De la conception UML vers son implémentation JAVA

Sujet : Commande de son menu « fast food » en ligne

Nous souhaitons réaliser un logiciel permettant la prise d'une commande d'un menu sur internet jusqu'à son retrait dans un restaurant.

Comment passer ma commande en ligne ?

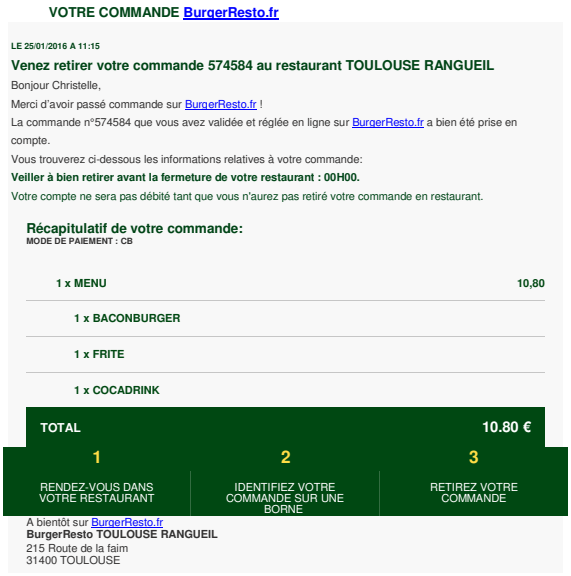
Il faut avoir un compte client, à défaut il faut en créer un. Ci-dessous le genre d'interface permettant de se connecter afin de passer sa commande.

Après s'être connecté, le client peut passer sa commande.
Pour simplifier le logiciel à réaliser, le client ne pourra commander que des menus. Ils seront obligatoirement composés d'un burger, d'un accompagnement et d'une boisson.
Le client entre ensuite ses coordonnées bancaires pour valider sa commande.

Apparaîtra alors à l'écran un fichier à imprimer contenant un numéro de commande, la date et l'heure de la commande et le récapitulatif de la commande.

Comment récupérer ma commande en ligne ?

Aux bornes de commande
- Sélectionnez la touche « Retrait commande en ligne »,
- Saisissez votre numéro de commande à 6 chiffres sur le clavier virtuel,
- Validez votre commande,
- Récupérez votre commande au comptoir dédié.



Que se passe-t-il si je ne viens pas retirer ma commande dans la journée ?

Votre commande en ligne expirera automatiquement à minuit, le jour où vous avez effectué votre commande. Vous ne serez donc pas débité(e), car votre compte ne l'est pas tant que vous n'êtes pas venus valider votre commande sur une borne en restaurant.

Comment puis-je consulter l'historique de mes commandes ?

Pour consulter vos commandes passées, connectez-vous à votre compte personnel sur le site internet. Et rendez-vous dans la rubrique « Historique de commandes ».

Travail à effectuer pour le premier TD

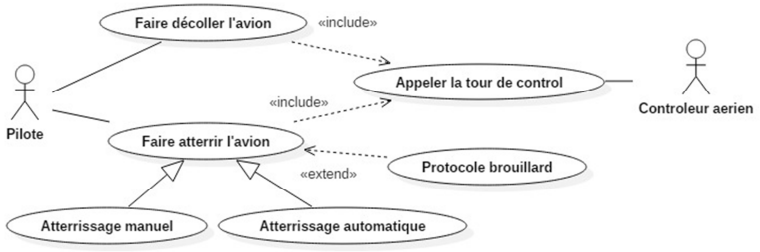
A partir du sujet, identifier les différents :
- acteurs qui utiliseront le logiciel.
- cas d'utilisation permettant de répondre à l'ensemble des fonctionnalités demandées.

Donnez les diagrammes séquences systèmes des différents cas pour permettre la création d'un prototype.

Rappel de cours

Diagramme de cas

Il montre les interactions fonctionnelles entre les acteurs et le système que l'on souhaite étudier.



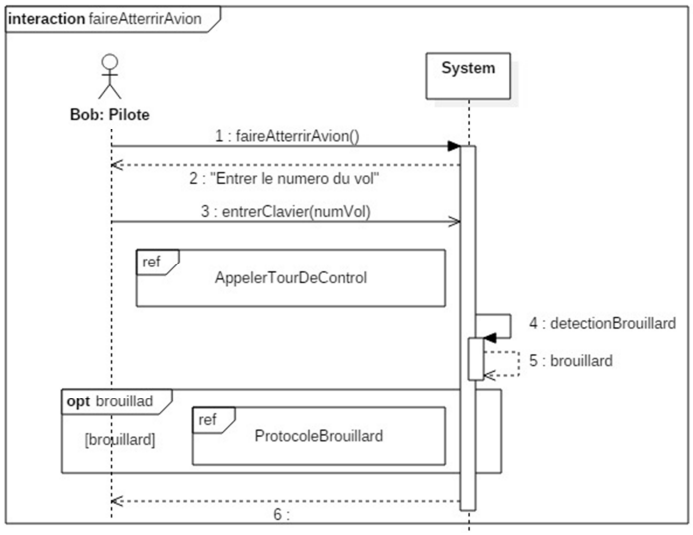
ATTENTION un diagramme de cas est un diagramme statique, il ne décrit pas l'enchainement entre les cas. Par exemple on n'exprime pas ici le fait qu'il faut que l'avion ait décollé pour qu'il puisse atterrir.

Inclusion : le cas d'utilisation inclus de façon obligatoire un autre cas.
Extension : le cas d'utilisation inclus de façon optionnelle un autre cas.
Généralisation : les cas d'utilisation enfants héritent de la description de leur parent.

Diagramme de séquence système

Il permet de décrire les cas d'utilisation en montrant les interactions entre le (ou les) acteur(s) et le système. Ce diagramme est dynamique, c'est-à-dire qu'il prend en compte les enchaînements des interactions.
Pour chaque cas d'utilisation nous créons un diagramme de séquence système.

Nous devons donc retrouver les mêmes acteurs qui participent au cas d'utilisation.



Le premier message :
- est un appel à la méthode *faireAtterrirAvion*,
- va de l'acteur vers le système,
- porte le nom du cas,
- déclenche une activation dans le système qui dure jusqu'à la fin du cas.

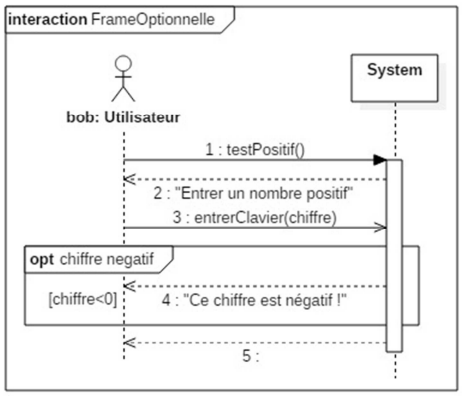
Les cas inclus dans le diagramme de cas d'utilisation se retrouvent en tant que référence (cadre avec le mot clef « ref ») dans le diagramme de séquence système.
Par contre pour les extensions (« extend »), les références sont dans un cadre optionnel (opt) ou un cadre alternatif (alt).

Le system communique avec l'acteur en affichant du texte (noté entre guillemets). Cette interaction est notée avec une flèche en pointillé (⇢).

Les données utilisées par le système ou affichées à l'utilisateur, proviennent obligatoirement soit de l'acteur, soit du système :
- l'acteur entre des données en tapant du texte sur le clavier (entrerClavier(texte)) ou en utilisant un objet graphique (exemple : clickBouton(nomBouton)). Cet envoi de données est noté avec une flèche ouverte (→),
- le système peut s'envoyer un message à lui-même qui déclenchera une activité. Suite à cette activité le système retourne un résultat (une donnée) qui sera exploitée par la suite.

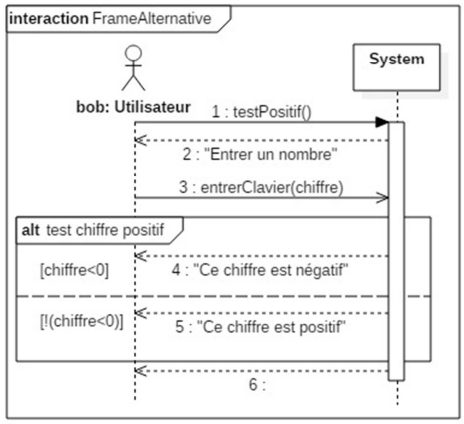
Ces données pourront servir de garde dans les cadres d'interaction (opt : fragment ne s'exécutant que si la garde est vrai, alt : seul le fragment possédant la garde vrai s'exécute, loop : le fragment s'exécute tant que la garde est vrai), être affichées ou être fournies au système.

Les frames



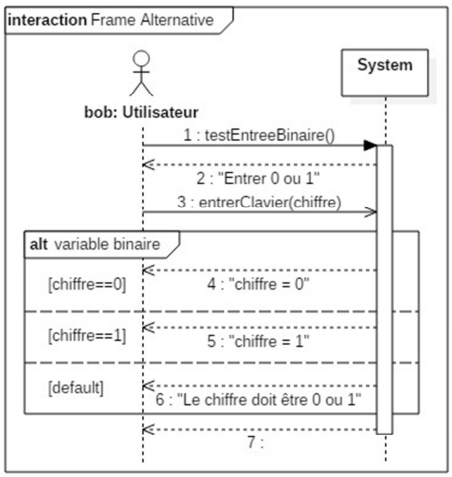
Code de l'algorithme

```
1   VARIABLES
2     chiffre EST_DU_TYPE NOMBRE
3   DEBUT_ALGORITHME
4     AFFICHER "Entrer un nombre positif"
5     LIRE chiffre
6     SI (chiffre<0) ALORS
7       DEBUT_SI
8       AFFICHER "Ce chiffre est négatif"
9       FIN_SI
10  FIN_ALGORITHME
```



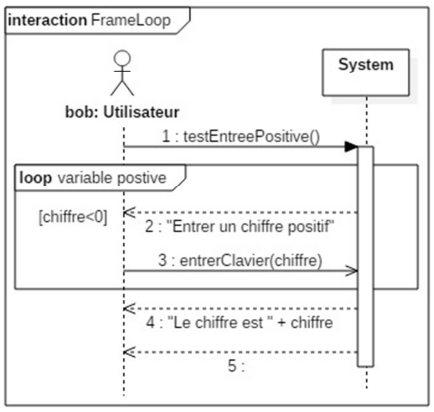
Code de l'algorithme

```
1   VARIABLES
2     chiffre EST_DU_TYPE NOMBRE
3   DEBUT_ALGORITHME
4     AFFICHER "Entrer un nombre"
5     LIRE chiffre
6     SI (chiffre<0) ALORS
7       DEBUT_SI
8       AFFICHER "Ce chiffre est négatif"
9       FIN_SI
10      SINON
11        DEBUT_SINON
12        AFFICHER "Ce chiffre est positif"
13        FIN_SINON
14  FIN_ALGORITHME
```



Code de l'algorithme

```
1   VARIABLES
2     chiffre EST_DU_TYPE NOMBRE
3   DEBUT_ALGORITHME
4     AFFICHER "Entrer 0 ou 1"
5     LIRE chiffre
6     SELON (chiffre) FAIRE
7       DEBUT_SELON
8       CAS 0 : AFFICHER "chiffre = 0"
9       CAS 1 : AFFICHER "chiffre = 1"
10      SINON : AFFICHER "Le chiffre doit être 0 ou 1 !"
11      FIN_SELON
12  FIN_ALGORITHME
```



Code de l'algorithme

```
1   VARIABLES
2     chiffre EST_DU_TYPE NOMBRE
3   DEBUT_ALGORITHME
4     TANT_QUE (chiffre<0) FAIRE
5       DEBUT_TANT_QUE
6       AFFICHER "Entrer un chiffre positif"
7       LIRE chiffre
8       FIN_TANT_QUE
9     AFFICHER "Le chiffre est"
10    AFFICHER chiffre
11  FIN_ALGORITHME
```