

# Master Team

# Project WS 2022:

# Risto

Team 3 of Global Distributed Software Development (Hochschule Fulda):

- Luis Miguel García Marín (Team Lead)  
[luis-miguel.garcia-marin@informatik.hs-fulda.de](mailto:luis-miguel.garcia-marin@informatik.hs-fulda.de)
- Jesús Moreno Durán (Back End Lead)
- Noman Ali (Front End Lead)
- Paras (GitHub Master)
- Vichitar Dagar (Back End Team Member)
- Alhassane Dondo Toure (Front End Team Member)

Milestone 2

Date: 11.12.2022

History Table	
Date submitted	Date revised
22.12.2022	22.12.2022

## **1. Functional Requirements - prioritized**

1. Restaurants can register in the system. (Priority: 1)
2. Restaurants can upload information about themselves to subscribe to the service. (Priority: 1)
3. Site administrators can approve or decline restaurant info for posting. (Priority 1)
4. Site administrators can deal with typical admin duties. (Priority 1)
  - 4.1. Site administrators can manage user registrations.
  - 4.2. Site administrators can accept or decline review posts.
  - 4.3. Site administrators can delete posts.
  - 4.4. Site administrators can ban users with bad behavior.
5. Customers can search for restaurants. (Priority 1)
6. Customers can check the table status of a restaurant (live view). (Priority 3)
7. Customers can manage their reservations. (Priority 1)
  - 7.1. Customers can make a reservation.
  - 7.2. Customers can change their reservation.
  - 7.3. Customers can cancel their reservation.
8. Customers can make orders of food before their arrival to a restaurant in which they have made a reservation. (Priority 1)
9. Customers can make orders of food after their arrival to a restaurant in which they have made a reservation. (Priority 2)
10. Customers can manage posts of reviews.
  - 10.1. Customers can post reviews. (Priority 1)
  - 10.2. Customers can edit their reviews. (Priority 2)
  - 10.3. Customers can delete their reviews. (Priority 1)
11. Managers (Host/hostess) can review daily calendar. (Priority 1)
12. Managers (Host/hostess) can check incoming guests to greet them. (Priority 1)
13. Managers (Host/hostess) can check the table status of their restaurant. (Priority 2)
14. Managers (Host/hostess) can set tables for free when they are available. (Priority 1)
15. Managers (Host/hostess) can set parking slots for free when they are available. (Priority 1)
16. Waiters can check the food order list. (Priority 1)
17. Waiters can fetch orders and their corresponding tables. (Priority 1)
18. Waiters can set food orders which they have given to a table as delivered. (Priority 1)
19. Customers can check the food processing time. (Priority 2)
20. Customers can check the list of food with their price. (Priority 1)
21. Customers can reserve a parking spot near the restaurant parking point. (Priority 1)
22. Customers can chat with restaurant managers. (Priority 1)

## **2. List of main data items and entities (expand as necessary)**

1. **Restaurant:** This term refers to any listed restaurant in the software offering reservation services. Restaurants serve as the primary source of information for the application.
2. **Resource:** A term used to refer to any storage entity of the software. For instance, a food item, an image, or any data element are all resources.
3. **User:** Any client of the software interacting with the interface to use the services offered. A user is further categorized into Customer, Guest, Administrator, Waiter and Manager.
4. **Customer:** A known client of the software making use of the features offered. A client may manage their own data, posts or reservations. A customer can further read only the data public on the website.
5. **Guest:** An anonymous client of the software making use of the features offered without registering for the use of service. Guests can only browse website and cannot use the reservation services unless they are registered.
6. **Staff:** An entity that represents a set of users that represent a particular restaurant. It is a generalization for Waiter, and Manager.
7. **Waiter:** A user entity used to represent a staff member particularly responsible to serve customers. A waiter takes care of taking customer orders manually and serve their orders.
8. **Administrator:** A term used to represent a user responsible for approving customer posts or requests. Administrators can also manage user registrations and other maintenance related information on the software.
9. **Manager:** A user responsible for regulating the information related to the restaurant they belong to. They can manage vacancy of tables, and the daily calendar.
10. **Reservation:** An entity used to represent the process involving a customer reserving a seat(s) at a Restaurant.
11. **Restaurant Lookup:** An entity used to represent the process of searching for specific restaurants by the users.
12. **Food:** An entity to represent one specific food in regard to a particular food category and a particular restaurant.
13. **Table:** An entity used to represent a seat belonging to a particular restaurant. One table can only be reserved by one customer, however, a customer may reserve more than one tables.
14. **Table Map:** An entity used to represent a layout map of tables in a particular restaurant. It will represent information about the arrangement of tables in a restaurant.

**15. Posts:** An entity that represents a request for reservation or a query, or a feedback uploaded by a particular user.

**16. Food Category:** A logical entity used to characterize a particular set of similar food entities.

**17. Order:** An entity used to represent a customer's food request to a restaurant. It can refer to multiple food items.

**18. Notification:** A logical entity used to represent information delivered to the customer from a restaurant or the software end.

**19. Discount:** An entity used to represent an amount that is offered as a discount to a particular customer for a particular order or food item. Discounts can be linked to a particular food category or a particular restaurant and are volatile in nature.

**20. Promotion:** used to refer to the logic of a restaurant being highlighted on the landing page of the software for a particular amount of time. Promotions usually include occasional discounts.

**21. Parking:** An entity used to represent the logical concept of parking of a customer vehicle in the restaurant parking lot.

**22. Chat Logs:** A chat log represent message communication between Customer and Staff. Any message will be stored in the chat logs entity.

**23. Review:** A review represent user's feedback in regards to a restaurant or a food item belonging to a particular restaurant.

### **3. UI Mockups and Storyboards (high level only)**

#### **1. Restaurant registration and administrator approbation and typical duties:**

LOGO

Log inRegister

## LOG IN

Email

Password

Log in

LOGO

Log inRegister

## REGISTER

Register as a customer

Register as a restaurant

LOGO

Log inRegister

## REGISTER AS A CUSTOMER

Name

Email

Password

Confirm password

Register

LOGO

Log inRegister

## REGISTER AS A RESTAURANT

Restaurant name	Manager name
<input type="text"/>	<input type="text"/>
Restaurant nickname	Manager email
<input type="text"/>	<input type="text"/>
Restaurant address	Password
<input type="text"/>	<input type="password"/>
Restaurant telephone	Confirm password
<input type="text"/>	<input type="password"/>
Restaurant type	
Chinese restaurant ▾	

Register

LOGO

Log inRegister

## REGISTER AS A RESTAURANT


Success!

Once your restaurant is revised by our administrators you will be able to log in with the submitted information.

Go back to home page

LOGO

Admin Panel

 Administrator

## WELCOME ADMINISTRATOR!

View pending restaurant registrations


View pending review posts

View list of users


View list of restaurants


LOGO


Admin Panel

 Administrator

**PENDING RESTAURANT REGISTRATIONS**


Name: ExampleEmail: example@mail.com

Name: Example2Email: example2@mail.com

Name: Example3Email: example3@mail.com

LOGO

Admin Panel

 Administrator

**PENDING RESTAURANT REGISTRATIONS**

Restaurant name

Example

Restaurant nickname

example

Restaurant address

example 1

Restaurant telephone

123456789

Manager name

Name

Manager email

example@mail.com

Restaurant type

Chinese restaurant

Decline

Accept



LOGOAdmin PanelAdministrator

**PENDING REVIEW POSTS**

Post by example\_customer to example\_restaurant  
16.12.2022 11:04

➔

Post by example\_customer to example\_restaurant  
16.12.2022 10:04

➔

Post by example\_customer to example\_restaurant  
16.12.2022 9:04

➔

LOGOAdmin PanelAdministrator

**PENDING REVIEW POSTS**

Post by example\_customer to example\_restaurant:  
16.12.2022 11:04

Customer email  
example@mail.com

☒ Quick Service

☐ Polite Behaviour

☒ Delicious Food

☐ Value For Money

☒ Other: Example review.


★★★★★

Decline


Accept


LOGO

Admin Panel


 Administrator

## LIST OF USERS




 Name: Example Email: example@mail.com 

Ban user

 Name: Example Email: example@mail.com 


Ban user

 Name: Example Email: example@mail.com 


Ban user

LOGO

Admin Panel

 Administrator

## BAN USER

 Name: Example Email: example@mail.com


Reason

Cancel

Ban

LOGO

Manager Panel

 Manager

Guest

Reservation list

Table reservations

Messages

My Restaurant

Restaurant photo

Restaurant name

Edit information

InfoMenuTable viewParkingReviews

Restaurant name

phone


hours and days

Address

Description

LOGO

Manager Panel

 Manager

### EDIT RESTAURANT INFORMATION

Restaurant name

Restaurant\_name

Restaurant nickname

Restaurant\_nickname

Restaurant address

Restaurant\_address

Restaurant telephone

Restaurant\_telephone

Restaurant photo

↑

Restaurant table map

↑

Restaurant parking slots

30

Restaurant type

Chinese restaurant


▼

Cancel

Save

LOGO

Manager Panel

 Manager

[Guest](#)

[Reservation list](#)

[Table reservations](#)

[Messages](#)

[My Restaurant](#)

Restaurant photo

Restaurant name

Edit information

Info

Menu

Table view

Parking

Reviews

Photo

Food1

Edit

Ingredients

Delete

Photo

Food2

Edit


Ingredients

Delete

Add food

LOGO

Manager Panel

 Manager

### FOOD INFORMATION

Food name

Food photo

Food category

Chinese food


Ingredients

Cancel

Save

LOGO

Admin Panel

 Administrator

Restaurant photo

Restaurant name

Reject information

Ban

Info

Menu

Table view

Parking

Reviews

Restaurant name

phone


hours and days

Address

Description

LOGO

Admin Panel

 Administrator

REJECT RESTAURANT INFORMATION

Photo

Name: Example

Email: example@mail.com

Warn about


Example warn reason.

Cancel

Warn

LOGO

Admin Panel

 Administrator

## BAN RESTAURANT

Photo

Name: Example

Email: example@mail.com

Reason

Example ban reason.

Cancel

Ban

## 2. Search for a restaurant and booking a table:

Risto

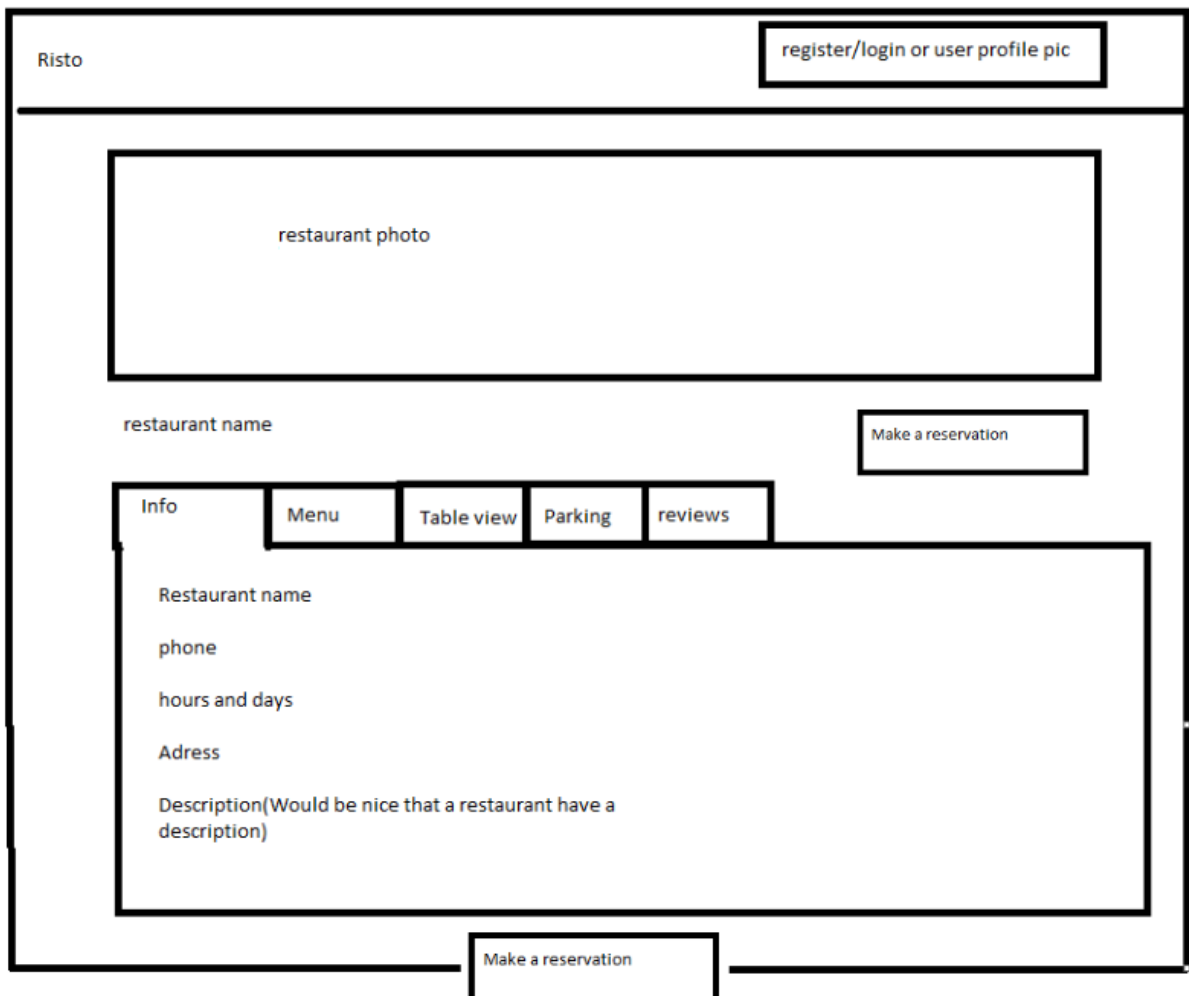
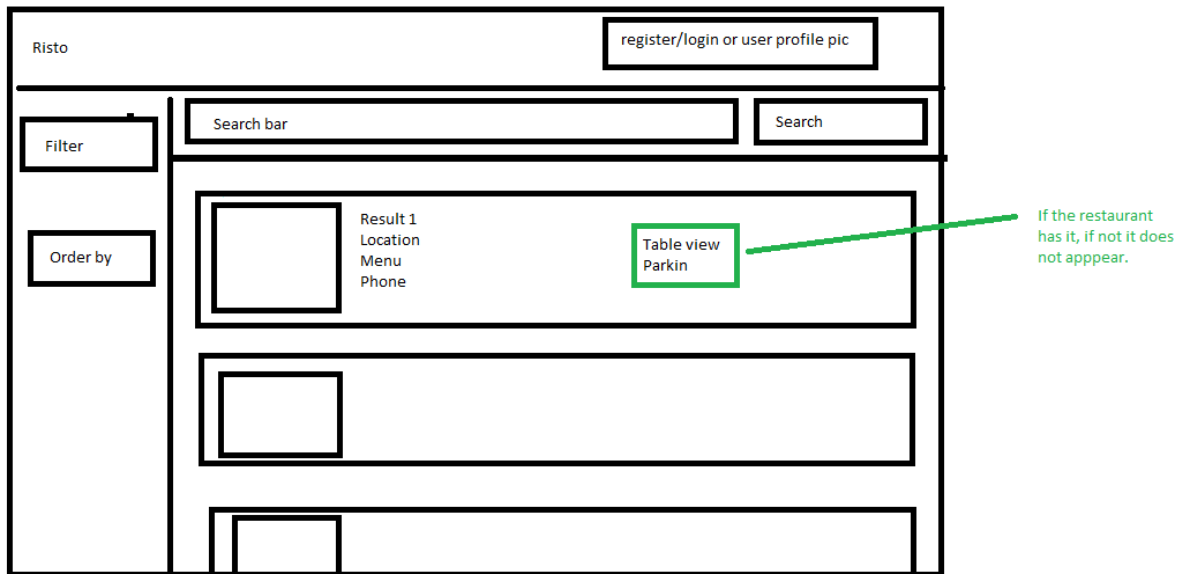
register/login or user profile pic

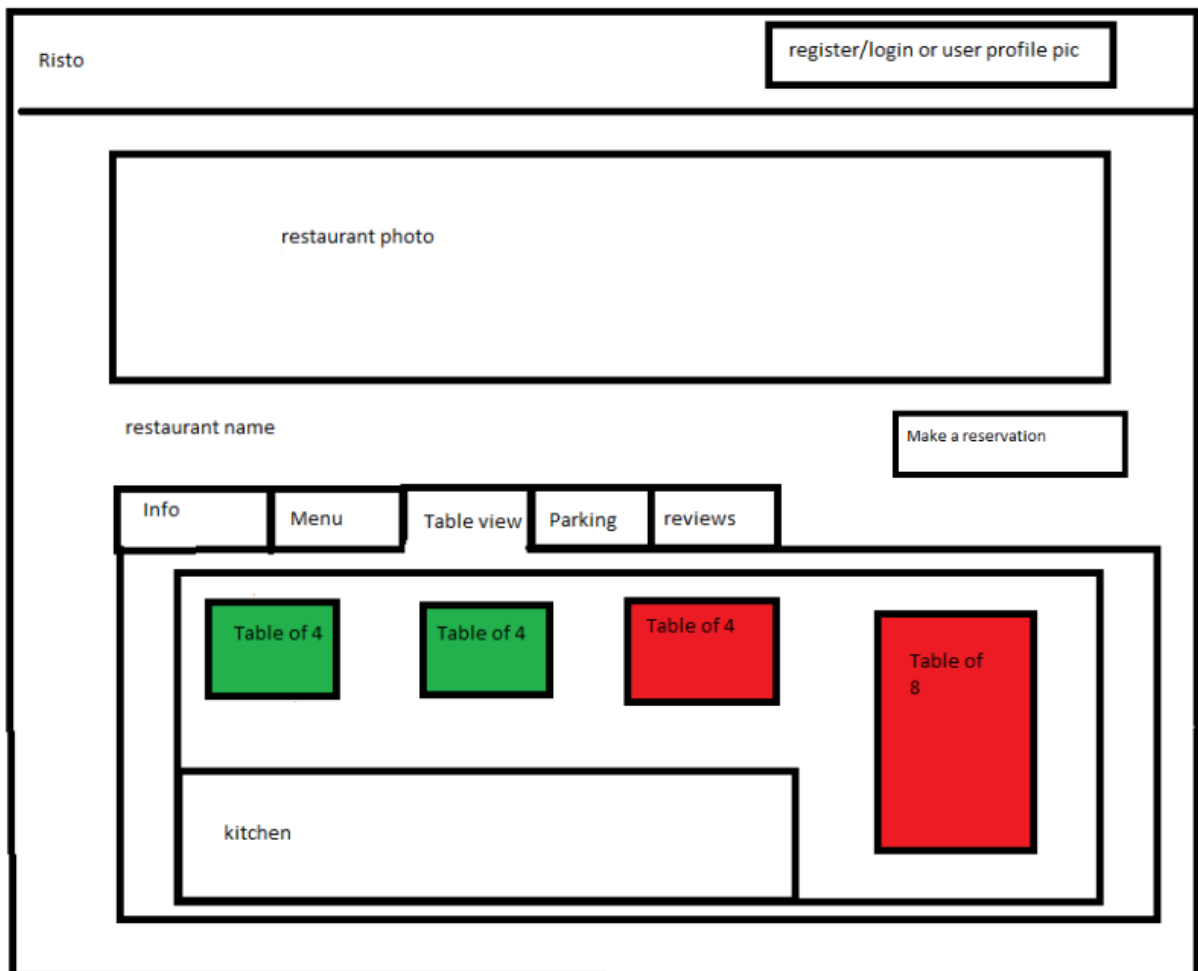
Search bar

Search button


Category

Order by







Risto

Reservation

photo

Restaurant\_name

Date and hour

Number of people

Select date


Select hour

Select a table (if possible)

Select food

Parking (if possible)

Make a reservation

Risto

Reservation

photo

Restaurant\_name

Select a free table

Table of 4


Table of 4

Table of 4

Table of 8

kitchen

Save table

Risto 

---

Reservation

photo

Restaurant\_name

Select food

Photo

Name

Ingredients

-

0

+

⋮

Commentary

Save food

Risto 

---

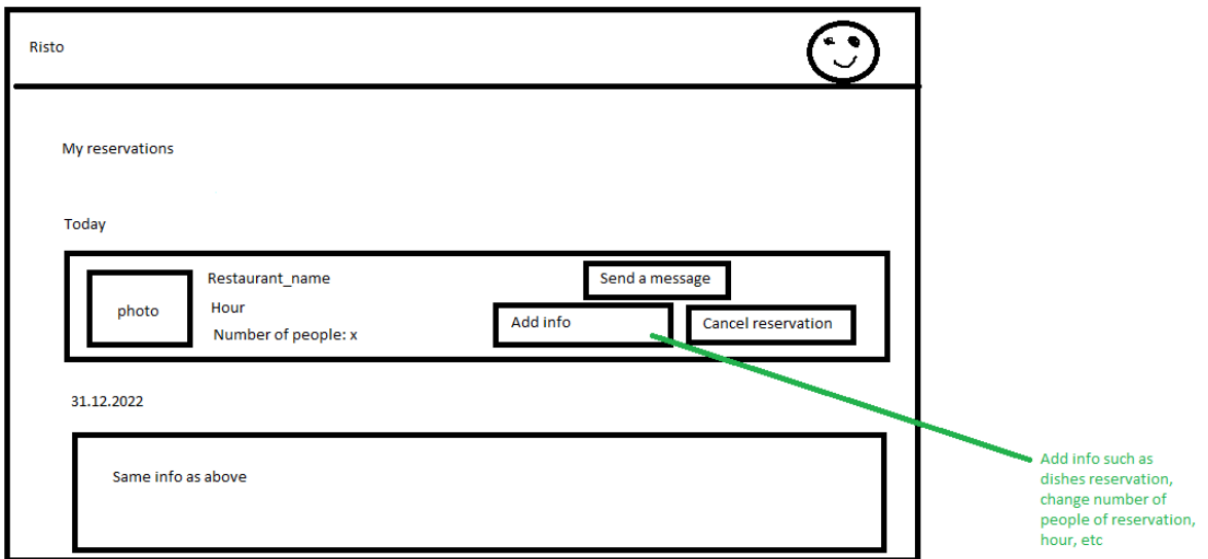
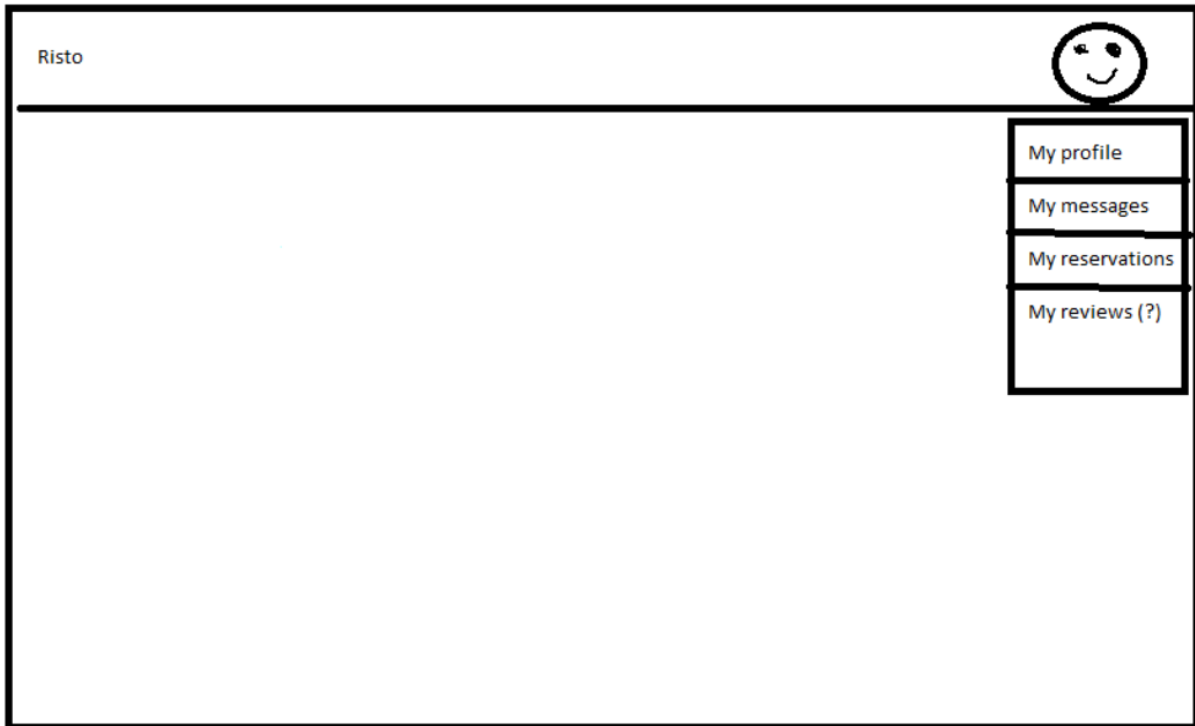
Reservation


photo

Restaurant\_name

Parking places you will need

SAve parking



Risto


Edit reservation

photo

Restaurant\_name

Date and hour

Select date

Select hour

Number of people

Select a table (if possible)


edit food

Parking (if possible)

Save changes

Cancel changes

### 3. Checking incoming guests and setting free table occupation:

Risto
Manager Panel


[Guest](#)  
[Reservation list](#)  
[Table reservations](#)  
[Messages](#)  
[My Restaurant](#)

Today's Reservations -

Table no. - XX Guest Name	inform Chef at 12:30	pre-order: xyz(if any)	Time of arrival - 00:00 parking : yes/no	➔
Table no. - XX Guest Name	NO Pre-orders	pre-order: xyz(if any)	Time of arrival - 00:00 parking : yes/no	➔
Table no. - XX Guest Name	inform Chef at 12:30	pre-order: xyz(if any)	Time of arrival - 00:00 parking : yes/no	➔



Mr. Guest

Table no. - XX

Arrival Timings :- 00:00

Pre-ordered food details

Name	quantity
1. xyz	2
2. abc	0.5
3. pqr	1

Table View



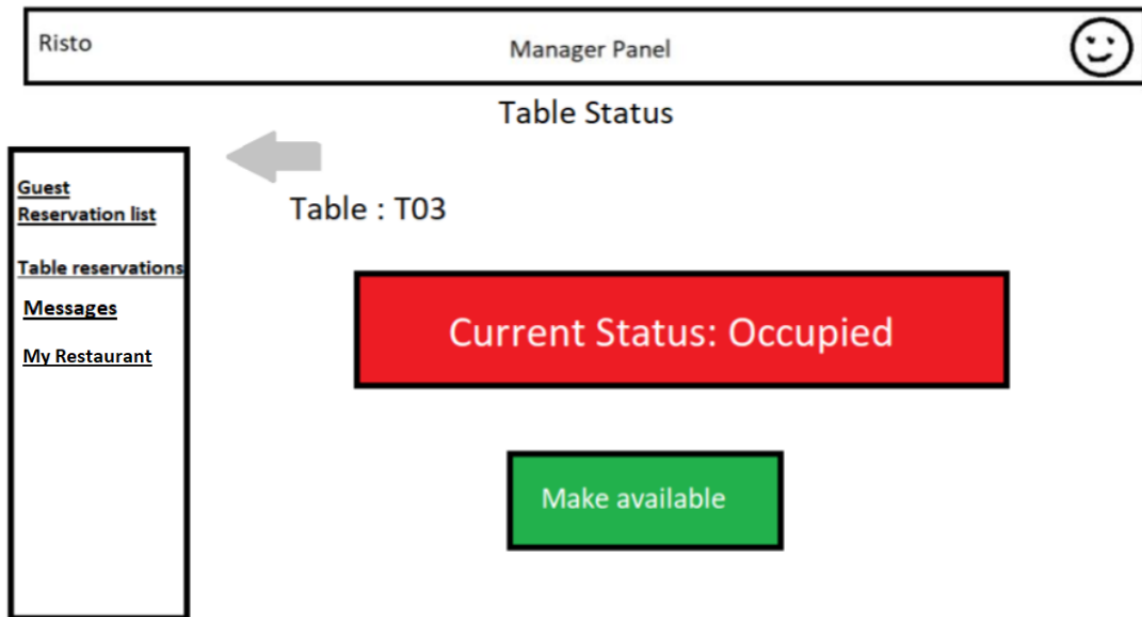
Parking: 2/Not Required

Table Status

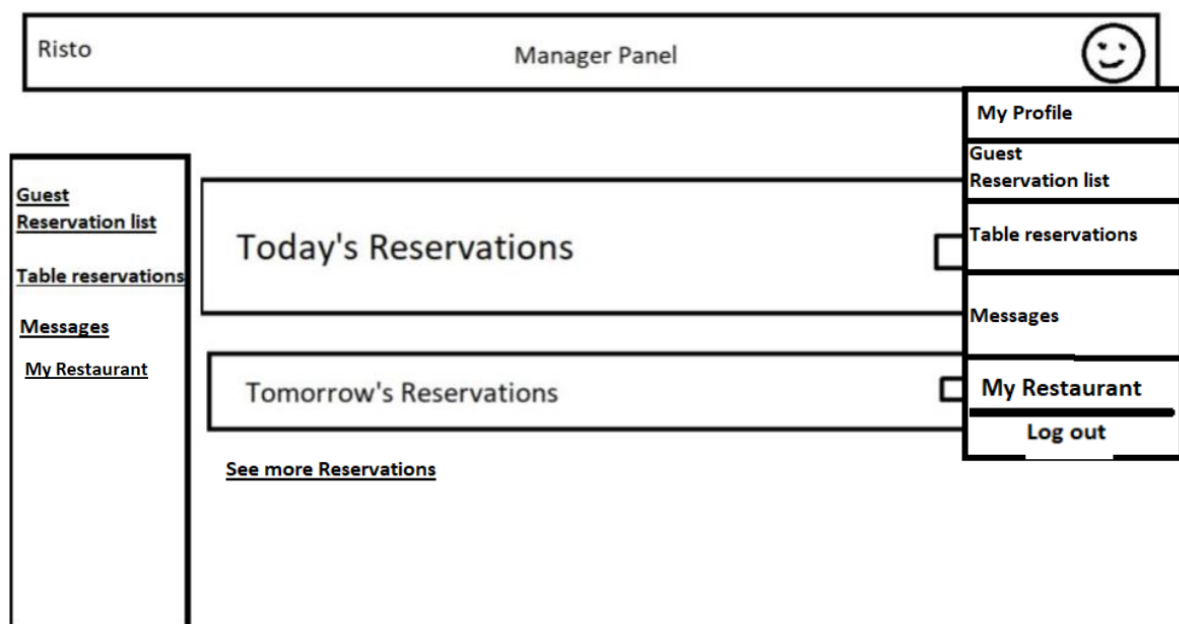
At time : 00:00 

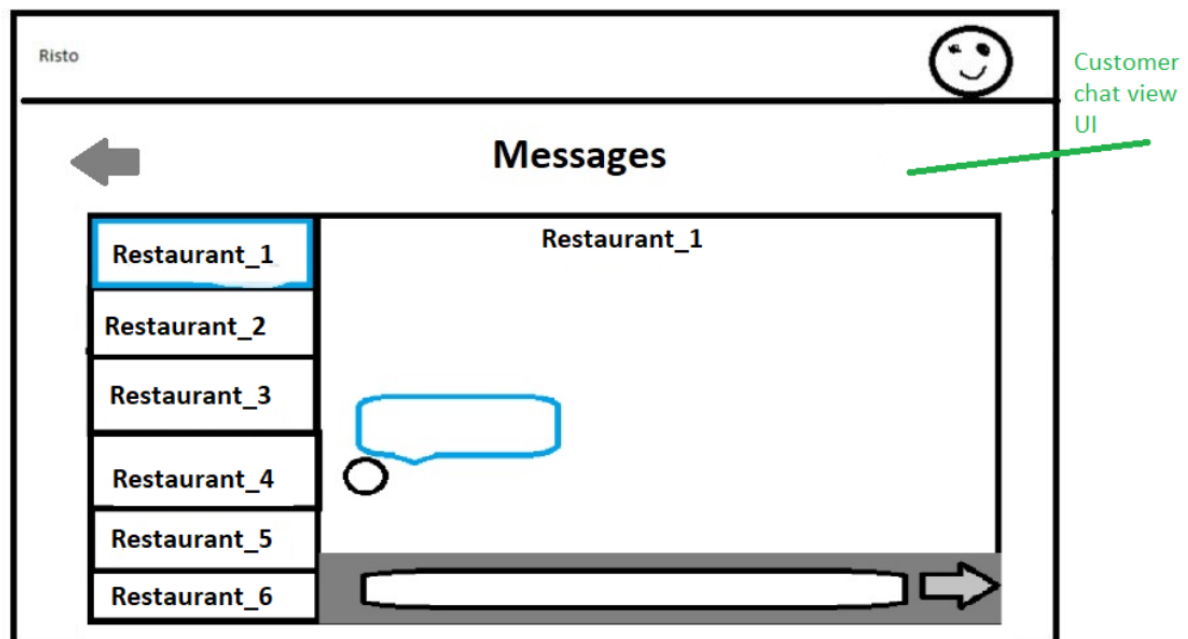
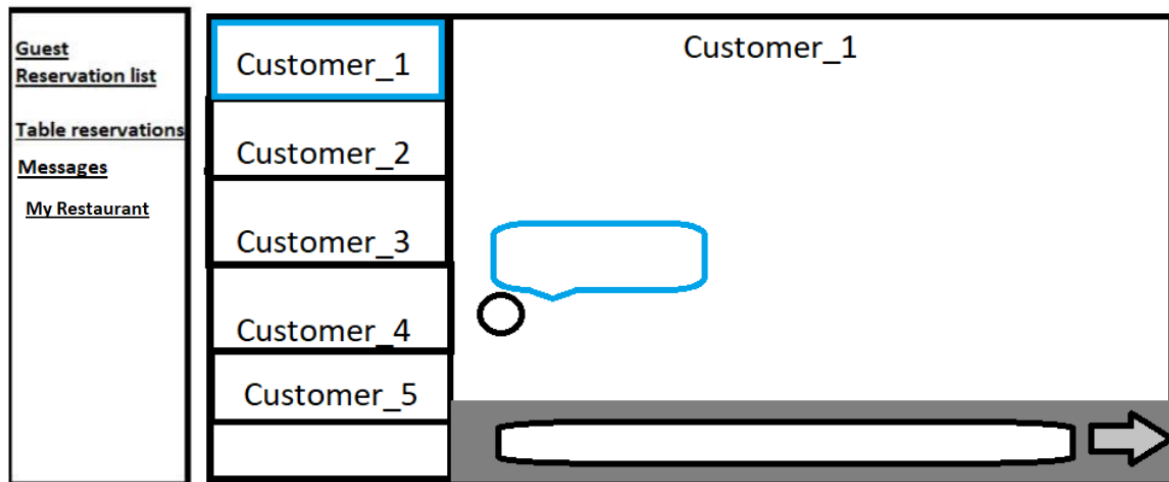
Guest  
Reservation list  
Table reservations  
Messages  
My Restaurant





#### 4. Customer and manager communication:





## 5. Ordering Food:

LOGO

Modify Food

Customer

Go Back

Photo

Restaurant Name  
Date - Time  
Number Of People

Order ID

Photo

Food Item 1  
Food Item Quantity

☒

5

+  
-

Price

.

.

.

Current Total \$

Add Food

Remove Selected

LOGO

Reservation X

Customer

Go Back

Photo

Restaurant Name  
Date - Time  
Number Of People

Current Table  
Call Waiter

Photo

Food Item 1  
Food Item Quantity

Food being processed  
Expected Wait Time: 15 mins

.

.

.

Modify

## 6. Reviewing restaurants:



LOGO

My Reviews

Customer

Go Back

Photo	Reservation 1 Date - Time	Restaurant Order ID	Rate
Photo	Reservation 2 Date - Time	Restaurant Order ID	<div>Edit Rating</div> <div>Del Rating</div>
.			
.			
.			

LOGO

Review/Edit Review Restaurant X

Customer

Go Back

How many stars would you like to give?

★

★

★

★

★

Submit

Why drove your decision?

☒ Quick Service
☒ Polite Behaviour

☒ Delicious Food
☐ Value For Money


☒ Other:

I don't want to tell.

## 7. Checking and giving back orders:

LOGO

List of Food Items


Waiter

Go Back

Food Item	Quantity	Price
Chinese	3	12,99
Donner	5	29,99
Pizza	1	25,99
<b>Total <u>Prise</u></b>		<b>68,97</b>

DONE

LOGO

List of Food Orders


Waiter

Go Back

Order No.	Table No.	Status
1	3	DONE
2	5	PENDING
3	1	PREPARING

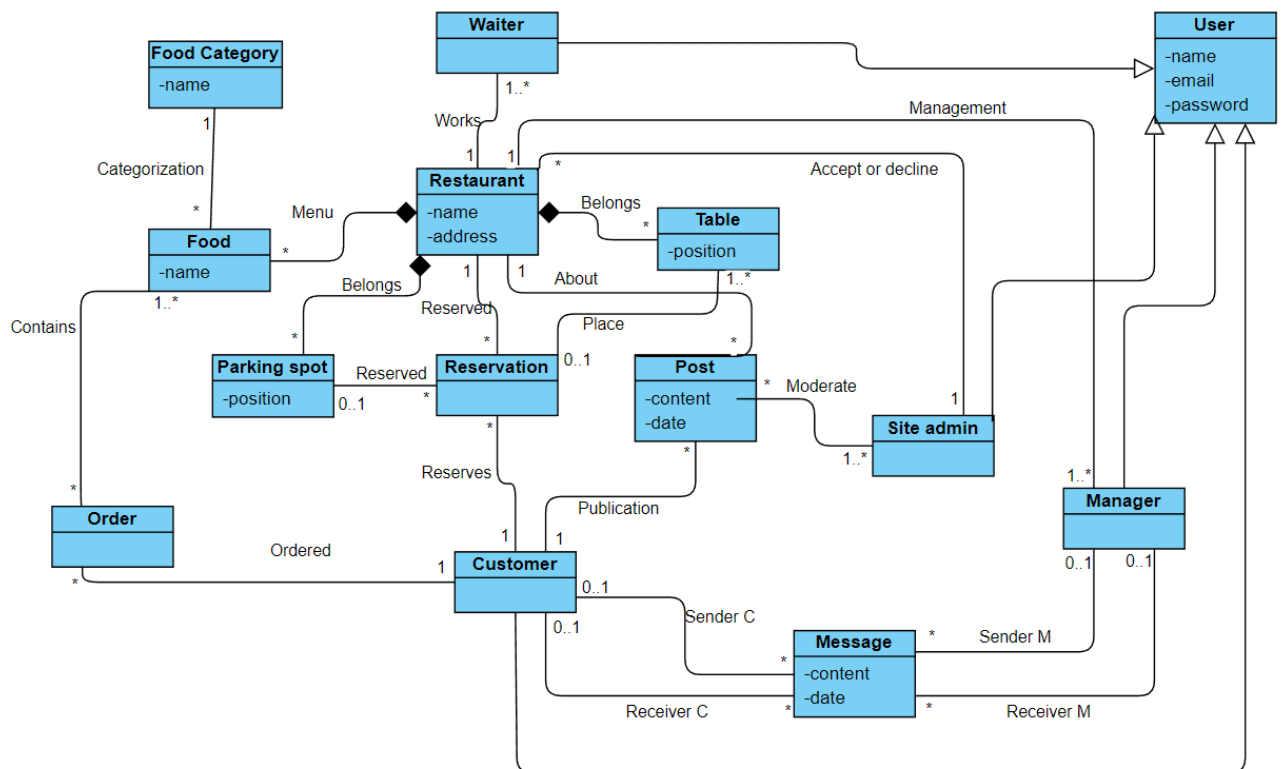
DONE



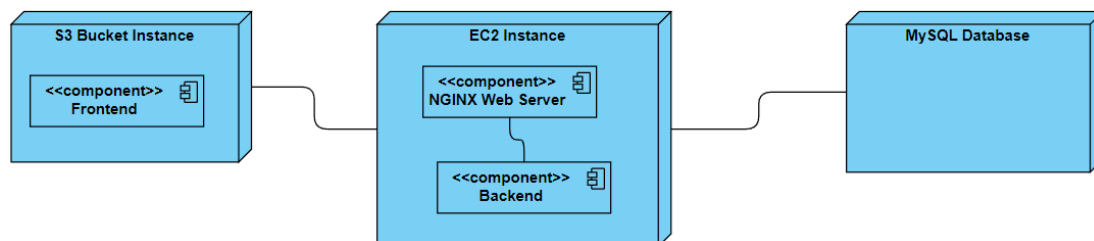
Search/filter architecture and implementation: Sorting based on location and reviews. Filtering based on postal code. We will show the results to the customer as we receive the results from the DB.

## 5. High Level UML Diagrams

a) High-level UML class diagram:



b) UML Component and deployment diagram:



## 6. Identify actual key risks for your project at this time

### • Skills risks:

- A risk is that our team is not very used to the technologies Node.js and Express. A solution to this problem would be to learn about these technologies at the same time we are developing.
- Also, not everyone has the same knowledge of programming languages such as Node.js and React etc... A solution to this could be to arrange some meetings to share our knowledge.

### • Schedule risks:

- A risk could be that we do not have enough time to implement all the functional requirements, such as the live view, which is more complex. In that case we are going to manage the time the best that we can but also we are not going to set a maximum priority for this requirement.

### • Technical risks:

- A risk could be that we do not know a formal way to clone the repository to the backend machine instance, since the repository is private. But a solution for this problem at the moment is to download the code in our machines and copy it to the AWS instance machine.
- Another risk could be that, when we are meeting, we sometimes encounter technical problems related to the hearing, probably related to the internet network. This could be solved by restarting the internet and waiting but it is inevitable to lose some time.

### • Teamwork risks:

- A risk could be that our team member Alhassane Dondo Toure can communicate better in German than in English. A solution to this problem could be that, when he does not understand something, he can ask the rest of the team and they will repeat it slowly. Also, our team member Paras can speak German better than the rest of the team, so he could explain to Alhassane in German.

### • Legal/content risks:

- A risk could be that the basic set of images provided is not enough for our application. In that case, a solution might be to use copyright free images and include their author in a reference section.

## 7. Project management

To assign our tasks we are using Trello, and we have created a dashboard for each milestone (with the lists: To do, Doing, To be revised, Done/Agreed).

In this case of Milestone 2, we have assigned each point of this document to one or more members of our team. The distribution is as follows:

1. Functional Requirements - prioritized: Noman Ali.
2. List of main data items and entities (expand as necessary): Vichitar Dagar
3. UI Mockups and Storyboards (high level only): Paras, Noman Ali, Alhassane Dondo Toure, Luis Miguel García Marín, Jesús Moreno Durán, Vichitar Dagar

4. High level Architecture, Database Organization: Jesús Moreno Durán, Vichitar Dagar.
5. High Level UML Diagrams: Jesús Moreno Durán, Luis Miguel García Marín
6. Identify actual key risks for your project at this time: Luis Miguel García Marín, Alhassane Dondo Toure.
7. Project management: Luis Miguel García Marín