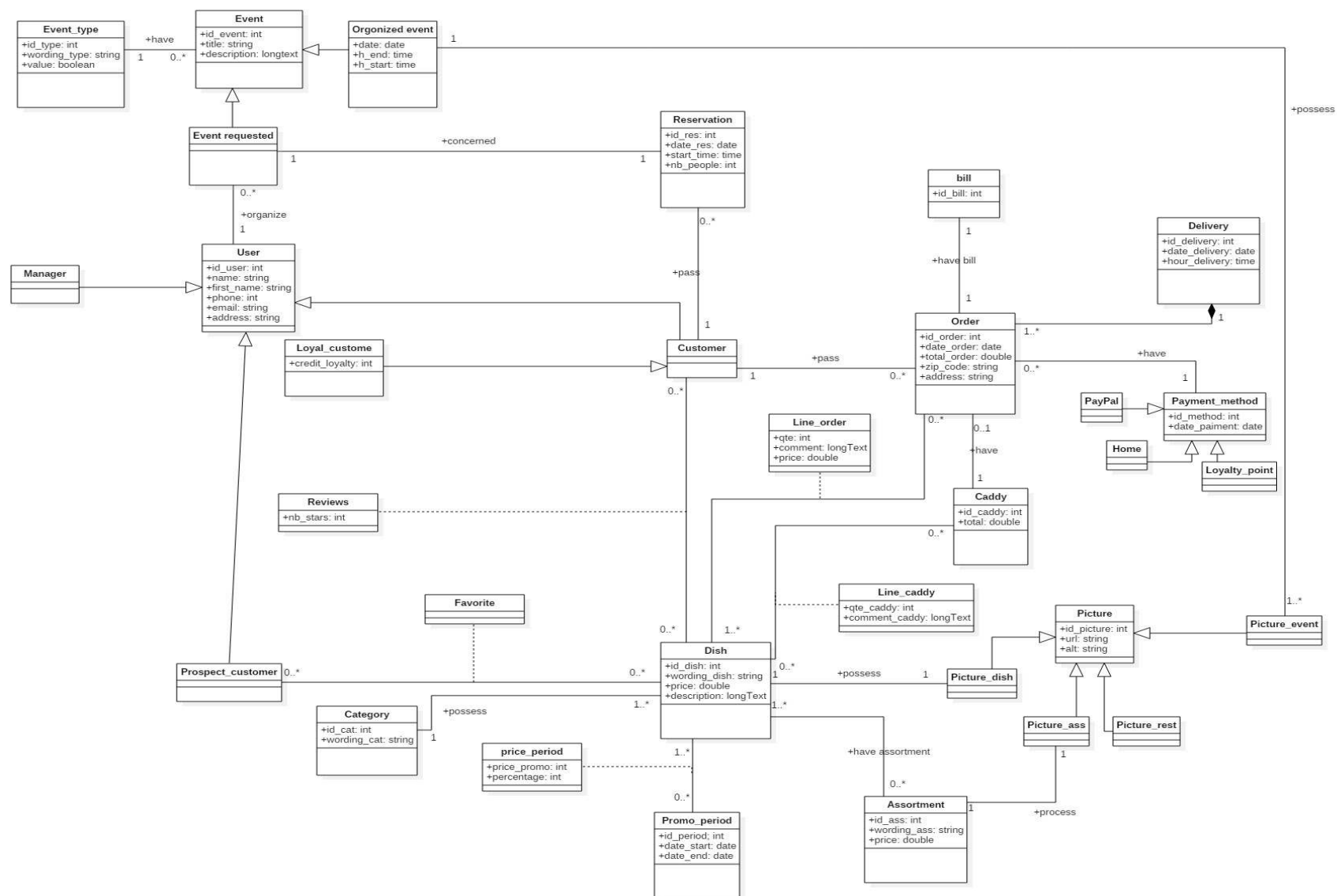


Figure 24: the class diagram

2.2. Optimization of semantics by the l'OCL

2.2.1. what is the OCL

Object Constraint Language (OCL) is a formal language of object-oriented constraints with a syntax, a grammar, a semantics (Can be manipulated by a tool). It is applied on UML diagrams. The class diagram does not make it possible to detail all the constraints on the relations between the classes. For this reason, we add the OCL semantics which mainly allows us to express two types of constraints on the state of an object, A set of objects:

Invariants that must be permanently respected.

Pre and post conditions for an operation: pre-condition must be checked before execution and post-condition must be checked after execution.

2.2.2. Optimized class diagram

Chapter 3 : Conceptual Study

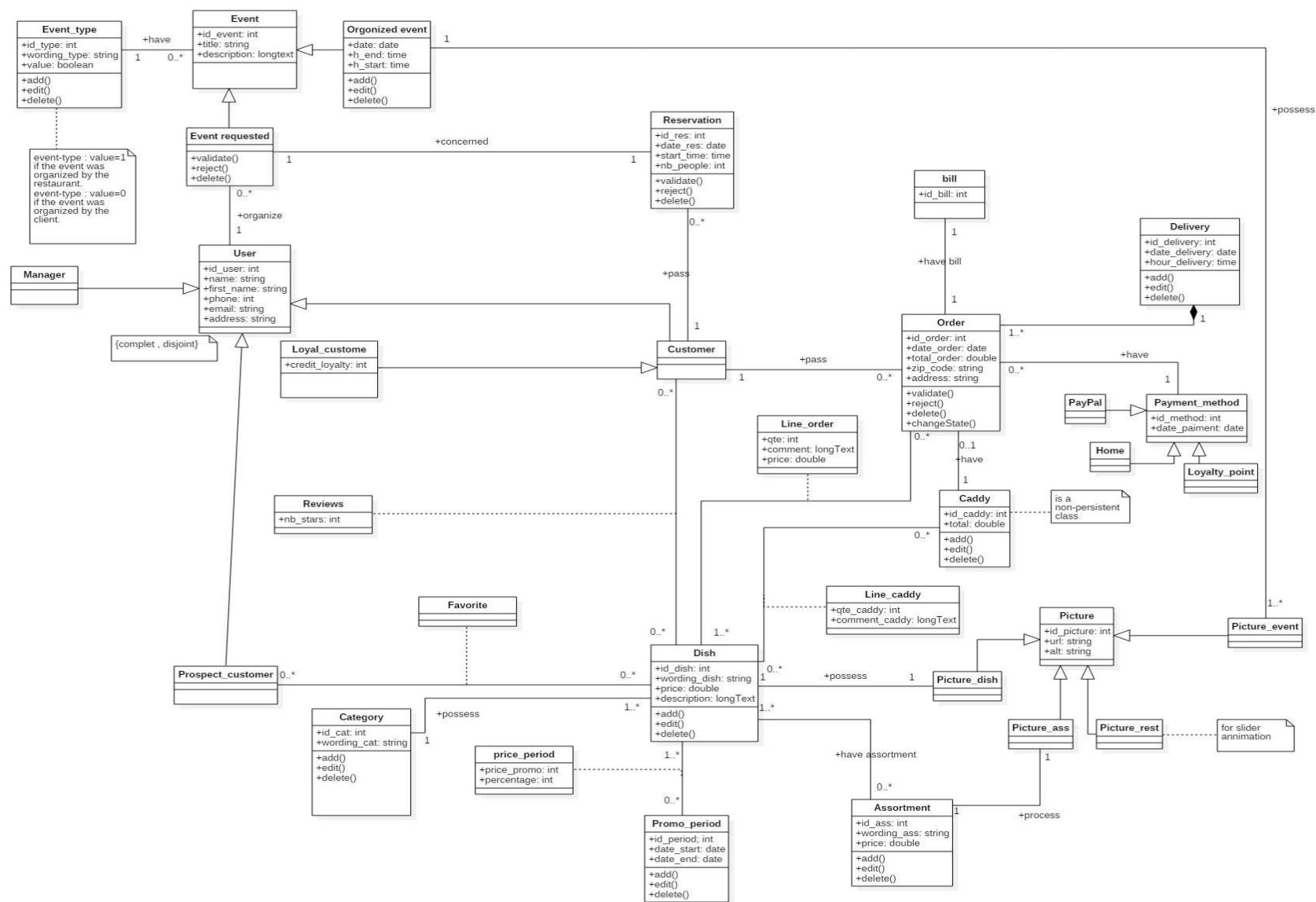


Figure 25: Class diagram with "OCL

2.3. Package class diagram

After identifying the different classes, we proceed to grouping in packages, organizing these packages to present a good overall modularity of the system and so a good distribution of the development.

For this project we have used functionality as the criteria for decomposition in layers, i.e. we have broken down our classes in package diagram according to the most significant functionalities, while respecting the strong cohesion between the classes within the same package and low coupling in multiple packages together [W5].

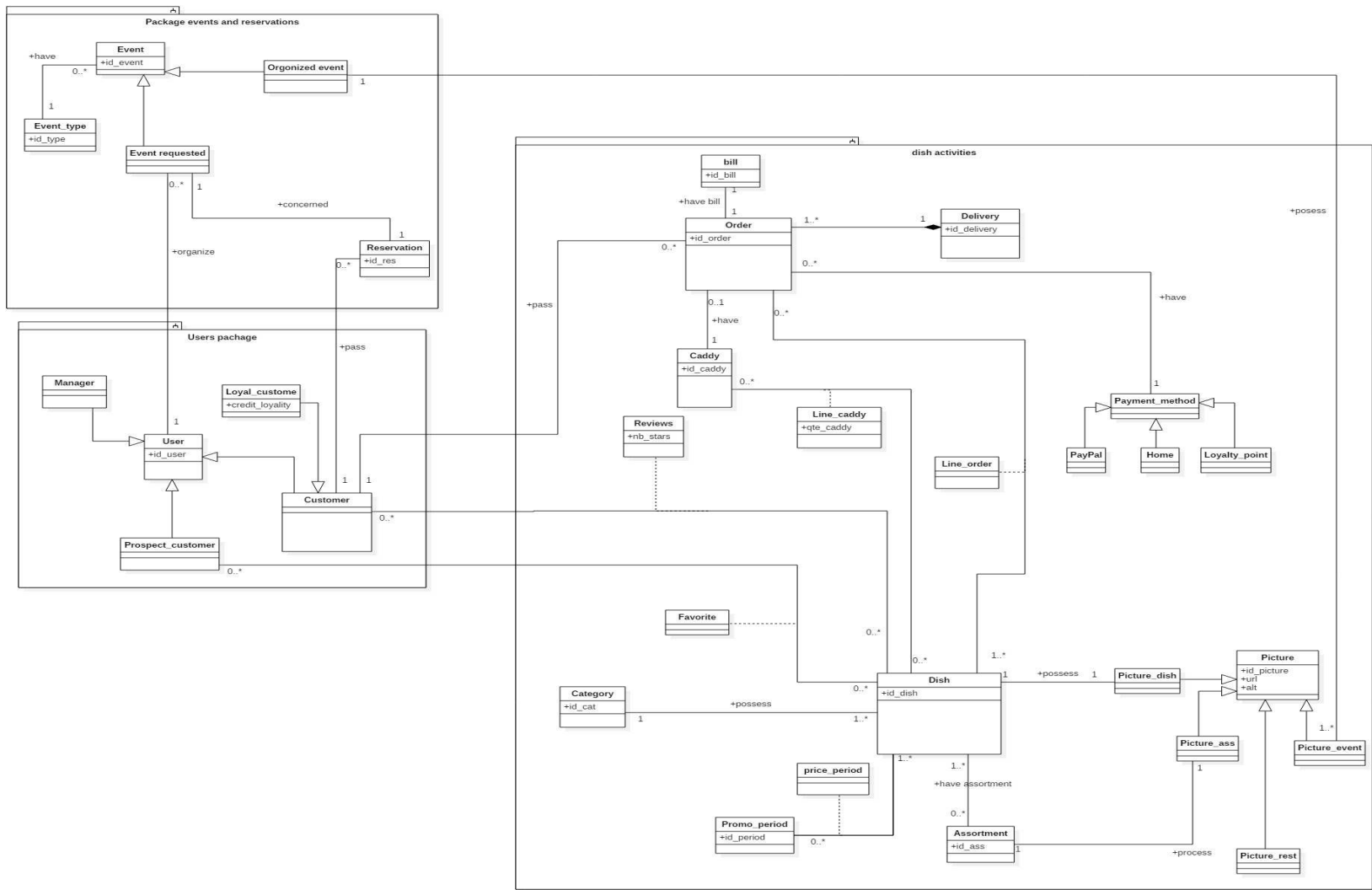


Figure 26: Package class diagram

2.4. Deployment diagram

The deployment diagram is an important step in the development of a system. It describes the physical location of the hardware used and the distribution of the components on this hardware. It also displays the relationship between the software and hardware of the system components on one hand, and the physical distribution of the treatment, on the other.

The runtime environment or material resources are called « **Nodes** ». Parts of a system that is running on a node are called « **Artifacts** » [W6].

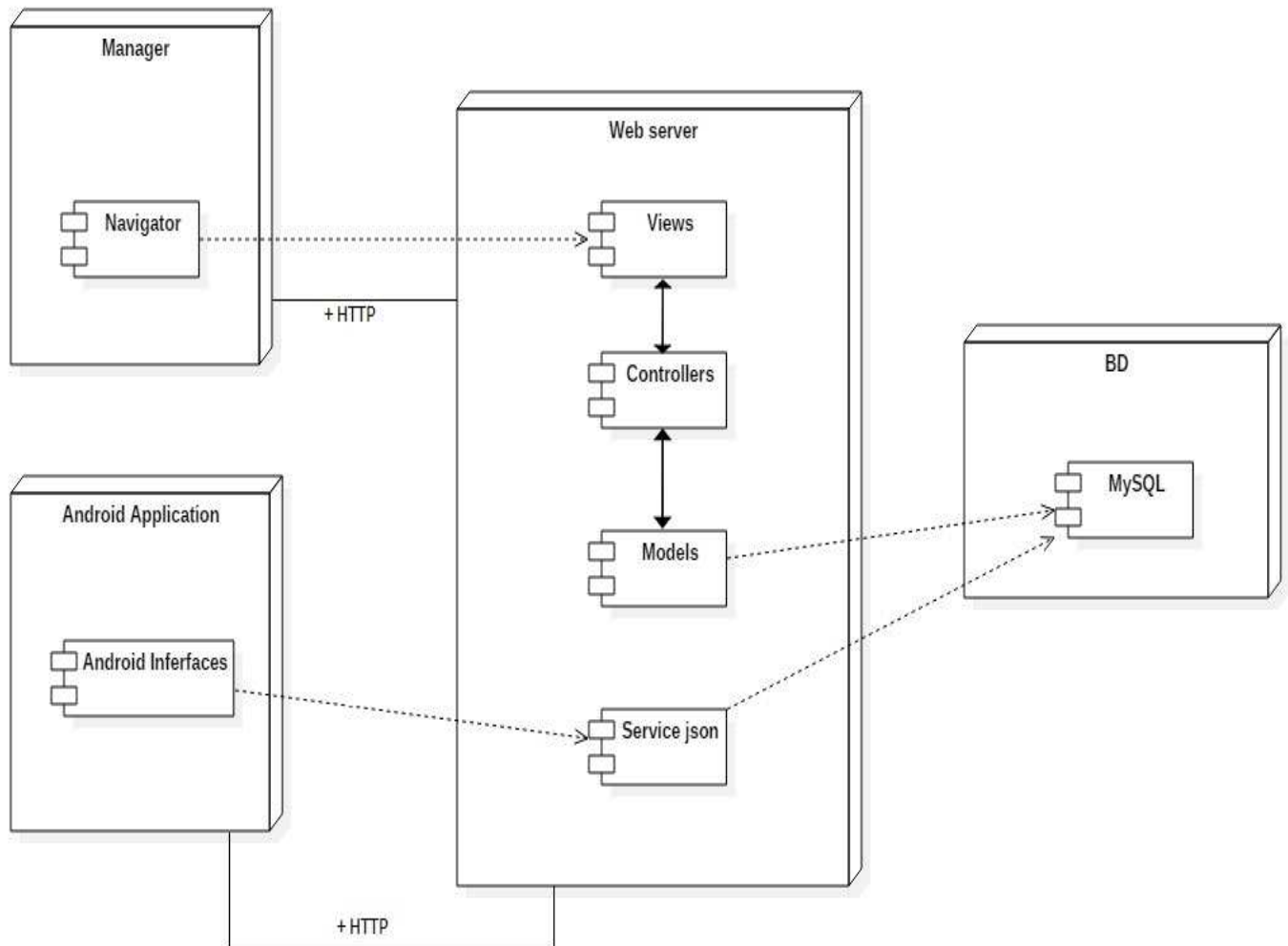


Figure 27: Deployment diagram

3. Dynamics study

3.1. Activity diagram

Activity diagram represents graphical displays of workflows of activities and actions in the system. On the one hand to consolidate a use case specification, on the other hand to design a method.

3.1.1. Authentication

The diagram below represents the possible process for a user to authenticate the entry of a session.

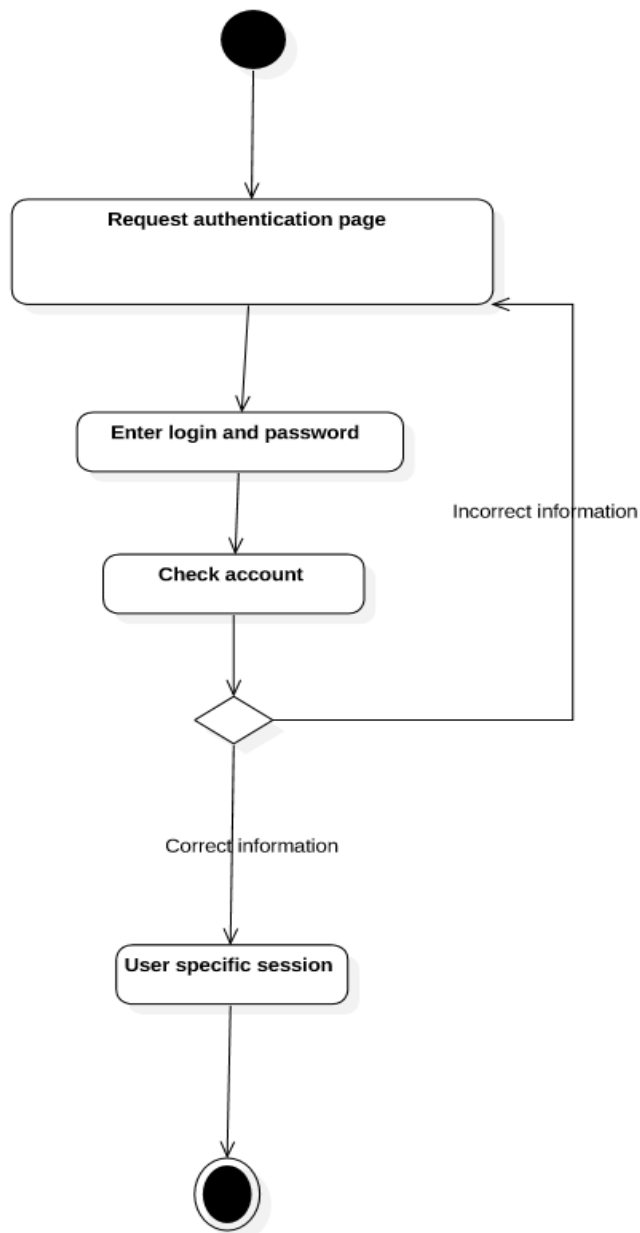


Figure 28: Activity diagram "Authentication"

3.1.2. Add reservation

The diagram below represents the possible process for the manager to create a new customer booking.

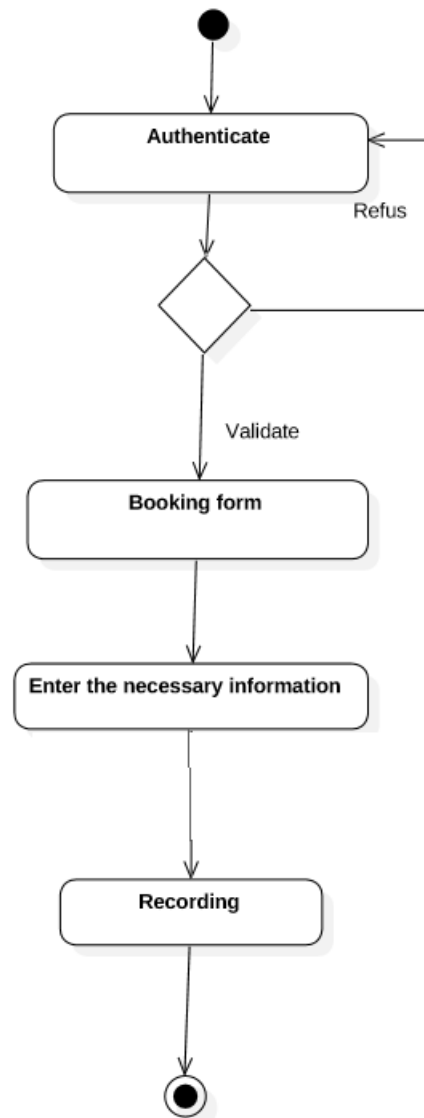


Figure 29: Activity diagram " Add reservation "

3.1.3. Manage order

The diagram below represents the possible process for the manager to manage orders sent by customers.

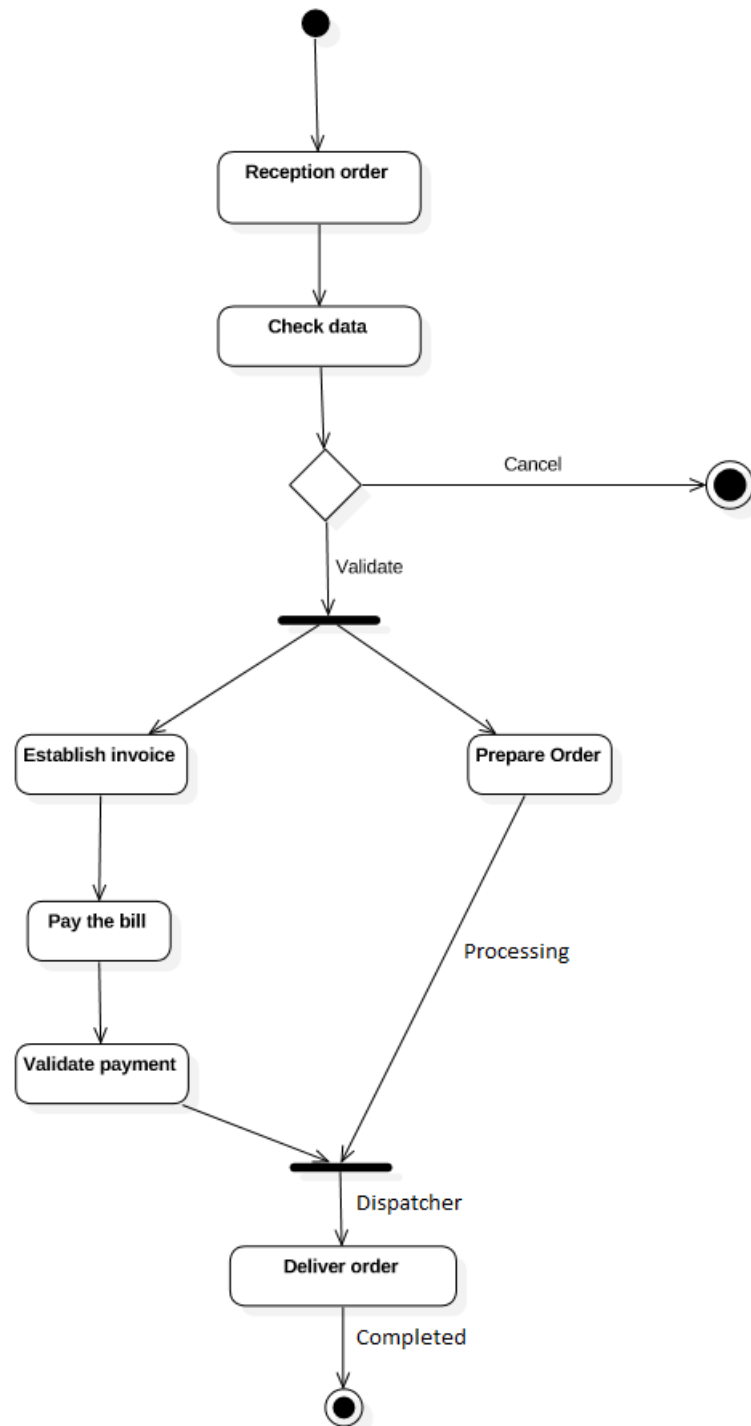


Figure 30: Activity diagram " Manage order "

3.2. Sequence diagram

The sequence diagram can represent collaborations between objects from a temporal point of view; it is the emphasis on the chronology of the transmissions of messages. The main information contained in a sequence diagram is messages between the lines of life, presented in chronological order. The goal is to describe how to place the shares between actors or objects.

3.2.1. Sequence diagram « Edit dish »

The manager accesses the interface «list of dishes» which is an array that contains all the dishes that will be displayed. Subsequently he chooses a dish from the list and click « Edit ». An editing interface opens and modifies the data and then clicks on the « Save » button. Then the changed data will be verified, if the wording of the dish already exists or the fields are not filled out an error message will be displayed, otherwise, data will be sent and confirmation message is displayed.

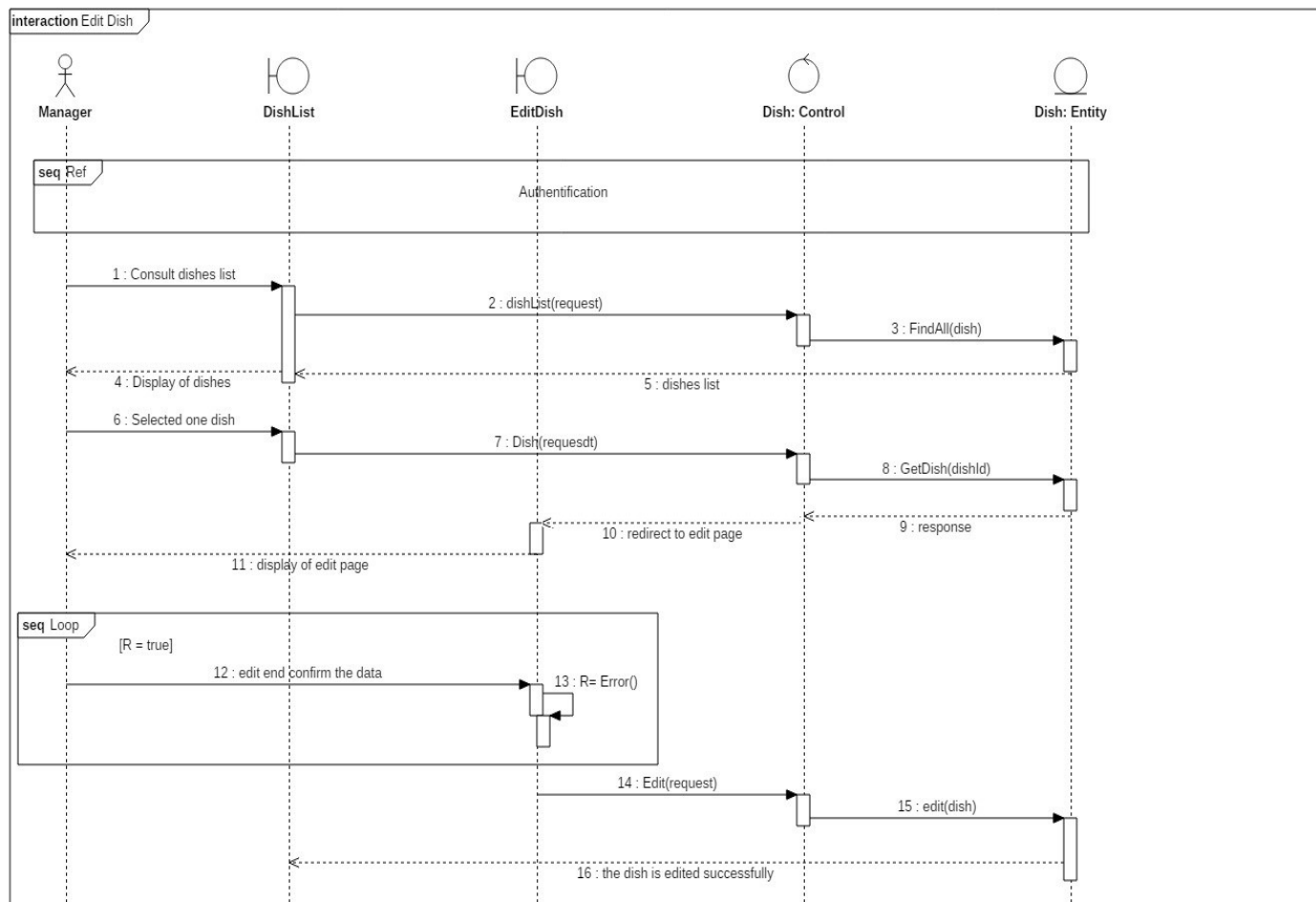


Figure 31: Sequence diagram « Edit dish »

3.2.2. Sequence diagram « Add dish »

In order to add a dish, the Manager must fill in a form then clicks on the « Save » button. Subsequently the data sent will be verified, If the wording of the dish exists already or several fields are empty an error message will be displayed, otherwise, data will be sent and confirmation message is displayed in the interface « Dish List ».

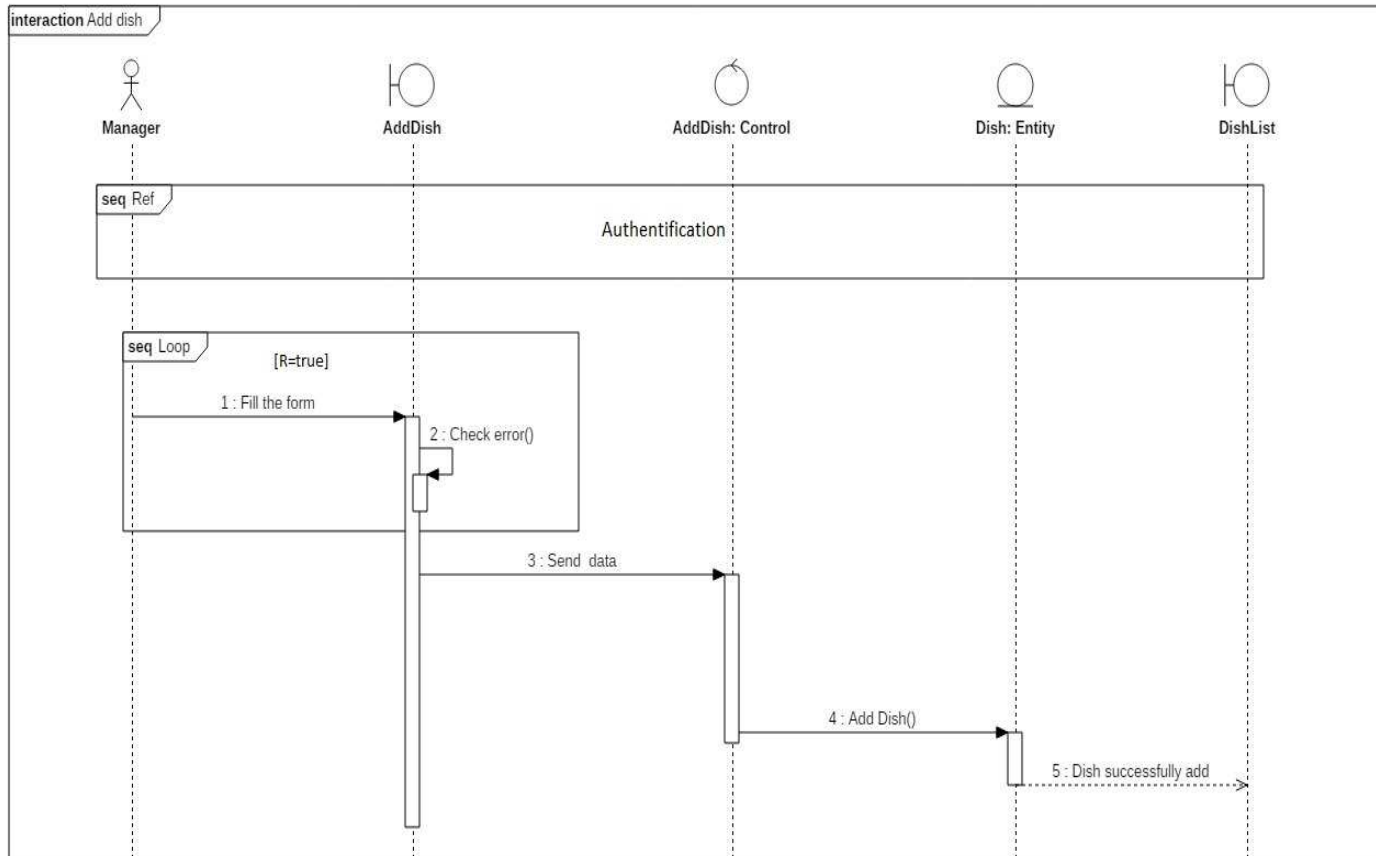


Figure 32: Sequence diagram « Add dish »

3.2.3. Sequence diagram « Manage order »

The manager accesses the interface « **order list** », an array that contains all the orders will be displayed. Subsequently he chooses an order from the list and click on « Show ». Then, he will be directed to the page « order details » that contains all the information needed for this order. If the later is not yet validated or rejected the « Validate » and« Reject » buttons will be displayed and the administrator can validate or dismiss it by clicking on these buttons. If one of these two actions is valid the two buttons will be no longer visible. If case it is valid three buttons will appear so as to be able to change the status of the order.

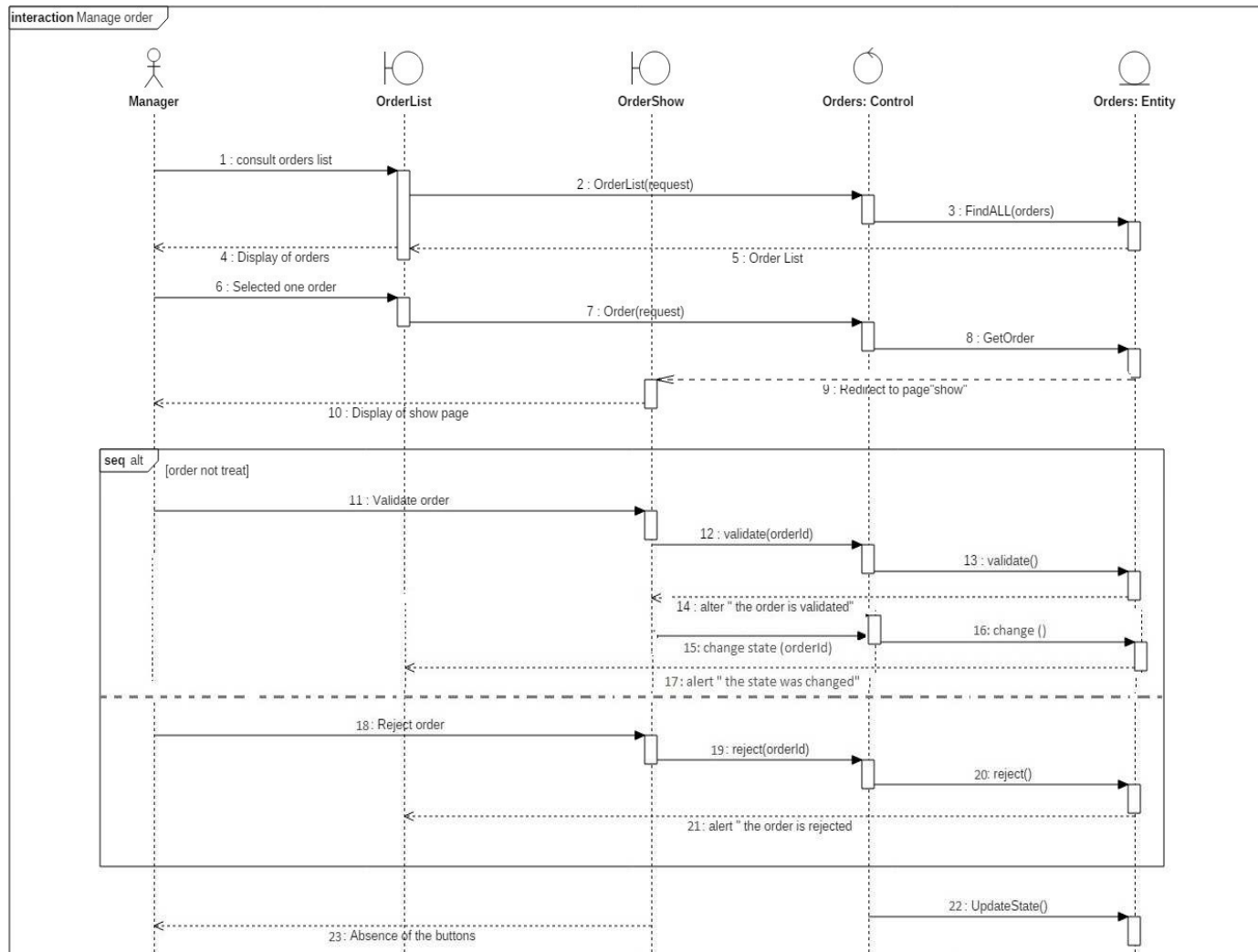


Figure 33: sequence diagram « manage order »

3.2.4. Sequence diagram « Consult statistics »

The administrator chooses the type of statistics he wishes to consult, and subsequently the corresponding page will be displayed.

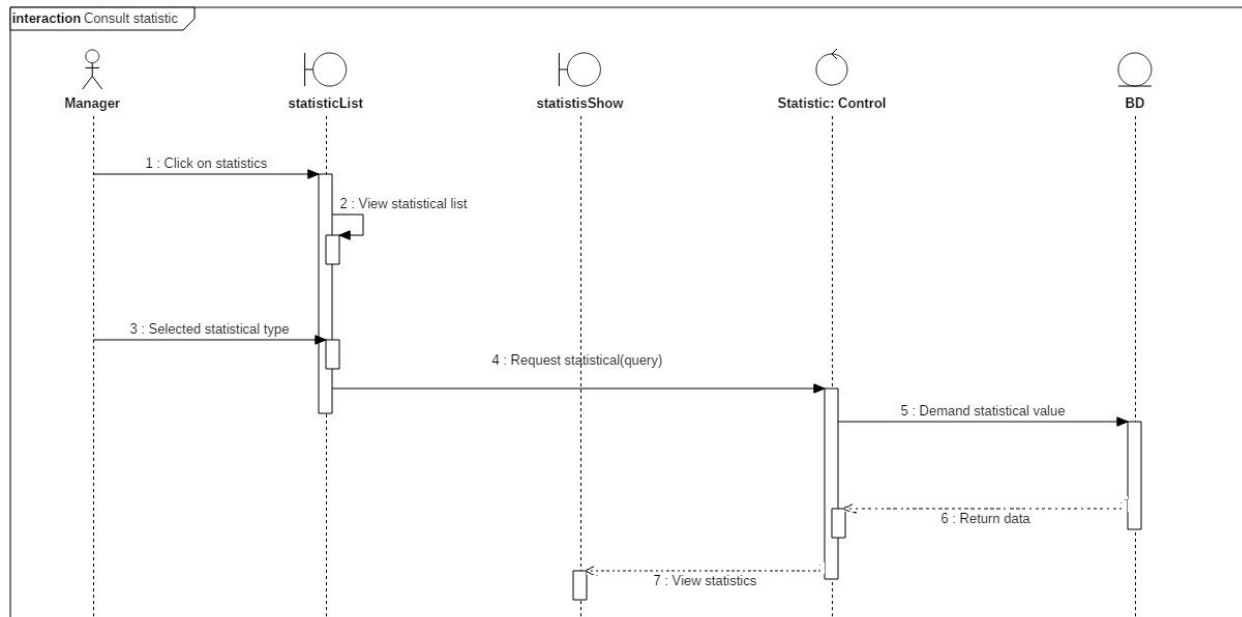


Figure 34: Sequence diagram « Consuler Statistics »

3.3. State transition diagram

The State transition diagram describes the dynamic behaviour of an entity it presents the state changes to an object or possible component. It is a graph composed by a set of nodes (State) and a set of arcs (Transition).

In our case we will describe state changes possible for the user of mobile application [2].

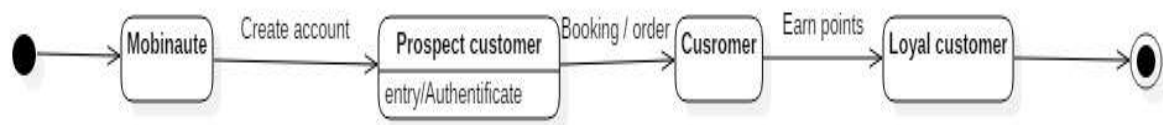


Figure 35: State transition diagram

4. Conclusion

In this chapter, we looked at the overall design of the application by drawing the class diagram and sequence diagram construction. We turn now to the representation of the development environment of the hardware and software that were used for the realization of this application, the logical and physical data, the principle model of ergonomics, the main final interfaces of the application, as well as evaluation tests.

Chapter 4 : Implementation

Chapter 4 : Implementation.....	46
1. Introduction	47
2. Development environment.....	47
2.1. Hardware environment.....	47
2.2. Software environment.....	47
3. Principle of ergonomics	49
3.1. Template « Back-office » part.....	49
3.2. Choice of colors	49
3.3. The web page structure.....	50
4. Aspects of physical model.....	51
5. Main interfaces	58
5.1. Back-office part.....	58
5.1.1. Connection Interface	58
5.1.2. « Home page » interface.....	59
5.1.3. « order List » Interface	60
5.1.4. « Details order » Interface	60
5.1.5. « Bill » Interface	62
5.1.6. Statistics interface « turnover by zip code ».....	62
5.1.7. Statistics interface « number of orders per month »	63
5.1.8. Statistics interface « number of booking per day ».....	63
5.1.9. Statistics interface « Dishes ordered by month »	64
5.1.10. « dish list » Interface	64
5.1.11. « Details dish » Interface	65
5.1.12. « Add promotion » Interface.....	66
5.1.13. « loyalty system » Interface	67
5.2. Front-office part	67
5.2.1. Splach interface	67
5.2.2. Home interface	68
5.2.3. Menu interface	68
5.2.4. Details dish interface	69
5.2.5. Delivery interface	69
5.2.6. Sign in interface	70
5.2.7. Account interface	70
5.2.8. Order interface.....	71
5.2.9. “Manage loyalty points” interface.....	72
5.2.10. “Pay with PayPal” interfaces.....	73
6. Conclusion	73

1. Introduction

In this chapter and after detailed design of the application we will study the different existing technologies that can be used for implementing our project.

First of all, let us describe the hardware and software used in the development environment, as well as the technologies employed. Then we are going to represent the physical and logical database model utilized. Finally, we will present some screenshots of the interfaces of our application.

2. Development environment

2.1. Hardware environment

The development of the application was made on a computer having the following characteristics:

Brand	Asus
Processor	Intel® Core™ i5-6300HQ CPU @ 2.30 GHz- 6 Mo cache
RAM	12 Go DDR4
Graphics card	NVIDIA GeForce GTX 960M 4 Go dedicated
Hard drive	1To
Operating system	Windows 10 Professional 64 bits

Table 6: The characteristics of the used machine

2.2. Software environment

For the implementation of this application we used different software and programming languages.

✓ **Symfony Framework:**



Symfony The project was planned to be designed with a Symfony Framework which permits significant progress for powerful and fast development. It allows it to be structured with a clear and maintainable code.

Symfony is a Framework developed by the French company SensioLabs PHP. It provides 1A methodology (conventions of writing and organizing, discipline of product code, MVC), tools (GRUD, Generation of Amin, plugins/bundles). It integrates some great features and tools that are not too much like the Doctrine ORM and Template engine « Twig ». [W7]

✓ **PHPStorm:**



PHPStorm is an IDE developed by JetBrains team, which allows developers to become more efficient and productive in their projects and to increase the quality of their codes, it offers better code completion and on-the-fly error prevention.

It is a powerful tool and oriented towards professionals, fits under different operating systems such as: Windows, Linux, GTK, OSX...? [W8].

✓ **AndroidStudio:**



Android Studio is an IDE developed by Google to create Android applications; it integrates an abundance of tools and features to design quality applications which allow either auto completion or code parsing.

It offers design projects that adapt to several types of Android devices such as tablets, smartphones and TV. It also offers the possibility to build multilingual applications [W9].

✓ **Genymotion**



Genymotion is an emulator to test Android applications without a dedicated device. It is the fastest emulator and works with Virtual Box. This emulator makes it possible to download different models of devices via the interface of the program [W10].

✓ **WampServer**



WampServer is a platform for Web development for dynamic Web applications. It allows for both local and online deployment. It contains a web server **Apache**, a scripting language **PHP**, MySQL for database management system and **PHPMyAdmin** for database administration [W11].

✓ **Adobe Photoshop**



Adobe Photoshop is a graphics software, for the editing and correction of digital or scanned images developed by Adobe Systems Incorporated. It is among the best photo editing softwares [W12].

✓ **HTML5**



HTML5 (**H**yper**T**ext **M**arkup **L**anguage), is the basic language used to create websites, it is inspired by the XML and works on the principle of the nested tags and it is used to structure and prioritize the formatting of a web page [W13].

✓ CSS3



CSS3 (**C**ascading **S**tyle **S**heets), is a language that allows to format the HTML or XML files through appearance as the colors, borders, fonts... and investment such as the width and height properties... [W14].

✓ JavaScript



JavaScript is a language based on the ECMAScript standard object oriented programming. It allows to improve the HTML language allowing to execute commands on the customer and d boost web pages [W15].

3. Principle of ergonomics

3.1. Template « Back-office » part

A Template makes it easier to design a website and its update, both in terms of content and presentation.

In our project we have adopted a Bootstrap Template and changed it so that the latter comply with our needs.

3.2. Choice of colors

Colors are therefore a key element in the design of our website but they also represent and emphasis the brand identity. Colors must bring coherence and harmony to the visual world in order to make the site more comfortable to use and improve the interest of targeted users.

In our case colors are imposed by the customer and the following link shows the selection made: <http://www.theoasisstalbens.co.UK/> :



Yellow gives an impression of heat and light. It is the color of good humour and joy of living. It is used to represent the agri-food sector because it increases the viewer's appetite.



Orange is associated with energy and action. It evokes the Sun, heat, fire, light, and autumn. Orange values communication and creativity. It transmits joy, fun and optimism. Its multiple features allow it to adapt to e-commerce, communication or mobility.



It is a neutral color that, when used on a large scale, gives an impression of blandness, without special attraction. It is used to represent the culinary field (chocolate and coffee) [W16].

3.3. The web page structure

✓ Sidebar

The sidebar listed to the left of every page contains links for quick access to different features.

✓ Header

The header is the top of a web page containing the photo of the owner placed on the left which leads to his profile and a notification icon indicated the existence of new users of the Android app.

A clickable icon on the right allows the manager to be disconnected.

✓ Content

Our web page encloses content that is placed in the Center. This is the content that the administrator wants to visit.

✓ Footer

It is the section that is situated at the bottom of the page which shows copyright sign.

4. Aspects of physical model

Assortment

Column	Type	Description
Id	int(11)	Assortment identifier
picture_id	int(11)	The image identifier corresponding to this assortment
wordingAss	varchar(255)	Assortment name
Price	Double	Assortment price

Table 7: table assortment

Bill

Column	Type	Description
id	int(11)	Bill identifier
order_id	int(11)	The order identifier corresponding to this bill
date_bill	datetime	The date when the bill was created

Table 8: Bill table

Category

Column	Type	Description
Id	int(11)	Category identifier
wordingCat	varchar(255)	Category name

Table 9: Category table

Configuration

Column	Type	Description
name	varchar(255)	The configuration name
value	varchar(255)	Identifier the type per a value

Table 10: Configuration table

Customer

Column	Type	Description
Id	int(11)	Customer identifier
user_id	int(11)	User identifier

Table 11: Customer table

Prospect Customer

Column	Type	Description
Id	int(11)	Prospect customer identifier
user_id	int(11)	User identifier

Table 12: Prospect customer

Delivery

Column	Type	Description
Id	int(11)	The delivery identifier
dateDelivery	Date	Date of delivery
hourDelivery	varchar(255)	Hour of delivery

Table 13: Delivery table

Dish

Column	Type	Description
Id	int(11)	The dish identifier
category_id	int(11)	The category identifier corresponding to this dish
picture_id	int(11)	The picture identifier corresponding to this dish
wordingDish	varchar(255)	The dish name
price	Double	The dish price
description	Longtext	The dish description

Table 14: Dish table

Dish_assortment

Column	Type	Description
dish_id	int(11)	The dish identifier
assortment_id	int(11)	The assortment identifier

Table 15: Dish assortment table

Eventorganized

Column	Type	Description
id	int(11)	The event organized identifier
type_event_id	int(11)	The identifier of the type corresponding to this event
title	varchar(255)	The event title
date_event	Date	The event date
startTime	Time	The start time of the event
endTime	Time	The end time of the event
description	Longtext	The description of the event

Table 16: event organized table

Eventrequested

Column	Type	Description
Id	int(11)	The identifier of the event requested
type_event_id	int(11)	The identifier of the type corresponding to this event
reservation_id	int(11)	The identifier of the reservation corresponding of this event
Title	varchar(255)	The event title
Description	longtext	The description of the event
View	tinyint(1)	The request seen or no
Validate	tinyint(1)	The request validate or no

Table 17: event requested table

Favorites

Column	Type	Description
user_id	int(11)	The identifier of user
dish_id	int(11)	The identifier of dish

Table 18: Favorites table

Fos_user

Column	Type	Description
Id	int(11)	The identifier of user
Username	varchar(180)	The user name
username_canonical	varchar(180)	The user name canonical
Email	varchar(180)	The email of user
email_canonical	varchar(180)	The email canonical of user

Enabled	tinyint(1)	
Salt	varchar(255)	
Password	varchar(255)	The user password
last_login	datetime	The date of last login
confirmation_token	varchar(180)	
password_requested_at	datetime	
Roles	longtext	The role of user
Name	varchar(100)	The user name
firstName	varchar(100)	The user first name
Phone	int(11)	The number phone of user
Address	varchar(255)	The user address
View	tinyint(1)	
date_registration	datetime	

Table 19: Fos user table

Line_order

Column	Type	Description
Id	int(11)	The identifier of the line order
Qte	int(11)	Le quantity of the dishes
comment	longtext	The comment corresponding to the order
order_id	int(11)	The identifier of the order
dish_id	int(11)	The identifier of the dish
Price	Double	The price of dish of the order

Table 20: line order table

Loyal_customer

Column	Type	Description
Id	int(11)	The identifier of the loyal customer
credit_loyalty	int(11)	The total of loyalty points
user_id	int(11)	The identifier of the user

Table 21: Loyal customer table

Manager

Column	Type	Description
Id	int(11)	The identifier of the manager
user_id	int(11)	The identifier of the user

Table 22: Manager table

method_paiement

Column	Type	Description
id	int(11)	The method payment identifier
order_id	int(11)	The order identifier corresponding to this method
date_paiement	datetime	The date of payment
type	varchar(100)	The type of payment method

Table 23: Method payment table

Orders

Column	Type	Description
id	int(11)	The identifier of the order
dateOrder	Datetime	The date of order sending
total	Double	The price total of the order
view	tinyint(1)	The order is view by the manager or not
validate	int(11)	The order is validate or not
user_id	int(11)	The identifier of the user
address	varchar(255)	The user address
zip_code	int(11)	The zip code of user

Table 24: Orders table

Picture_assortment

Column	Type	Description
id	int(11)	The identifier of the picture
url	varchar(255)	Picture extension
alt	varchar(255)	Picture name

Table 25: Picture assortment table

Picture_dish

Column	Type	Description
Id	int(11)	The identifier of the picture
url	varchar(255)	Picture extension
Alt	varchar(255)	Picture name

Table 26: Picture dish table

Picture_event

Column	Type	Description
Id	int(11)	The identifier of the picture
url	varchar(255)	Picture extension
Alt	varchar(255)	Picture name
event_id	int(11)	The identifier of event type

Table 27: Picture event table

Picture_resto

Column	Type	Description
Id	int(11)	The identifier of the picture
url	varchar(255)	Picture extension
Alt	varchar(255)	Picture name

Table 28: Picture resto table

Point_loyalty

Column	Type	Description
id	int(11)	The identifier of loyalty techniques
type	varchar(255)	The type of loyalty techniques
description	Longtext	The description of loyalty techniques
state	tinyint(1)	The chosen fidelity technique
ref	varchar(10)	The loyalty technique reference

Table 29: Point loyalty table

Price_period

Column	Type	Description
Id	int(11)	The identifier of the associative classe price period
promo_period_id	int(11)	The identifier of the promotion period
Percentage	int(11)	The percentage of that period

Table 30: price period table

Price_period_dish

Column	Type	Description
price_period_id	int(11)	The identifier of price period
dish_id	int(11)	The identifier of the dish

Table 31: Price period dish table

Promo_period

Column	Type	Description
Id	int(11)	The identifier of the promotional period
dateStart	Date	The start date of promotion period
dateEnd	Date	The end date of promotion period

Table 32: Promo period table

Reservation

Column	Type	Description
Id	int(11)	The identifier of the reservation
dateRes	Date	The date of the reservation
startTime	Time	The start time of the reservation
nbPeople	int(11)	Number of people corresponding to this reservation
user_id	int(11)	The identifier of the user who pass the reservation
State	tinyint(1)	The state of the reservation validate or not
View	tinyint(1)	The reservation request view or not

Table 33: Reservation table

Review

Column	Type	Description
id	int(11)	
dish_id	int(11)	The identifier of the dish
user_id	int(11)	The identifier of user who posted a review
nb_stars	int(11)	The number of stars posted

Table 34: Review table

Type_event

Column	Type	Description
id	int(11)	The identifier of the event type
wordingType	varchar(255)	The name of the type
value	tinyint(1)	Indicate whether the type is a requested event or an organized event

Table 35: Type event table

Interpretation of physical model

The « **Point_loyalty** » table contains retention techniques that is why has not got any relationship with other tables that is why it does not exist in the class diagram.

The table « **configuration** » allows the Manager to control the fields of the Android app.

5. Main interfaces

This part is to present the main interfaces of our realization that illustrate the different use cases already seen.

5.1. Back-office part

5.1.1. Connection Interface

Figure 37 shows the connection interface, it contains a login form that treats the data entered, as well as a link to a forgotten password recovery interface.

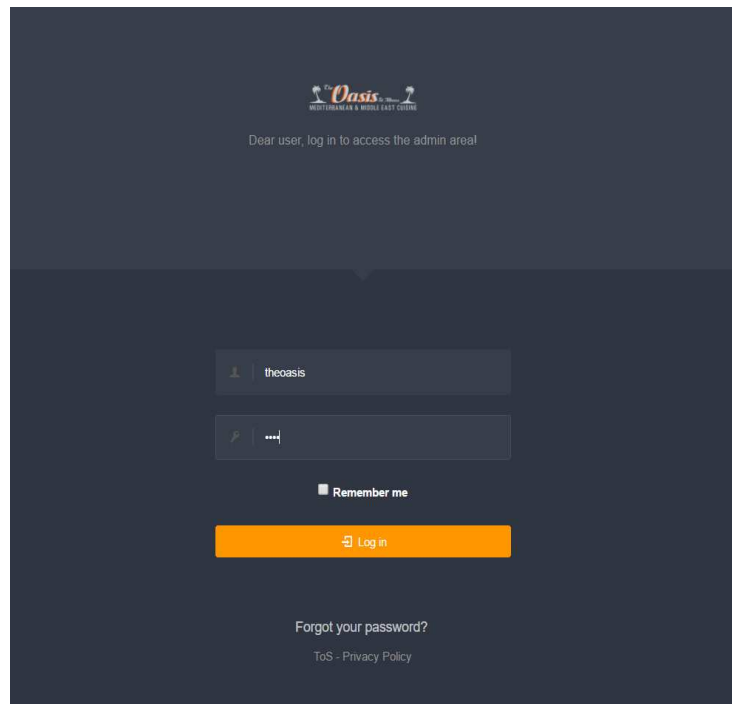


Figure 36: Connection Interface

5.1.2. « Home page » interface

This interface represents the home page contains a descriptive paragraph about the restaurant.

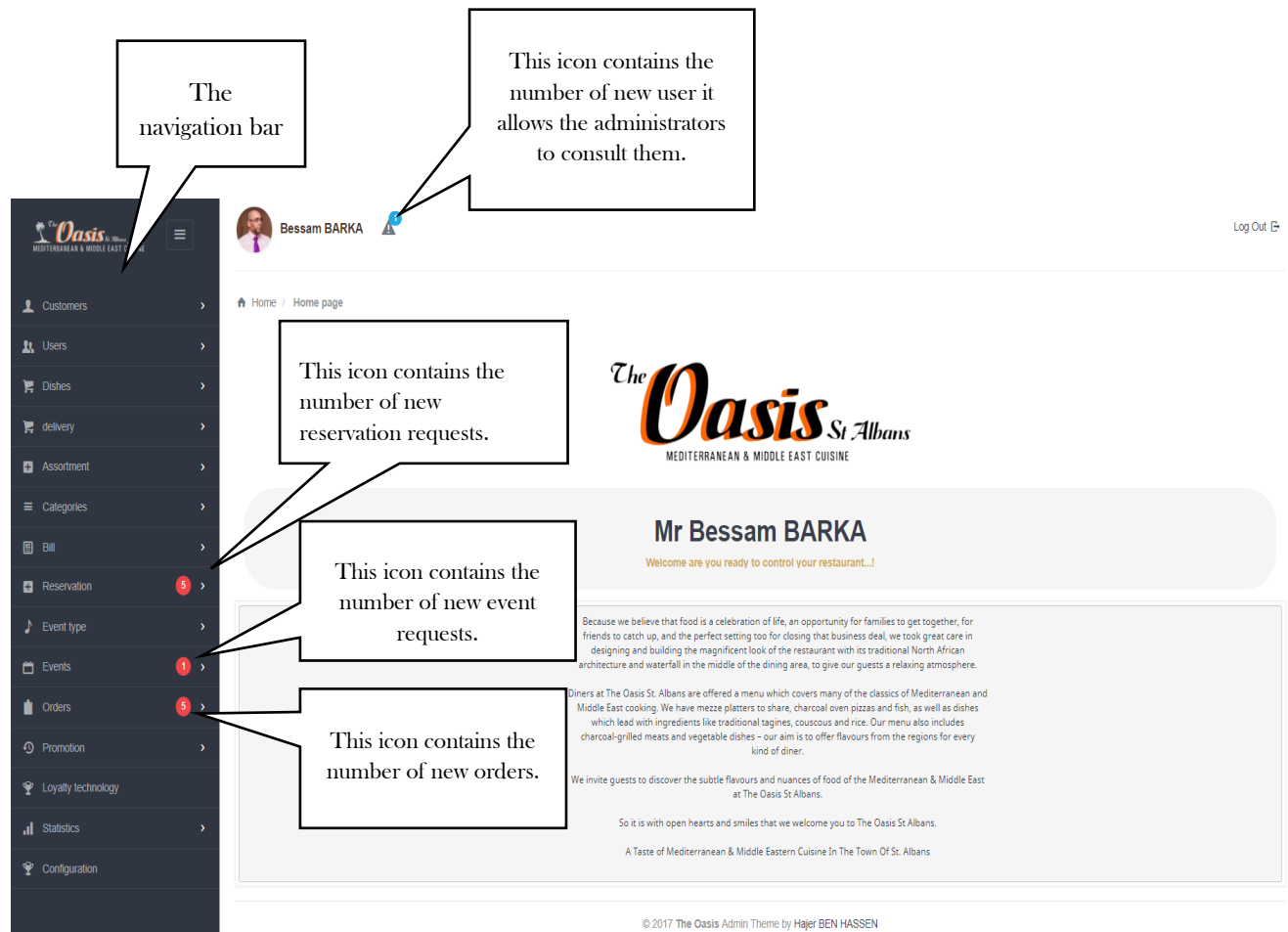


Figure 37: « Home page » interface

5.1.3. « order List » Interface

This interface contains the list of orders with the ability to review and delete them.

Home / List orders

Orders list

	Date Order	Total	Actions	State
	2017-06-10 am30 00:00	24	i Show x Delete	Reject
	2017-06-13 am30 00:00	59	i Show x Delete	Dispatcher
	2017-06-12 am30 00:00	89	i Show x Delete	Processing
	2017-05-10 am31 00:00	789	i Show x Delete	Waiting
	2017-05-16 am31 00:00	241	i Show x Delete	Waiting
	2017-07-03 am31 00:00	157	i Show x Delete	Waiting
	2017-04-11 am30 00:00	354	i Show x Delete	Waiting
	2017-04-13 am30 00:00	432	i Show x Delete	Waiting
	2017-06-29 am30 00:00	0	i Show x Delete	Waiting

Figure 38: « order List » Interface

5.1.4. « Details order » Interface

This interface contains all the details about the order selected, and through this interface the administrator can validate, reject the order.

Home / Details Order

[Details order](#)

Date order : 2017-06-10

Customer
Bessam
Barka

Full address
london , 4000
phone : 2147483647

#	Product	Comment	Quantity	Price
	Mix appetisers	nothing	5	123
	Vine Leaves	chips	2	150
	Foccacia Rossa (tomato garlic Bread)	no falafel	2	63
	Bamia B'Zeit		1	12

Grand Total: 24 £

[✓ Validate](#) [x Reject](#)

Figure 39: « Details order » Interface

Chapter 4 : Implementation

If the administrator valid the order, three new buttons should be display, to change the status of the order.

[Home](#) / [Details Order](#)

[Details order](#)

The Order is validated

Date order : 2017-06-10

Customer
Bessam
Barka

Full address
london , 4000
phone : 2147483647

#	Product	Comment	Quantity	Price
	Mix appetisers	nothing	5	123
	Vine Leaves	chips	2	150
	Foccacia Rossa (tomato garlic Bread)	no falafel	2	63
	Bamia B'Zeit		1	12

Grand Total: **24 £**

Processing

Dispatched

Complited

Figure 40: « change status order » Interface

5.1.5. « Bill » Interface

This interface allows the administrator to view all details of the order manager, and he have the possibility to printing it.

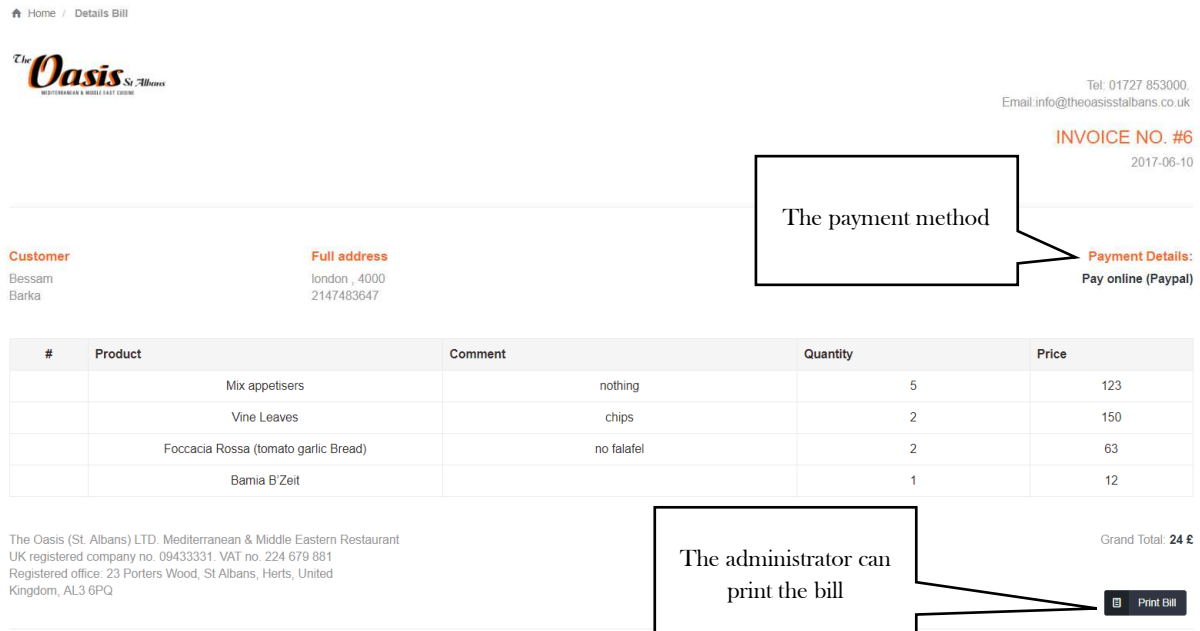


Figure 41: « Bill » Interface

5.1.6. Statistics interface « turnover by zip code »

This interface allows the consultation of annual turnover by postal code.



Figure 42: Statistics interface « turnover by zip code»

5.1.7. Statistics interface « number of orders per month »

This interface provides the manager to consult the number of requests orders in the current month. Par exemple in the month June '6' we received two orders



Figure 43: Statistics interface « number of orders per month »

5.1.8. Statistics interface « number of booking per day »

This interface allows the manager to know the number of booking per day.

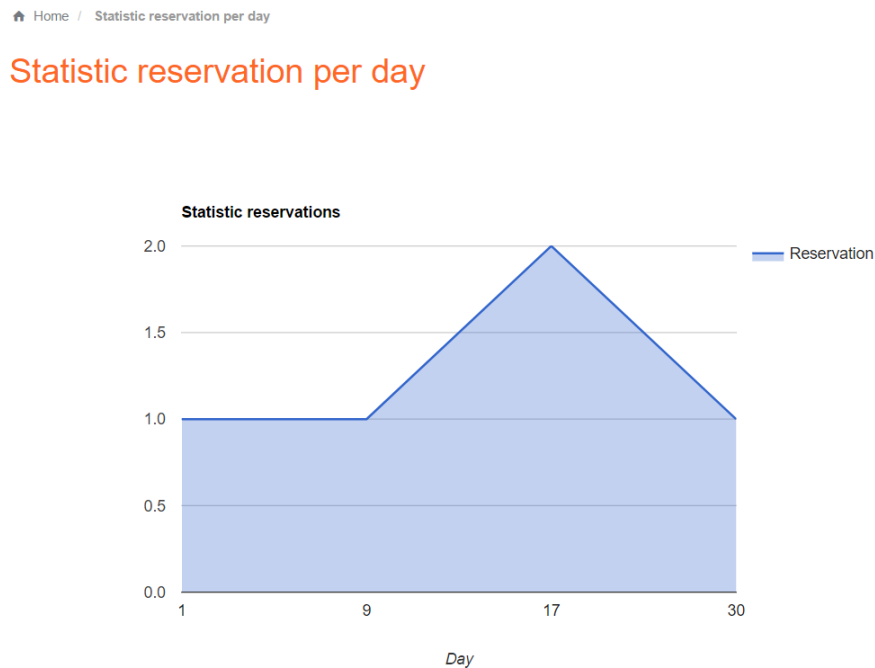


Figure 44: Statistics interface « number of booking per day »

5.1.9. Statistics interface « Dishes ordered by month »

This interface shows the percentage of the dishes ordered according to their categories.

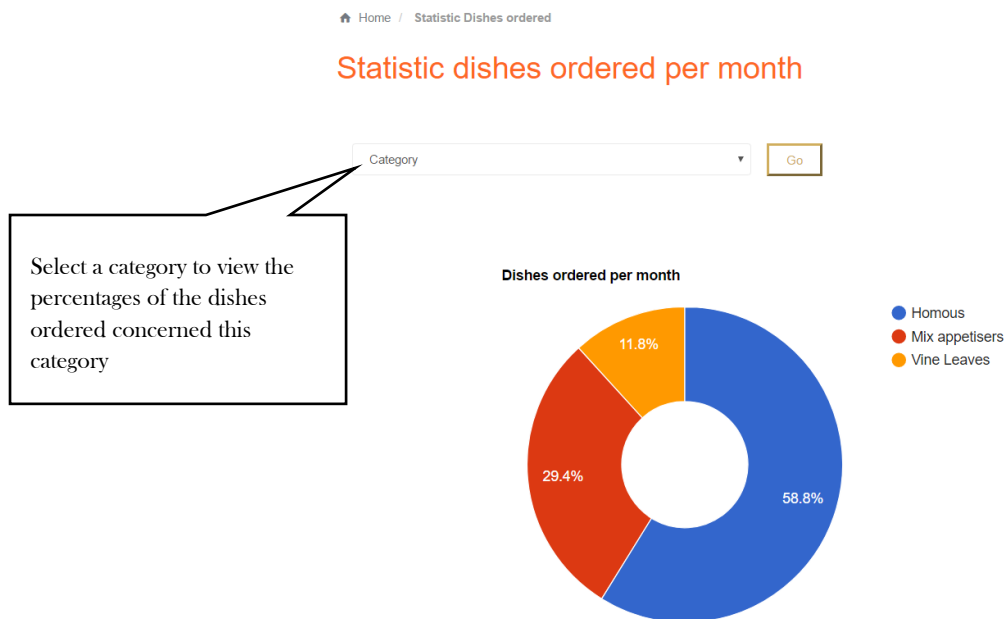


Figure 45: Statistics interface «Dishes ordered by month »

5.1.10. « dish list » Interface

This interface contains the dishes list with the possibility to consul, modify, and delete them.

Home / Dishes list

Dishes list +

Dish	Category	Price	Picture	Assortment	Actions
Homous	Cold Starters	4 £		<ul style="list-style-type: none"> Lebanese Rice Oasis Kabsa Rice 	Show Edit Delete
Labneh	Cold Starters	3.75 £		✖	Show Edit Delete
Baba Ghanouj	Cold Starters	4.5 £		✖	Show Edit Delete
Mix appetisers	Cold Starters	9.95 £		✖	Show Edit Delete
Moutabal	Cold Starters	3.95 £		✖	Show Edit Delete

« Previous 1 2 Next »

© 2017 The Oasis Admin Theme by Hajer BEN HASSEN

Figure 46: « dish list » Interface

5.1.11. « Details dish » Interface

From this interface the administrator can view the notes of each dish put by the customer, and see the details of the dish.

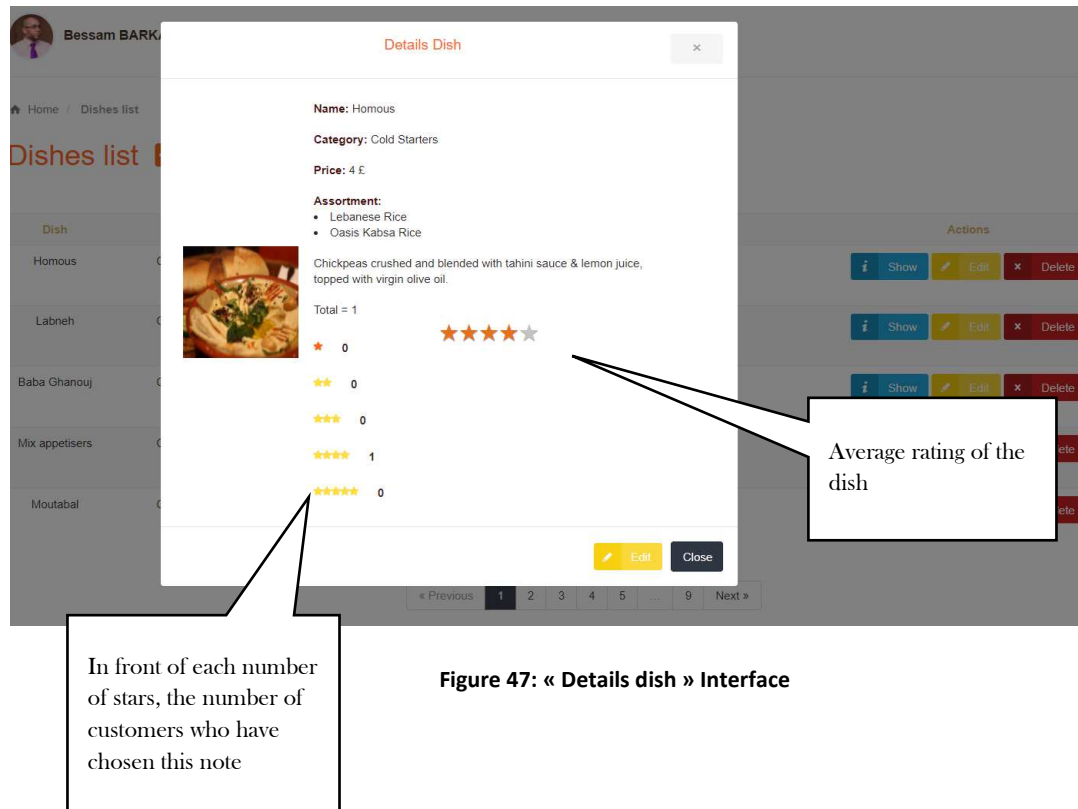


Figure 47: « Details dish » Interface

5.1.12. « Add promotion » Interface

Figure 49 presents an interface from which the manager can add a promotional period with different percentages. Each percentage linked with a list of dishes.

Home / Add promotion

New Promotion

Period Promotion

Percentage

30

Percentage of the first list of dishes in the same promotion

Dishes

Homous
Labneh
Baba Ghanouj
Mix appetisers
Moutabal

delete

Percentage

20

Percentage of the second list of dishes in the same promotion

Dishes

Baba Ghanouj
Mix appetisers
Moutabal
Vine Leaves
Bamia B'Zeit

delete

+ Add new

Add new percentage and dishes

Date start

30/06/2017

Date end

31/07/2017

Save

Figure 48: « Add promotion » Interface

5.1.13. « loyalty system » Interface

From this interface the administrator managed the loyalty system, he has the possibility to choose a technique.

Home / List of cat

Loyalty technology

Choose a loyalty system

Choose

Wordings Category	Active	Actions
Per unit of product points		<div>Edit</div> <div>Delete</div>
By purchase level		<div>Edit</div> <div>Delete</div>
By purchase threshold		<div>Edit</div> <div>Delete</div>
No type	✓	<div>Edit</div> <div>Delete</div>

© 2017 The Oasis Admin Theme by Hajer BEN HASSEN

Figure 49: « loyalty system » Interface

5.2. Front-office part

5.2.1. Splach interface



Figure 50: Splach interface

5.2.2. Home interface

This interface presents the main fonctionnalities.

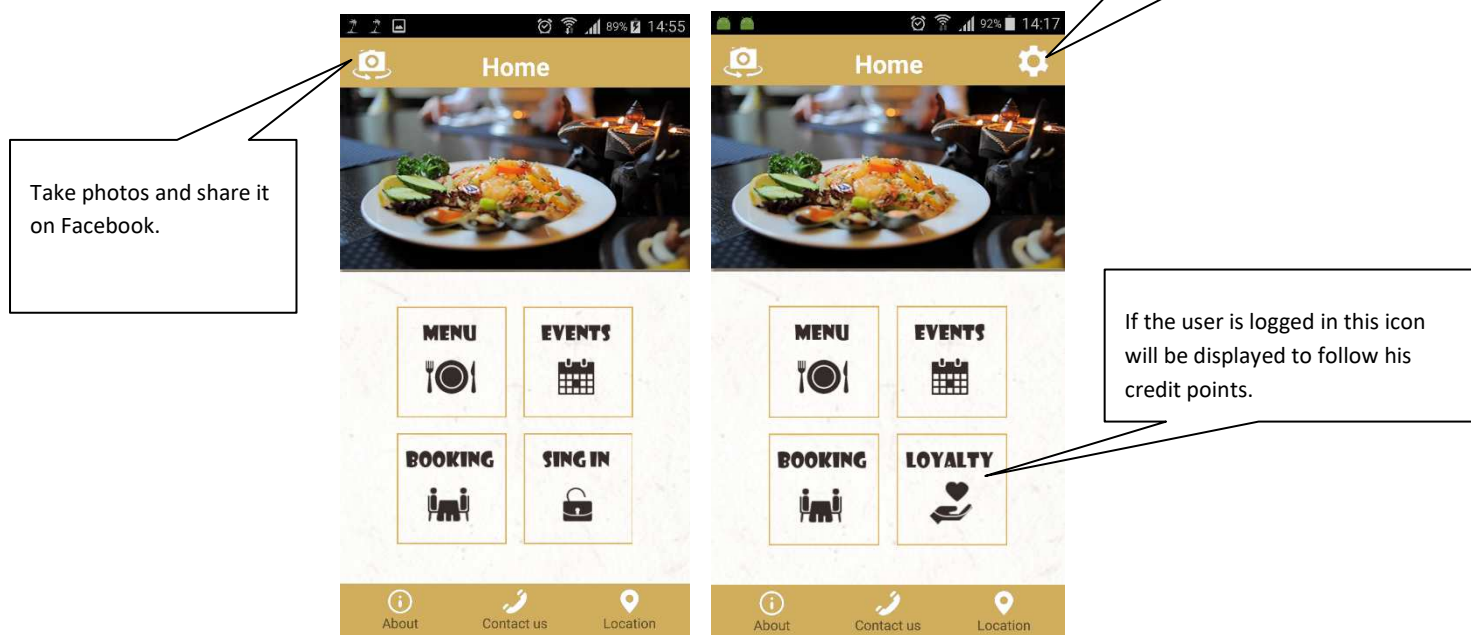


Figure 51: Home interface

5.2.3. Menu interface

This interface presents all the categories.

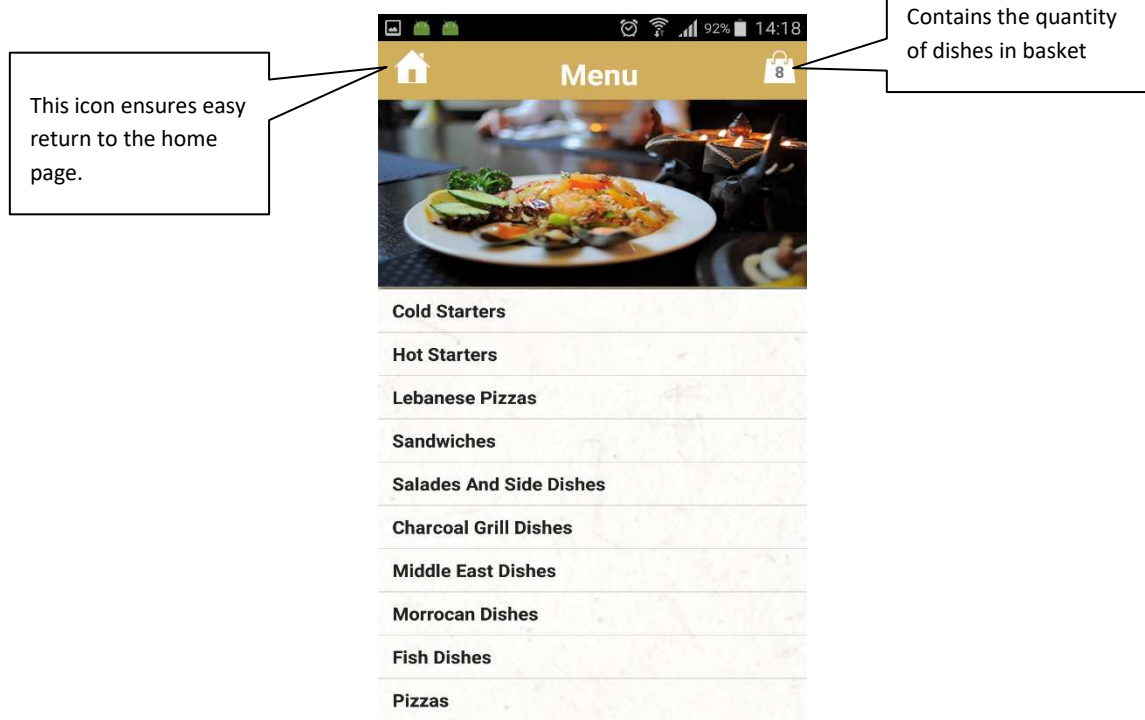


Figure 52: Menu interface

5.2.4. Details dish interface

This interface includes the dish details and from this interface the user can add to favorites, vote, choose quantity and he has the possibility to send a special description.

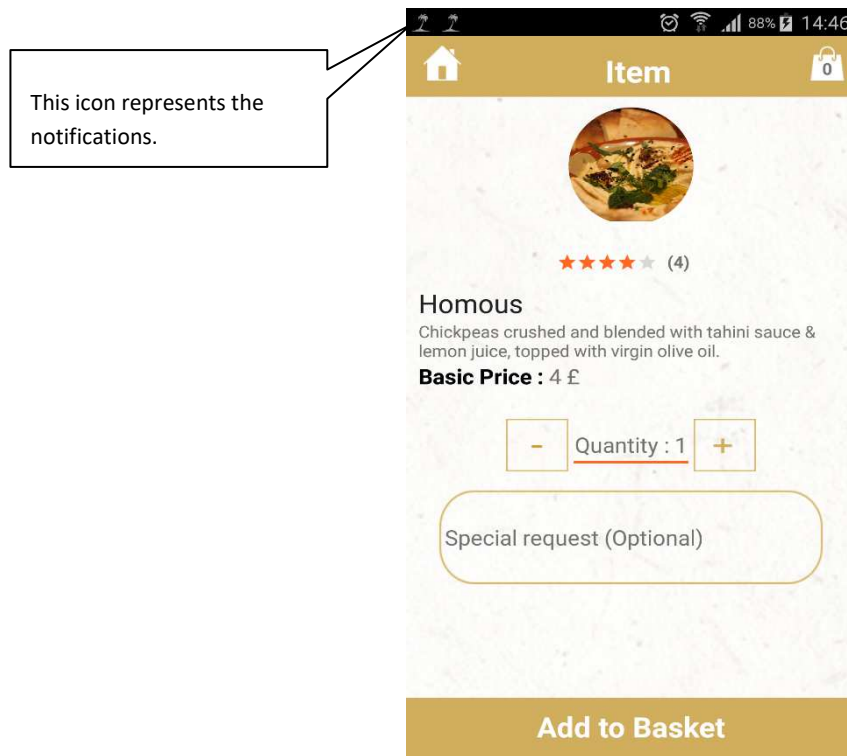


Figure 53: Details dish interface

5.2.5. Delivery interface

From this interface the user can select the type of delivery, choose the date and time of order and enter his address.

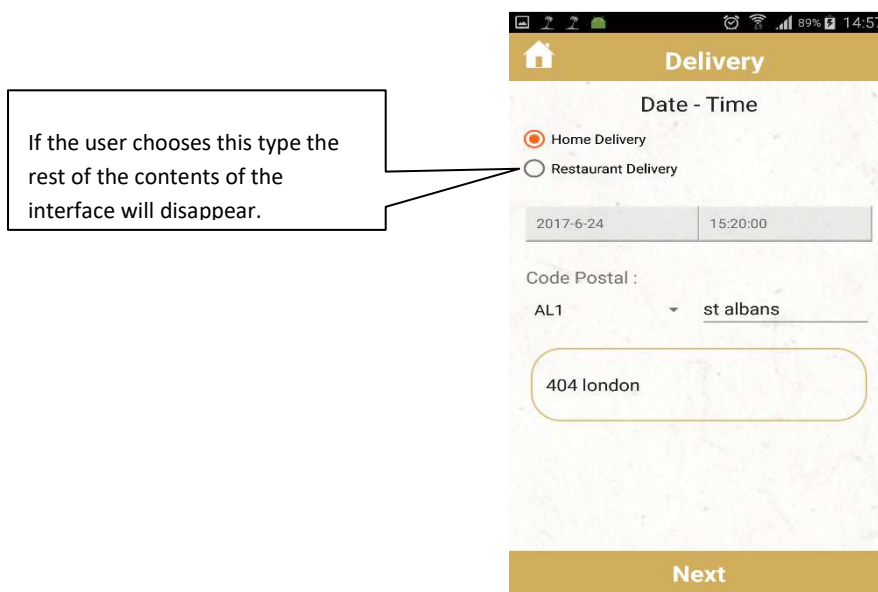


Figure 54: Delivery interface

5.2.6. Sign in interface

Figure 52 shows the connection interface, it contains a login form that treats the data entered, as well as a link to a forgotten password recovery interface and a link to the sign up interface.

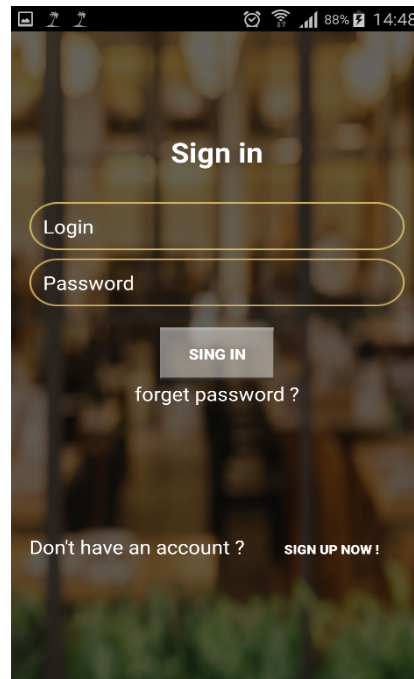


Figure 55: Sign in interface

5.2.7. Account interface

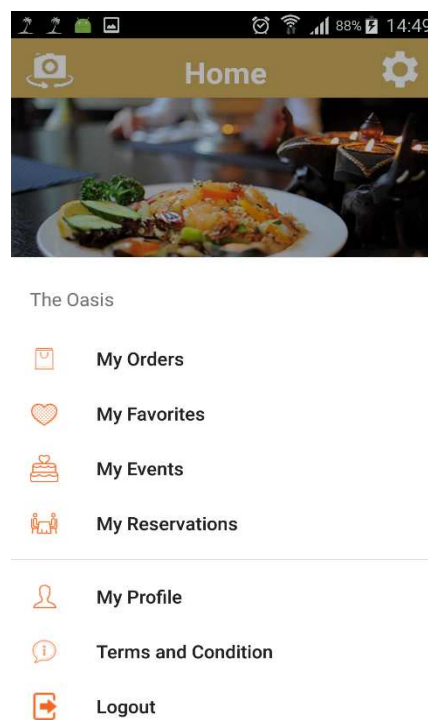


Figure 56: Account interface

5.2.8. Order interface

This interface contains the order details.

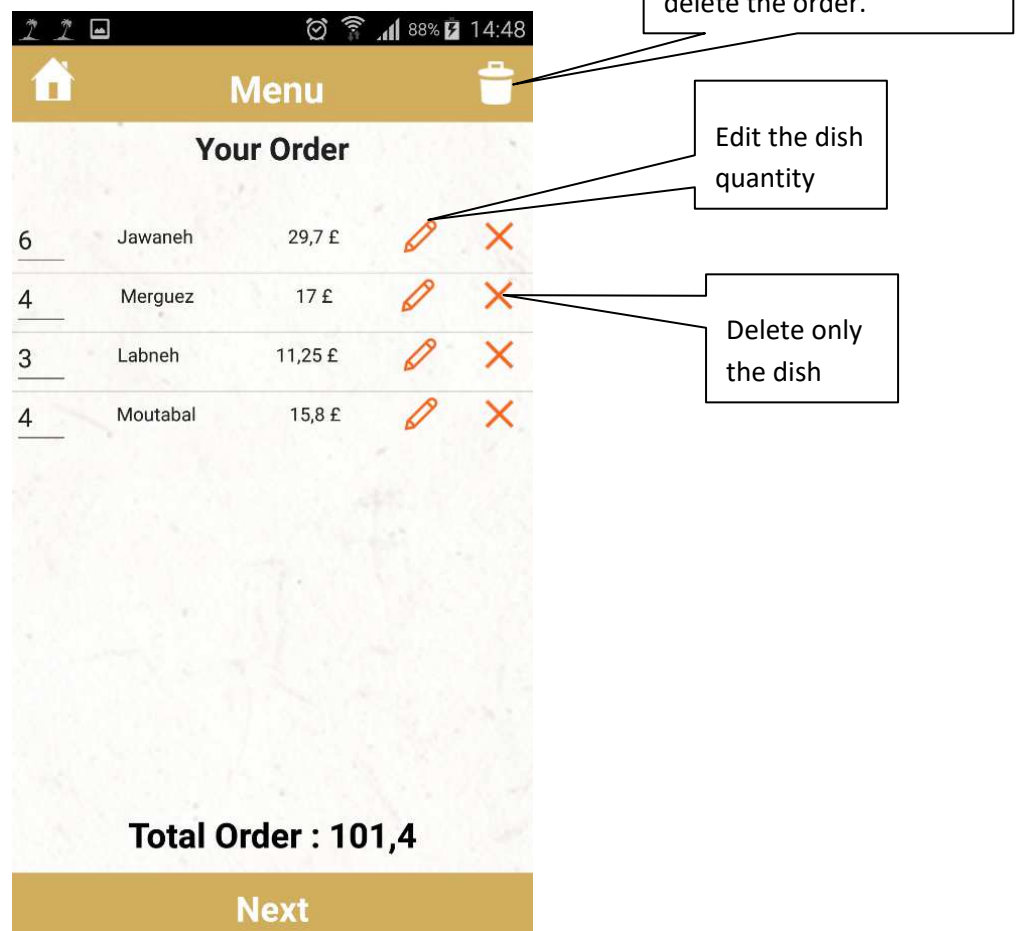


Figure 57: Order interface

5.2.9. “Manage loyalty points” interface

From this interface the loyal customer can consult and convert his points.

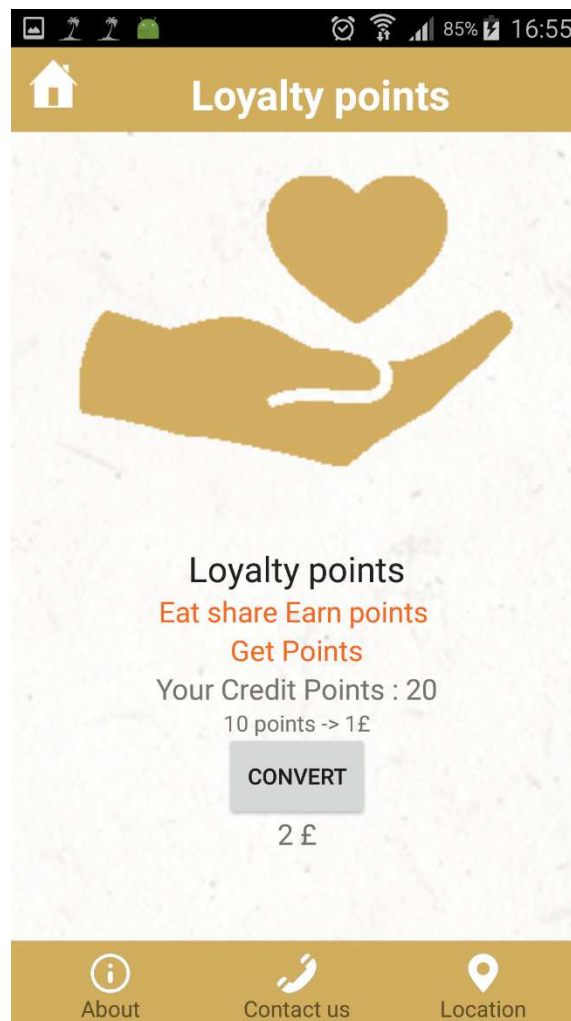


Figure 58: "Manage loyalty points" interface

5.2.10. “Pay with PayPal” interfaces

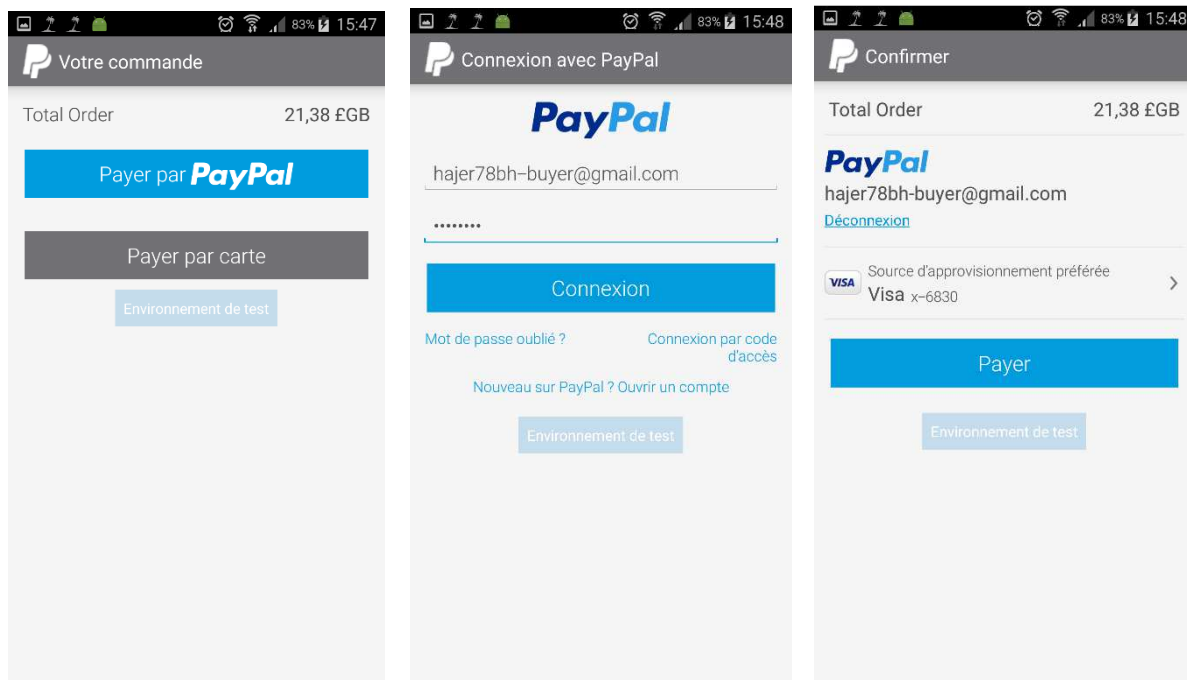


Figure 59: "Pay with PayPal" interfaces

6. Conclusion

Throughout this chapter, we introduced our working environment that consists of both hardware and software tools. In addition, we presented the technologies used, the principles of ergonomics, finally, we concluded by the most significant graphical interfaces.

General conclusion

General conclusion

This is the required end of studies project carried on the Higher Institute of Computer Science and Communications situated in Hammam Sousse. The objective of this project was to design and develop an Android and web (part Back-Office) application for the restaurant named "The Oasis.

The majority of existing applications related to this project that are available on the market offer limited functionalities. We have created an application that allows the user to view offers and promotions offered by the restorer, place an order online, book a table, organize events, as well as we allowing customers to earn loyalty points and give a rating on each dish.

On the other hand, we have developed a web application to the Back-Office allowing the administrator to manage mobile application and see different statistics.

Concerning the adopted approach, in first place we collect necessary information to draw up a statement of the existing situation and present an overview of the issue then finally we conclude by the proposed solution. Subsequently, we started a careful analysis of the main features of the application to be developed.

For the conceptual design, we have employed the UML unified modeling language. In this approach we presented a static and dynamic study in order to determine the overall structure of the application.

In the last part, we presented the used tools for the application's hardware and software, programming languages used and the most significant interfaces of our application.

All the features described at the level of functional specifications were developed and validated. However, there is not a perfectly completed project. Therefore, my project can be improved by the addition of other features like:

- ✓ A GPS chronometer provided for the customer to follow the route taken and the time required for the delivery of the order.
- ✓ Payment with the credit card.

For farther future work, we propose to develop a web version and a mobile version that is compatible with other mobile platforms.

The contribution of this work has been of considerable importance, it allows us to discover and well master development under the Framework of Symfony and the Android development environment and deepen my knowledge of programming in the JAVA language.

Bibliography

[1]: MVC Model

=> http://repository.nbyl.me/publications/thesis/these_licence_v1.pdf/ (accessed 17/04/2017).

[2]: State transition diagram

=> Benoit CHRROUX, Aomar OSMANI, Yann THIERRY-MIEG 12 January 2008 “UML 2 2nd edition”.

Webography

[W1]: e-commerce context

⇒ <https://www.statista.com/markets/413/topic/457/b2c-e-commerce/> (accessed 17/04/2017).

[W2]: e-fidelity context

⇒ <https://www.e-monsite.com/pages/tutoriels/e-commerce/comment-utiliser-le-programme-de-fidelite-de-la-boutique-en-ligne.html/> (accessed 31/05/2017).

[W3]: GanttProject

⇒ <http://vviale.developpez.com/tutoriels/ganttproject/> (accessed 27/04/2017).

[W4]: Class diagram

⇒ <http://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/diagramme-de-classe/> (accessed 20/04/2017).

[W5]: Package Class diagram

⇒ <https://www.modeliosoft.com/fr/ressources/exemples-de-diagrammes/diagrammes-de-classes-et-de-packages-uml.html/> (accessed 1/05/2017).

[W6]: Deployment diagram

⇒ https://www.ibm.com/support/knowledgecenter/fr/SS8PJ7_9.6.0/com.ibm.xtools.modeler.doc/topics/cdepd.html/ (accessed 20/05/2017).

[W7]: Symfony

⇒ <http://www.supinfo.com/articles/single/3195-symfony-c-est-quoi/> (accessed 05/05/2017).

[W8]: PHPStorm

⇒ <https://www.video2brain.com/phpstorm/> (accessed 05/05/2017).

[W9]: Android Studio

⇒ <https://developer.android.com/studio/index.html/> (accessed 05/05/2017).

[W10]: Genymotion

⇒ <https://www.tech-connect.info/tutoriels/genymotion-est-un-emulateur-android-pour-windows-mac-et-linux/3149/> (accessed 05/05/2017).

[W11]: WampServer

⇒ <http://www.wampserver.com/> (accessed 05/05/2017).

[W12]: Adobe Photoshop

⇒ <http://www.mosaique-info.fr/glossaire-web-referencement-infographie-multimedia-informatique/p-glossaire-informatique-et-multimedia/228-photoshop-definition.html/> (accessed 05/05/2017).

[W13]: HTML5

⇒ <https://openclassrooms.com/courses/dynamisez-vos-sites-web-avec-javascript/qu-est-ce-que-le-html5/> (accessed 05/05/2017).

[W14]: CSS3

⇒ <http://glossaire.infowebmaster.fr/css/> (accessed 05/05/2017).

[W15]: JavaScript

⇒ <https://openclassrooms.com/courses/dynamisez-vos-sites-web-avec-javascript/introduction-au-javascript/> (accessed 05/05/2017).

[W16]: Choice of colors

⇒ <http://evolutiongraphique.com/la-signification-cachee-des-couleurs-en-communication-visuelle/> (accessed 05/05/2017).

Abstract

This Project was conducted as part of my graduation project within « Higher Institute of Computer and Communication Techniques Hammam Sousse ». It aims at developing a mobile and web application for the restaurant “The Oasis”, to improve the efficiency of the external and internal management of the restaurant, to achieve a better organization of the work and get a competitive advantage.

The back-office part was developed by the Framework Symfony, and the front-office part was developed using Android studio.

Résumé

Ce projet a été mené dans le cadre de mon projet de graduation au sein du « Institut Supérieur des Techniques d'informatique et de communication Hammam Sousse ». Il vise à développer une application mobile et web pour le restaurant "The Oasis", afin d'améliorer l'efficacité de la gestion externe et interne du restaurant et d'atteindre une meilleure organisation du travail et de décrocher un avantage concurrentiel.

La partie Back-office a été développée par Framework Symfony, et la partie front-office a été développée à l'aide de Android studio.