Haute Ecole Condorcet : Informatique de Gestion

Projet ASP2021

DI FALCO SIMON & HUYGEBAERT AMANDINE



# L’énoncé

Le travail s’inspire de l’actualité. Votre client souhaite que vous réalisiez une plateforme web pour les restaurateurs souhaitant mettre en place un système de plats à emporter (take-away).

L’objectif de l’application est d’automatiser un maximum de tâches :

1. La plateforme est destinée à plusieurs restaurateurs.
2. Un restaurateur intéressé pourra inscrire son restaurant sur la plateforme.
3. Un restaurateur proposera des menus (par ex. : entrée-plat-dessert, 2 entrées-plat-dessert, …) et des plats à la carte.
4. Un plat à la carte a un prix. Un menu a un prix (pas une addition des prix des plats).
5. Chaque jour, le restaurateur propose deux services (service du midi et service du soir).
6. Un client passera sa commande pour un des services du jour (il ne peut commander pour un autre jour) au plus tard 2 heures avant le service.
7. Un client qui habite à proximité du restaurant pourra demander une livraison à domicile, moyennant frais supplémentaire.
8. Un client qui n’habite pas à proximité devra venir chercher ses plats sur place.
9. Pour passer la commande, le client doit créer un compte dans lequel il fournira toutes les informations nécessaires.
10. Seul un utilisateur authentifié peut passer une commande.
11. Un visiteur de la plateforme peut consulter les menus sans s’authentifier.
12. Un restaurateur devra s’authentifier pour ajouter son restaurant, gérer ses menus et gérer ses commandes.
13. …

# Procès-Verbal (réunions par ordre chronologique)

## Réunion 1

|  |  |
| --- | --- |
| *Membres du groupe présents* : DI FALCO SIMON, HUYGEBAERT AMANDINE  *DATE :* 16-03-2021  *DURÉE :* 50 minutes *LIEU :* Plateforme Discord  *OBJECTIF DE LA RÉUNION :* Mise au point au sujet du USECASE DIAGRAM et ses scénarios.  *La répartition des tâches :* Réalisation du diagramme des cas d’utilisations individuel et mise en commun. Mise au point d’un pattern à respecter au sujet des scénario et répartition des scénarios puis lecture ensemble. Dialogue facile, régulier et franc entre nous.   |  | | --- | | Répartition scénarios :  UC\_01: Consult Menu (Scénario) --> A UC\_02 : Create account (Scénario) --> S UC\_03 : Customer account UC\_04 : Restorer Account  UC\_05 : Authentification (Scénario) --> A UC\_06 : Order (Scénario) --> S UC\_07 : Validate Order (Scénario) --> A UC\_08 : Pay (Scénario) --> S UC\_09 : Cash Paiement  UC\_10 : Online Paiement UC\_11 : Giving Tip UC\_12 : Sign Restaurant (Scénario)--> A UC-13 : Menu Management (Scénario) --> S UC\_14 : Menus  UC\_15 : Dishes  UC\_16 : Manage Order (Scénario)--> A UC\_17 : Delivrery management  UC\_19 : Maintenance  UC\_20 : Account customer UC\_21 : Account restorer UC\_22 : Disconnect |   *Les problèmes rencontrés, les solutions et corrections apportées :*  SIMON : Difficulté de compréhension des use cases.  AMANDINE : Premier usecase diagram créé trop tôt. Mauvaise compréhension de ce qui m’était demandé. Jet 2 : J’ai réalisé que je n’avais pas compris :  *Extends* : Différence entre cas d’extension/point d’extension, sens de la flèche, le usecase divisé en deux.  *Solution* => Poser des questions à Mr Masset lors du cours. *Élaboration du scénario :* quelles étaient les limites du cas d’utilisation à décrire ?   *Solution* => Poser des questions à Mr Masset, demander un exemple.  *Correction* (en commun) : Mise en commun du diagramme et partage d’opinions ( prévu à l’initial )   * Questions relatives à l’énoncé : Aucune pour l’instant. |

## Réunion 2

|  |  |  |
| --- | --- | --- |
| *Membres du groupe présents* : DI FALCO SIMON, HUYGEBAERT AMANDINE  *DATE :* 21/03/21  *DURÉE :* 55 minutes *LIEU :* Plateforme Discord  *OBJECTIF DE LA RÉUNION :*  Correction du diagramme des Uses cases pour donner suite aux commentaires de monsieur Masset. Nouvelle répartition des scénarios.  *La répartition des tâches :*   |  | | --- | | Répartition  Tout a été fait en commun. | | Répartition scénarios :  UC\_01: Consult Menu 🡪 A UC\_02: Create account 🡪S  UC\_03: Authentication 🡪 A UC\_04: Add item to cart 🡪 S UC\_05: Consult cart 🡪 A  UC\_06: Order 🡪 S  UC\_07: Give an opinion 🡪 A UC\_08: Manage Account 🡪 S  UC\_09: Sign Restaurant🡪 A  UC\_10: Manage Orders 🡪 S UC\_11: Manage Menu 🡪 S |   *Les problèmes rencontrés, les solutions et corrections apportées :*  SIMON : Compréhension des extends.  AMANDINE : Mauvaise vision des choses pour l’analyse : cas de généralisations et spécialisations.  *Solution* => Discussion et mise en commun.  *Correction* (en commun) : Remise à 0 du diagramme.   * Questions relatives à l’énoncé : Aucune pour l’instant. |

## Réunion 3

|  |  |  |
| --- | --- | --- |
| *Membres du groupe présents* : DI FALCO SIMON, HUYGEBAERT AMANDINE  *DATE :* 23-03-21  *DURÉE :* 30 min *LIEU :* Discord  *OBJECTIF DE LA RÉUNION :*  Division des use cases  *La répartition des tâches :*   |  | | --- | |  | | Répartition  UC\_01: Consult Menu 🡪 A UC\_02: Create account 🡪S  UC\_03: Authentication 🡪 A UC\_04: Add item to cart 🡪 S UC\_05: Consult cart 🡪 A  UC\_06: Order 🡪 S  UC\_07: Give an opinion 🡪 A  UC\_08: Consult Orders 🡪 A UC\_09: Manage Account 🡪 S  UC\_10: Consult information 🡪 S  UC\_11: Modify information 🡪 S  UC\_12: Sign Restaurant🡪 A  UC\_13: Manage Orders 🡪 S  UC\_14: Consult orders 🡪 S  UC\_15: Update order status 🡪 S UC\_16: Manage Menu 🡪S  UC\_17: Add dish 🡪 S  UC\_18: Modify dish 🡪 S  UC\_19: Remove dish 🡪 S  UC\_20: Add menu 🡪 S  UC\_21: Modify menu 🡪 S  UC\_22: Remove menu🡪 S |   *Les problèmes rencontrés, les solutions et corrections apportées :*  SIMON :. /  AMANDINE : /    *Solution* => /  *Correction* (en commun) :  Modification du diagramme des use cases   * Questions relatives à l’énoncé : Aucune pour l’instant. |

|  |  |  |
| --- | --- | --- |
| *Membres du groupe présents* : DI FALCO SIMON, HUYGEBAERT AMANDINE  *DATE :* 30/03/21  *DURÉE :* 35 minutes *LIEU :* Plateforme Discord  *OBJECTIF DE LA RÉUNION :*  Mise en commun du diagramme de classe  *La répartition des tâches :*   |  | | --- | | Répartition  Tout a été fait en commun. | |  |   *Les problèmes rencontrés, les solutions et corrections apportées :*  SIMON : Les liens entre les classes  AMANDINE : Les liens entre les classes  *Correction* (en commun) : quelques ajouts et modifications lors de la mise en commun   * Questions relatives à l’énoncé : Aucune pour l’instant. |

## Réunion 4

# ANALYSE

## Use case diagram

### JET 1 – Mise en commun (20-03-2021)

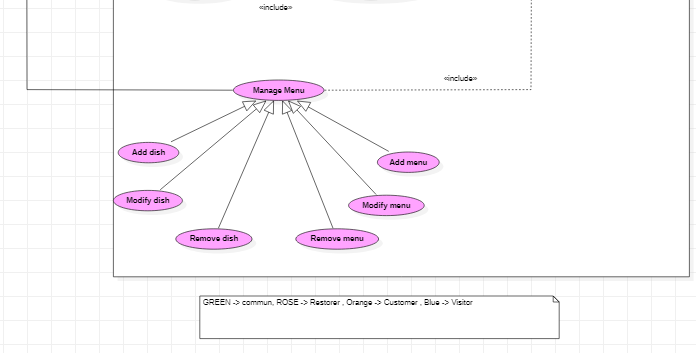
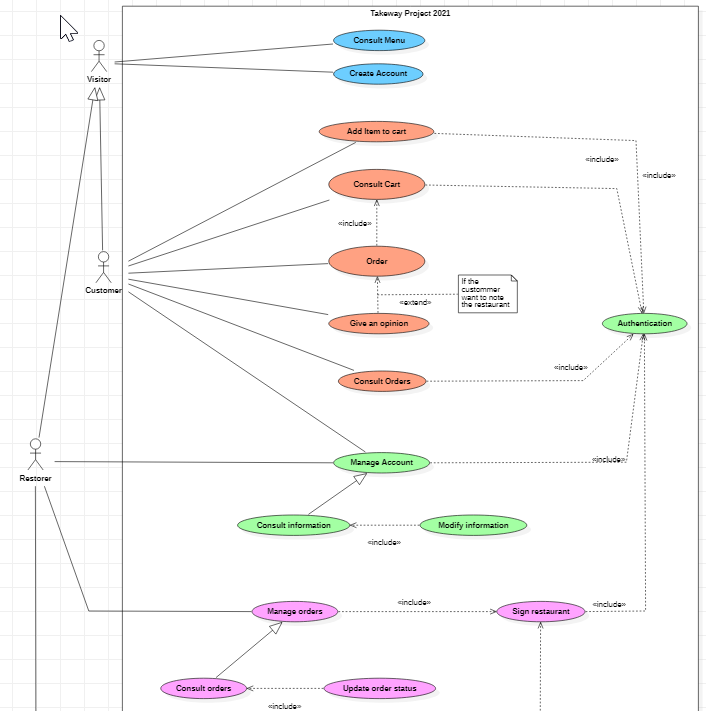
### Jet 2 – Mise en commun - proposition



### CORRECTION, après commentaires du professeur



## Correction pour le 26/03

****

##### Correction 26/03 (avec Monsieur Masset)

##### 

### Scénarios

|  |
| --- |
| ***PATTERN (ne sera pas sur le dossier final.)***  **Identifier** :  UC\_  **Actors:**  **Description:**  **Preconditions:**  **Assumptions:**  **Basic course:**  1.    **Alternate course:**  **Case A :**  **Case B :**  **Case C :**  **Inclusion cases:**  **Extension cases:** |

|  |
| --- |
| ***Consult Menu***  **Identifier**:  UC\_01  **Actors:**  Visitors  **Description:**  Everyone can consult menu, dishes, and global note.  **Preconditions:**   * 1. Application has been installed and launched.   2. Updated application.   3. Optional (authenticated).   **Assumptions:**   * 1. Network connexion established.   2. Database is online. ( if does it exist ? )   3. Restaurant and menus exist.   **Basic course:**  1. First screen: restaurant list.  User chooses the restaurant on the list and check the stars awarded to a restaurant.  2. Second screen: menus & dishes list. Applications displays all of menus and dishes, their brief descriptions and price. The user can check price of all menu or dishes. Also, he can check the overall rating and comments of other users about the restaurant.  3. Third screen: menu/dishes details. The user can check details of menu/dishes.  **Alternate course:**  **Case A:**  Database is offline.  Display error message.  **Case B:**  1. Button refresh has been selected.  Refresh list of Restaurant  **Case C:**  Screen 1/2/3. User click on Sign-up  Go to CREATE ACCOUNT (UC\_2 )  **Case D:**  Screen 1/2/3. User click on Login  Go to AUTHENTICATION (UC\_3 )  **Case E:**  2. URL comment selected.  Displays all of comments (box pop).  **Case F:**  Screen 2,3. Click on return  Go to Previous page.  **Case G:**  Screen 1,2,3. If the user is connected, adjust the page. |









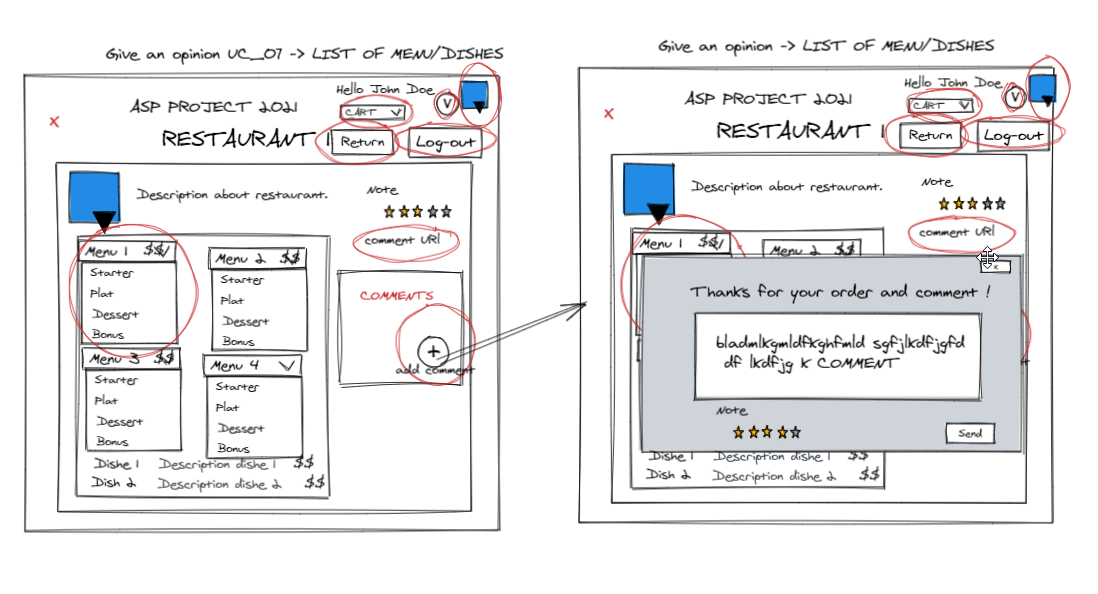
|  |
| --- |
| ***Authentication***  **Identifier**:  UC\_03  **Actors:**  Customer and restorer.  **Description:**  The user can connect to the Application.  **Preconditions:**  1. UC\_2  2. User clicked on “login” button on screen.  3. The user is on authentication page and has chosen the form (Customer or Restorer?)  **Assumptions:**  1. Network connexion established.  **Basic course:**  1. User type his e-mail and password account on form on authentication page.  2. Information are checked if the user password and the mail are correct. Also check if the person is a restorer or a costumer  3. User is connected as first name & last name  4. Go to main screen (list of restaurants)  **Alternate course:**  **Case A:**  2. Username is incorrect  Display the error message « Your username doesn’t exist. »  **Case B:**  2. Password is incorrect  Display the error message « Your password is incorrect. »  **Case C:**  2. Username & password are correct AND the type of account is CUSTOMER  4.Connected as CUSTOMER  **Case D:**  2. Username & password are correct AND the type of account is RESTORER  4.Connected as RESTORER  **Case E:**  Button return has been selected.  Go to main screen.  **Case F: (A DISCUTER)**  Button “forgot password” pressed.  The user will receive an email to reset the password.  **Case G:**  Button “logout” pressed everywhere in the application when the user is connected.  Disconnect the user.  **Inclusion cases**  Add Item to cart (UC\_04)  Consult Cart (UC\_05) Manage account (UC\_09) Sign restaurant (UC\_12) |



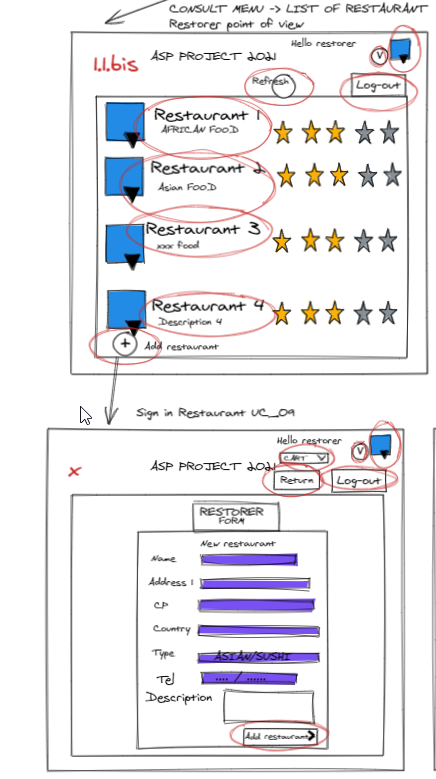
|  |
| --- |
| ***Consult Cart***  **Identifier**:  UC\_05  **Actors:**  Customer  **Description:**  The customer can check his cart at every moment or page.  **Preconditions:**  1. Customer authenticated. 2. Customer is on cart screen  **Assumptions:**  1. Network connexion established.  **Basic course:** 1. The total price is automatically calculated.  2. Customer can quickly check the content of his cart on the cart page.  3. Customer pass his order if his cart satisfying him. Go to Order screen.  **Alternate course:**  **Case A:**  1. Customer press (-) button.  Delete one item on the cart.  **Case B:**  2. Customer press “select all”  All the items on the cart are selected.  **Case C:**  2. Customer press “delete all” (select all activated)  All the items on the cart are delete. Price is recalculated.  **Case D:**  2. Customer selected many items and click on delete.  The items selected on the cart are delete. Price is recalculated.  **Case E:**  2. Customer selected many items and click on delete.  The items selected on the cart are delete. Price is recalculated.  **Case F:**  Button return has been selected.  Go to main screen.  **Inclusion cases**  Authentication (UC\_03)  Order (UC\_06) |



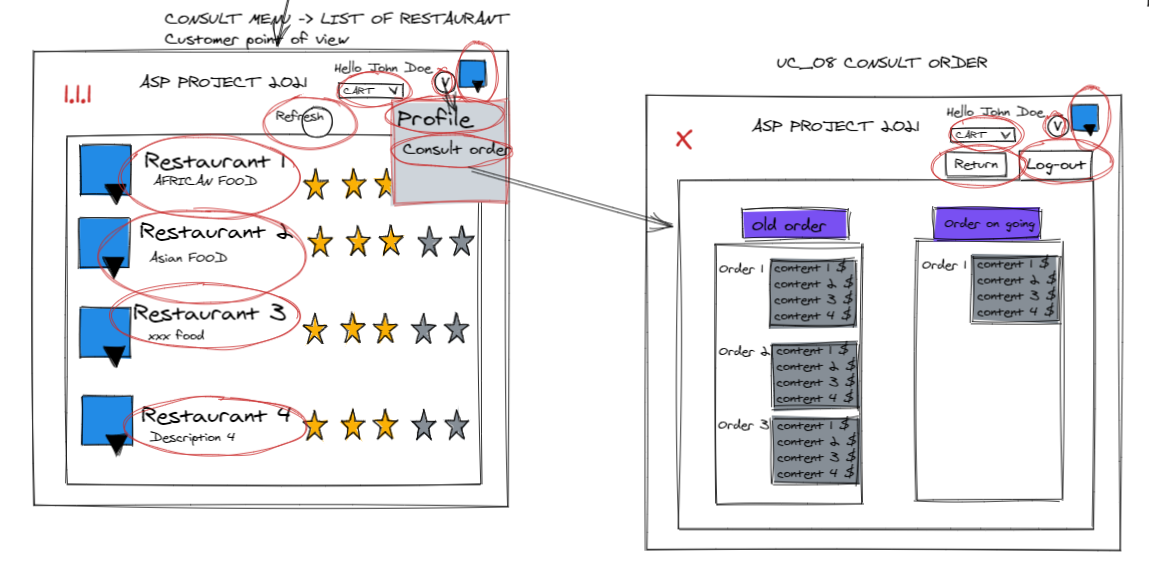
|  |
| --- |
| ***Give an opinion***  **Identifier**:  UC\_07  **Actors:**  Customer  **Description:** After an order, a customer can give an opinion about a restaurant. For each order it is possible to leave a comment.  **Preconditions:**  1. Order UC\_06 2. Order verified.  3. The customer should have at least one order in this restaurant.  **Basic course:** 1. Customer is on list of menu/dishes screen.  2. Customer click on (+)” add comment” button.  3. A windows pop.  4. The customer can type his opinion and choose number of stars and send.  5. Global note (stars) is recalculated.  6. The application knows that a comment has already been given for this order.   7. The message is saved and appears with the other comments. Return to the list of menus/dishes.  **Alternate course:**  **Case A:**   * 1. Stars field is empty.   Display error message.  **Case B:**  4. Press [X] on comment screen.  Quit comment screen and return at list of menus/dishes.  **Case C:**  Button return has been selected.  Go to main screen. |



|  |
| --- |
| ***Sign Restaurant***  **Identifier**:  UC\_12  **Actors:**  Restorer  **Description:**  The restorer can add his restaurant on the list. It is possible to add several restaurants for a restorer.  **Preconditions:**  1. Restorer authenticated. 2. Clicked on “add restaurant” on consult menu screen while being Restorer account.  **Basic course:** 1. The restorer types the name, the address (cp/country/), the restaurant telephone, the type of food and a description on the “add restaurant form”  2. The restorer click on “add restaurant”.  3. Verification of data field.  4. The information are saved and and increment ID created  5. The restaurant is added to the list and the restorer and the application go to the list of restaurant screen.  **Alternate course:**  **Case A:**  3. Field entered incorrect or empty  Display error message and ask to restorer to enter information.  **Inclusion cases**  Authentication (UC\_03)  Manage order (UC\_13) Manage Menu (UC\_16) |



|  |
| --- |
| ***Consult orders***  **Identifier**:  UC\_08  **Actors:**  Customer  **Description:**  The customer can check his order on going and old order.  **Preconditions:**   1. Customer has been clicked on option button next to his name and has chosen “consult order”  **Basic course:**   1. Customer is on consult order screen. 2. The application generates/searches the order for this customer and displays. 3. Customer can check his order.     **Alternate course:**  **Case A:**  An order is finished when the customer check his order on going.  The application updates the information. Order on going will be on old order.  **Case B:** The customer click triangle button  The list of order is order by date of order.  **Inclusion cases:**  UC\_03 (Authentication) |



|  |
| --- |
| ***Create Account***  **Identifier**:  UC\_02  **Actors:**  Visitor  **Description:**  In this use case, the visitor will create an account. He will have the choice between creating a restorer account or a customer account.  **Preconditions:**  The user is on the creating account page.  **Basic course:**  1. The visitor has the choice between a restorer or customer account.  2. In both account he will have to fill a form.  4. He clicks on the button “Validate” to validate the creation of account.  5. The account is created, and the user (restorer or customer) is connected.    **Alternate course:**  **Case A :**  2.The visitor chose the customer account.  3. He must fill a form (name, last name, email, address, phone  number, password, gender).    **Case B :**  2.The visitor has chosen the restorer account.  3. He must fill a form (name, last name, email, address, phone  number, TVA number, password).  6. A page appears to proposal to sign a restaurant: UC\_12.  **Case C :**  2. The visitor decides to cancel the creation by clicking on a button.  **Case D:**  3. He must fill a form (name, last name, email, address, phone  number, TVA number, password).  4. He clicks on the button “Validate” to validate the creation of account.  5. The form is not correctly filled, the error(s) is/are pointed out. |

|  |
| --- |
| ***Add Item to cart***  **Identifier**:  UC\_04  **Actors:**  Customer  **Description:**  The aim of this use case is to add an item to the cart.  **Preconditions:**  The user is authenticated.  **Basic course:**  1. The customer is consulting the menu and decide to add an item to the cart.  2. He clicks on the “add to cart” button.  3. The item is added to the cart    **Alternate course:**  4. The customer decides to remove the/an item of the cart 🡪 UC\_05      **Inclusion cases:**  UC\_03: Authentication. |

|  |
| --- |
| ***Order***  **Identifier**:  UC\_06  **Actors:**  Customer  **Description:**  In this use case, the user will order what he has added to his cart.  **Preconditions:**  The cart is not empty.  **Basic course:**  1. The customer can validate his cart by clicking on a button.  2. A new page summarizes the order and asks for which address is this order  (the account address or an another adress).  3. The customer validates the order by clicking on a button.  4. A new page confirm that the order is validated and on this page the customer  can leave a comment and note the restaurant (with stars or numbers) ->  UC\_07.  **Alternate course:**    **Case A :**  1. The customer can come back to the menu (to add item for example) by  clicking on a button.  2. He is back on the menu page.  **Case B :**  3. The customer decides to cancel the order validation by clicking on a  button.  4.The customer is back on the cart page.  **Inclusion cases:**  UC\_05: Consult cart  **Extension cases:**  UC\_07: Give an Opinion |

|  |
| --- |
| ***Manage account***  **Identifier**:  UC\_09  **Actors:**  Customer, restorer  **Description:**  In this use case, the user can modify/consult information about his account.  **Preconditions:**  The user is on the manage account page (by clicking on his name in the upper right corner).  **Basic course:**  1. The user can consult and by this way modify his information.  **Alternate course:**  **Case A :**  1. The user wants to come back to the previous page by clicking on  The cancel button.    **Case B :**  2. The user decides consult his information -> Go to UC\_10 (Consult  Information)    **Inclusion cases:**  UC\_03: Authentication |

|  |
| --- |
| ***Consult Information***  **Identifier**:  UC\_10  **Actors:**  Customer, restorer  **Description:**  In this use case, the user can see all the information that he has entered during the registration.  **Basic course:**  1. The user can consult the information that he has entered during the  Registration (name, last name, address, phone number).    **Alternate course:**  **Case A :**  1. The user wants to come back to the previous page by clicking on  the comeback button.    **Case B :**  2. The user decides to modify the information by clicking on the modify  button 🡺 Go to UC\_11(Modify Information) |

|  |
| --- |
| ***Modify Information***  **Identifier**:  UC\_11  **Actors:**  Customer, restorer  **Description:**  In this use case, the user can modify all the information that he has entered during the registration.  **Preconditions:**  The user has clicked on a button to modify the information on the consult information page.  **Basic course:**  1. The user can modify the information that he has entered during the  Registration (name, last name, address, phone number).  2. The user modifies the information that he wants.  3. The user validates the modifications by clicking on a button.  4. The modifications have been validated.    **Alternate course:**  **Case A:**  1. The user wants to come back to the previous page by clicking on  the comeback button.  2. The user is back on the previous page.  **Case B:**  3. The user wants to cancel the modifications by clicking on  a button.  4. The user is redirected on the consult page.    **Case C:**  4. An alert (or something else) confirms that the modification failed with  the details of the errors (and pointing out the error(s))  **Inclusion cases:**  UC\_10: Manage Account |
| ***Manage Orders***  **Identifier**:  UC\_13  **Actors:**  Restorer  **Description:**  This use case is aimed to manage the orders that the restaurant received.  **Preconditions:**  The restaurant is signed.  **Basic course:**  1. The restorer chooses to consult the orders -> UC\_12(Consult orders)    **Alternate course:**    **Case A:**  1.The user can come back to the previous page by clicking on a button.    **Inclusion cases:**  UC\_12: Sign Restaurant |

|  |
| --- |
| ***Consult Orders***  **Identifier**:  UC\_14  **Actors:**  Restorer  **Description:**  In this use case, the restorer can consult all the orders that his restaurant received. There is two different categories Recent orders (about the current day), All orders (historic).  **Preconditions:**  The restaurant is signed and the user is on the consult order page.  **Basic course:**  1. The restorer can choose to see all the orders and see the recent  orders.    **Alternate course:**  **Case A:**  2. The restorer chose to see all the orders by clicking on this possibility.  3. A page shows all the orders ordered by day and hour.  4. The restorer can manage the status of the order (only for the order of  the day)  🡪UC\_13(Update Order Status).  5.The user can come back to the previous page by clicking on a button.  **Case B:**  2. The restorer chose to see the recent orders by clicking on  this possibility.  3. The restorer can manage the status of the order  -> UC\_13(Update Order Status).  4.The user can come back to the previous page by clicking on a button.  **Case C:**  2.The user can come back to the previous page by clicking on a button.    **Inclusion cases:**  UC\_11: Manage order |

|  |
| --- |
| ***Update Order Status***  **Identifier**:  UC\_15  **Actors:**  Restorer  **Description:**  In this use case, the restorer can update the order status (received, in progress, ready).  **Preconditions:**  The restaurant is signed, and the restorer is in the one of the consult pages.  **Basic course:**  1. The restorer can update the order status by three different states:   * + - Received: the restaurant has received the customer order     - In progress: the order is prepared     - Finish : the order is ready for the client.   2. The restorer change the status by clicking on the corresponding state.  3. The status is updated.  **Alternate course:**    **Case A:**  1.The user can come back to the previous page by clicking on a button.    **Inclusion cases:**  UC\_12: Consult Orders |

|  |
| --- |
| ***Manage menu***  **Identifier**:  UC\_16  **Actors:**  Restorer  **Description:**  In this use case, the restorer can manage his menu by adding/ removing/ modifying his menu: dishes and menus. The restorer can also prepare the menus for the two services of the day.  **Preconditions:**  The restaurant is signed, and the restorer is on the menu management page.  **Basic course:**  1. The restorer can modify all the information about an item of the menu and  he can also prepare the menus for the two services of the day.    **Alternate course:**  **Case A:**  2.The restorer decides to add a dish/menu by clicking on the adding  button.  3. He can enter all the different information: name, price, description,  the supplements (optional), hour of the service.  4. He validates the item by clicking on the validate button.  5. The item is added to the menu.  **Case B:**  2.The restorer decides to modify a dish/menu by clicking on the modifying  button.  3. He can modify all the different information: name, price, description,  the supplements (optional).  4. He validates the modification item by clicking on the validate button.  5. The item is added to the menu.  **Case C:**  2.The restorer decides to modify/add a dish or a menu by clicking on the  adding/modifying.  button.  3. He can modify all the different information: name, price, description,  the supplements (optional).  4. He decides to cancel the modification/adding by clicking on a button.  5. The restorer is back on the menu page.  **Inclusion cases:**  UC\_09: Sign Restaurant |

|  |
| --- |
| ***Add dish***  **Identifier**:  UC\_17  **Actors:**  Restorer  **Description:**  In this use case, the restorer adds dish to his menu.  **Preconditions:**  The restaurant is signed, and the restorer is on the menu management page.  **Basic course:**  1. The restorer can add a dish by clicking on a button.  2. The restorer must fill a form: name, description, price, hour of service,  supplements (optional).  3. The restorer validates the adding of the dish.  4. The dish is added to the menu.  5. The restorer is back on the management menu page.  **Alternate course:**  **Case A:**  2.The restorer decides to cancel the adding by clicking on a button.  3. The restorer is back on the management menu page.    **Case B:**  3.The restorer decides to cancel the adding by clicking on a button.  4. The restorer is back on the management menu page.  **Case C:**  4. The adding failed because of error(s): the error(s) is/are pointed out.  5. The user corrects the error(s).  3. The restorer validates the adding of the dish.  4. The dish is added to the menu.  5. The restorer is back on the management menu page. |

|  |
| --- |
| ***Modify dish***  **Identifier**:  UC\_18  **Actors:**  Restorer  **Description:**  In this use case, the restorer modifies dish from his menu.  **Preconditions:**  The restaurant is signed, and the restorer is on the menu management page.  **Basic course:**  1. The restorer can modify a dish by clicking on a button.  2. He can modify all the different information: name, price, description,  the supplements.  3. The restorer validates the modification(s) of the dish.  4. The dish is modified.  5. The restorer is back on the management menu page.  **Alternate course:**  **Case A:**  2.The restorer decides to cancel the modification by clicking on a button.  3. The restorer is back on the management menu page.    **Case B:**  3.The restorer decides to cancel the modification by clicking on a button.  4. The restorer is back on the management menu page.  **Case C:**  4. The modification failed because of error(s): the error(s) is/are pointed out.  5. The user corrects the error(s).  3. The restorer validates the modification(s) of the dish.  4. The dish is modified.  5. The restorer is back on the management menu page. |

|  |
| --- |
| ***Remove dish***  **Identifier**:  UC\_19  **Actors:**  Restorer  **Description:**  In this use case, the restorer removes dish from his menu.  **Preconditions:**  The restaurant is signed and the restorer is on the menu management page.  **Basic course:**  1. The restorer can remove a dish by clicking on a button.  2. The restorer validates the removal of the dish.  3. The dish is deleted.  4. The restorer is back on the management menu page.  **Alternate course:**  **Case A:**  2.The restorer decides to cancel the removal by clicking on a button.  3. The restorer is back on the management menu page. |

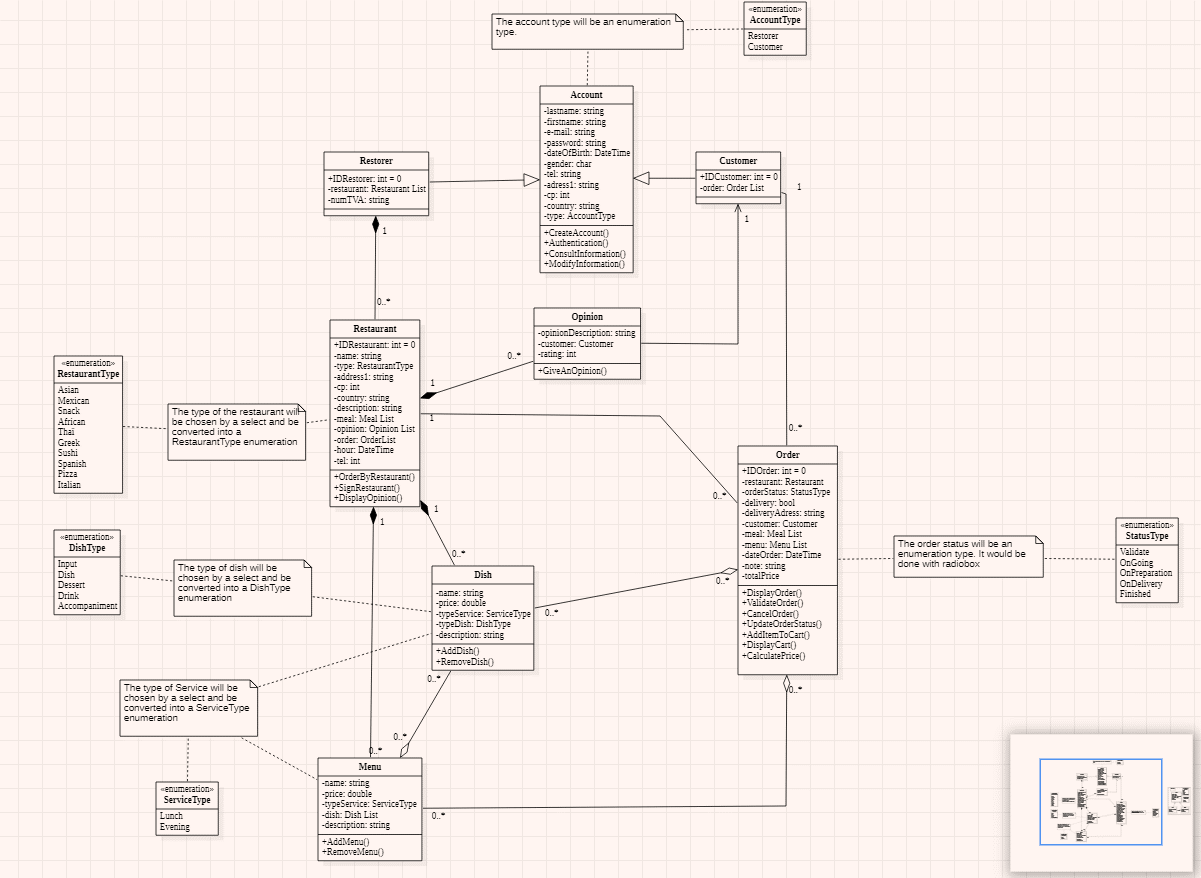
|  |
| --- |
| ***Add menu***  **Identifier**:  UC\_20  **Actors:**  Restorer  **Description:**  In this use case, the restorer adds a menu to the restaurant menu.  **Preconditions:**  The restaurant is signed, and the restorer is on the menu management page.  **Basic course:**  1. The restorer can add a menu by clicking on a button.  2. The restorer must add dish to compose the menu, he can add supplements  (optional).  3. The restorer sets the normal price (without all the supplements).  3. The restorer validates the adding of the menu.  4. The menu is added to the restaurant menu.  5. The restorer is back on the management menu page.  **Alternate course:**  **Case A:**  2. The restorer decides to cancel the adding of a menu by clicking on a  button.  3. The restorer is back on the management menu page.    **Case B:**  3.The restorer decides to cancel the adding by clicking on a button.  4. The restorer is back on the management menu page.  **Case C:**  4. The adding failed because of error(s): the error(s) is/are pointed out.  5. The user corrects the error(s).  6. The restorer validates the adding of the menu.  7. The menu is added to the restaurant menu.  8. The restorer is back on the management menu page. |

|  |
| --- |
| ***Modify menu***  **Identifier**:  UC\_21  **Actors:**  Restorer  **Description:**  In this use case, the restorer can modify a menu from his restaurant menu.  **Preconditions:**  The restaurant is signed and the restorer is on the menu management page.  **Basic course:**  1. The restorer can modify a menu by clicking on a button.  2. He can modify all the different information: dishes inside, price,  the supplements.  3. The restorer validates the modification(s) of the menu.  4. The menu is modified.  5. The restorer is back on the management menu page.  **Alternate course:**  **Case A:**  2.The restorer decides to cancel the modification by clicking on a button.  3. The restorer is back on the management menu page.    **Case B:**  3.The restorer decides to cancel the modification by clicking on a button.  4. The restorer is back on the management menu page.  **Case C:**  4. The modification failed because of error(s): the error(s) is/are pointed out.  5. The user corrects the error(s).  3. The restorer validates the modification(s) of the menu.  4. The menu is modified.  5. The restorer is back on the management menu page. |

|  |
| --- |
| ***Remove menu***  **Identifier**:  UC\_22  **Actors:**  Restorer  **Description:**  In this use case, the restorer removes menu from his restaurant menu.  **Preconditions:**  The restaurant is signed, and the restorer is on the menu management page.  **Basic course:**  1. The restorer can remove a menu by clicking on a button.  2. The restorer validates the removal of the menu.  3. The menu is deleted.  4. The restorer is back on the management menu page.  **Alternate course:**  **Case A:**  2.The restorer decides to cancel the removal by clicking on a button.  3. The restorer is back on the management menu page. |

## Class diagram

### JET 1 présenté ( - - )



Note : Bien zoomer

#### Correction J1

## Sequence diagram

### JET 1 présenté ( - - )

#### Correction J1

# Conclusion