

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.



INSCRIVEZ-VOUS
POUR POUVOIR COMMANDER EN LIGNE ET BÉNÉFICIER DE PROMOTIONS

Connexion via Facebook

OU

Tous les champs sont obligatoires

☐ M ☐ Mme

Prénom

Nom

JJ MM AAAA

Code postal

Mobile

Email

Mot de passe

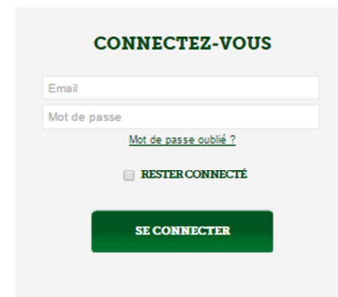
Confirmer le mot de passe

☐ J'accepte de recevoir des offres et des informations par SMS de la part de McDonald's.

☐ J'accepte de recevoir des offres et des informations par email de la part de McDonald's.

☐ RESTER CONNECTÉ

SUIVANT



CONNECTEZ-VOUS

Email

Mot de passe

[Mot de passe oublié ?](#)

☐ RESTER CONNECTÉ

SE CONNECTER

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

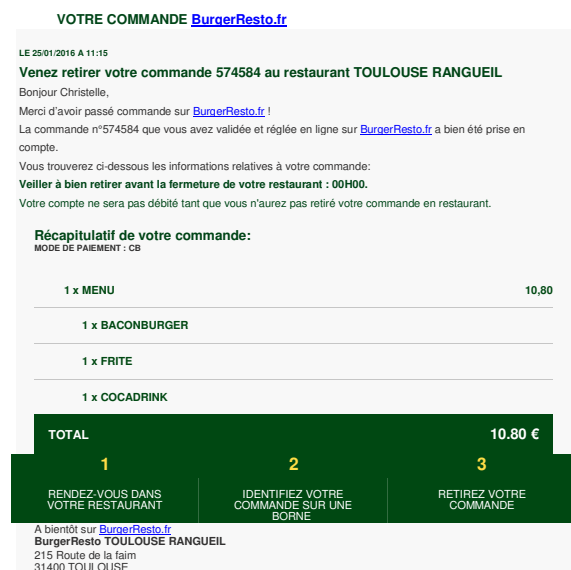
- Sélectionnez la touche « Retrait commande en ligne »,
- Saisissez votre numéro de commande à 6 chiffres sur le clavier virtuel,
- Validez votre commande,
- Récupérez votre commande au comptoir dédié.

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 ».



VOTRE COMMANDE [BurgerResto.fr](#)

LE 25/01/2016 A 11:15

Venez retirer votre commande 574584 au restaurant TOULOUSE RANGUEIL

Bonjour Christelle,

Merci d'avoir passé commande sur [BurgerResto.fr](#) !

La commande n°574584 que vous avez validée et réglée en ligne sur [BurgerResto.fr](#) a bien été prise en compte.

Vous trouverez ci-dessous les informations relatives à votre commande:

Veillez à bien retirer avant la fermeture de votre restaurant : 00H00.

Votre compte ne sera pas débité tant que vous n'aurez pas retiré votre commande en restaurant.

Récapitulatif de votre commande:
MODE DE PAIEMENT : CB

1 x MENU	10,80
1 x BACONBURGER	
1 x FRITE	
1 x COCADRINK	
TOTAL	10.80 €

1

RENDEZ-VOUS DANS VOTRE RESTAURANT

2

IDENTIFIEZ VOTRE COMMANDE SUR UNE BORNE

3

RETIREZ VOTRE COMMANDE

A bientôt sur [BurgerResto.fr](#)
BurgerResto TOULOUSE RANGUEIL
215 Route de la faim
31400 TOULOUSE

Complément du sujet

Pour simplifier une commande d'un client sera composée uniquement d'un hamburger, d'un accompagnement et d'une boisson. Une fois que sa commande a été validée, le client reçoit son numéro de commande et sur les écrans des cuisiniers apparaît une nouvelle commande qui sera normalement à faire dans la journée.

Affichage Client Veuillez sélectionner le numéro de votre hamburger : 1 : baconBurger 2 : chickenBurger 3 : cheeseBurger 2 Veuillez selectionner le numero de votre accompagnement : 1 : frites 2 : pommesChips 1 Veuillez selectionner le numero de votre boisson : 1 : coca 2 : orangeBulles 1 Votre numéro de commande est le : 1	Affichage Cuisinier Commande no 1 : chickenBurger, frites, coca
---	---

Cette commande sera stockée dans une base de données jusqu'au moment où le client passe la récupérer au restaurant grâce à son numéro. A ce moment sa commande sera déplacée dans son historique de commandes.

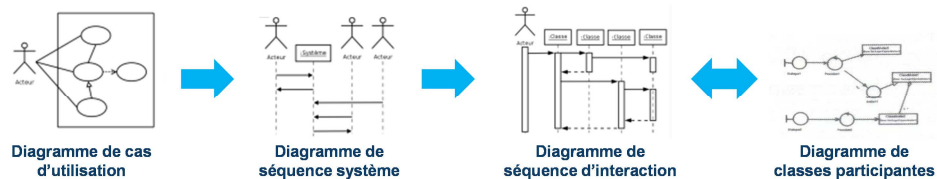
A la fin de la journée, quand le gérant procède à l'archivage, toutes les commandes qui n'ont pas été récupérées sont supprimées de la base de données.

Rappel de cours

Démarche de modélisation

Particularités des diagrammes :

- 1 control par cas
- 1 boundary par cas et par acteur

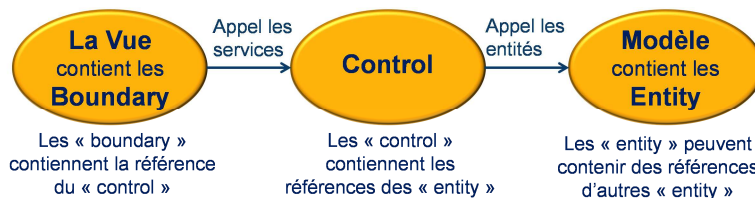


Modélisation MVC

Rappel du contexte MVC :

- la vue propose des actions à effectuer aux utilisateurs. La vue connaissant le contrôleur, utilise les services proposés par celui-ci pour répondre aux actions utilisateurs.
- Le contrôleur, connaissant les entités du modèle, interagit avec elles pour répondre aux services qu'il déclare.

Donc dans le contexte MVC, plus particulièrement dans une implémentation UML des stéréotypes Boundary, Control, Entity on obtient le schéma ci-dessous.



Les entités n'ont pas la référence du control et le control n'a pas la référence de la vue.

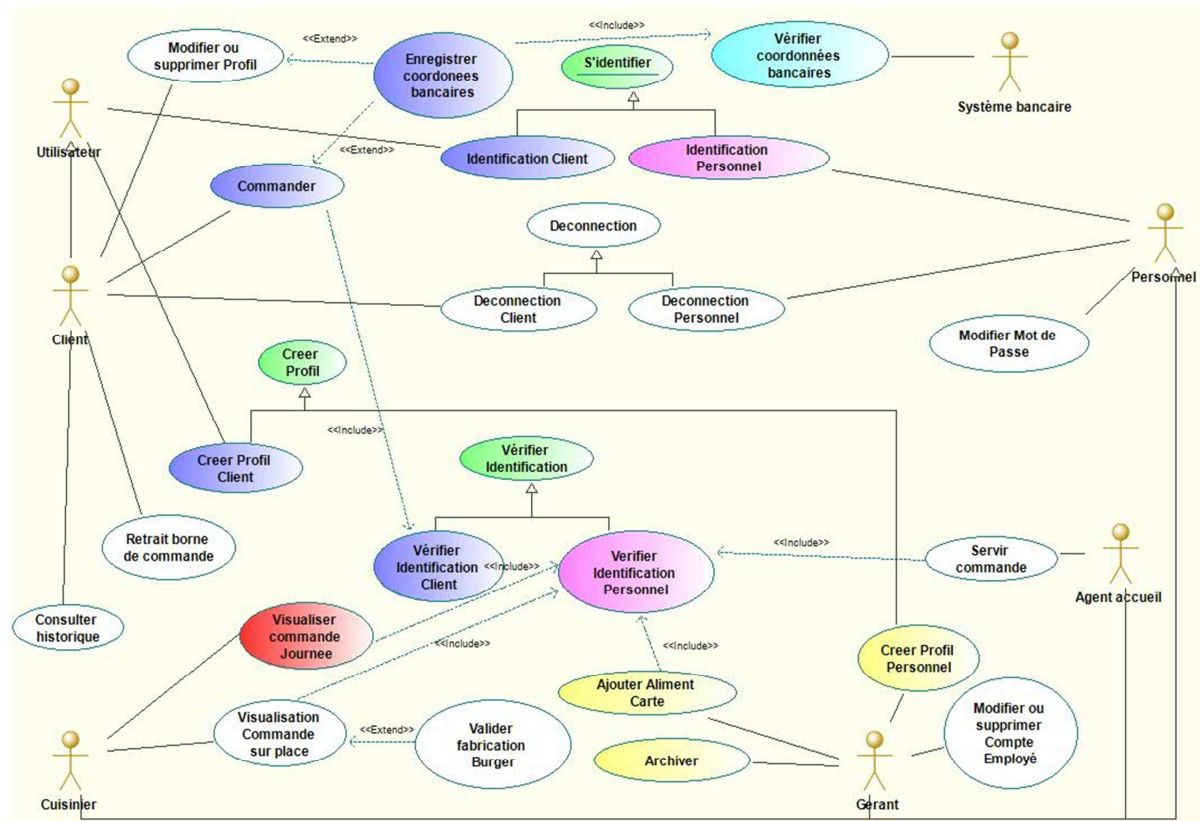
« Les constructeurs des **boundary** prennent en paramètres d'entrée :

- Le contrôleur du cas,
- Les contrôleurs des cas inclus et/ou étendu **SAUF** pour les cas inclus et/ou étendus qui ont une interaction avec l'acteur, à ce moment on passe son boundary.

Les **contrôleurs** ne possèdent pas de constructeur (sauf pour des initialisations métiers portant sur les entités exemple : initialisation des bases de données : BDClient, BDPersonnel et BDCommande et du menu du restaurant : Carte) ».

Démarche

Ci-dessous une vue d'ensemble des cas à implémenter.



Coder les cas dans l'ordre suivant :

- Créer profil,
- S'identifier,
- Vérifier identification,
- Ajouter aliment carte,
- Enregistrer coordonnées bancaires,
- Visualisation commande journée,
- Archiver.

Vérification du fonctionnement du cas

Démarche

Générer la méthode `toString` dans les entités : **Aliment, Carte, Commande, Personnel, Client, carteBancaire, BDCommande, BDClient et BDProfil.**

Pour effectuer la génération de la méthode `toString` faire :

Source > Generate `toString`...

Dans le contrôleur du cas ajouter la méthode `verification` qui permet d'afficher l'état des objets qui ont été modifiés.

Dans le paquetage `testTextuel` télécharger depuis Moodle le test à effectuer.

Exemple

Pour le cas « créer profil gérant » :

Générer la méthode `toString` dans les entités `Client`, `Personnel` et `BDPersonnel`.

```
public class Client extends Profil {
...
    // TODO pour test
    @Override
    public String toString() {
        return "Client [nom=" + getNom() + ", prenom=" + getPrenom()
            + ", login=" + getLogin() + ", mdp=" + getMdp() + ", connecte="
            + connecte + ", carte =" + carteBancaire + "]\n";
    }

    public class Personnel extends Profil {
...
        @Override
        public String toString() {
            return "Personnel [nom=" + getNom() + ", prenom=" + getPrenom()
                + ", login=" + getLogin() + ", mdp=" + getMdp() + ", gerant="
                + gerant + ", connecte=" + connecte + "]\n";
        }
    }

    public class BDPersonnel {
        private Map<Integer, Personnel> listePersonnel = new HashMap<Integer, Personnel>();
...
        // TODO pour les tests
        @Override
        public String toString() {
            return "BDPersonnel [listePersonnel=" + listePersonnel + "]\n";
        }
    }
}
```

Ajouter dans le control la méthode `visualiserBDUtilisateur` qui permet d'afficher l'état des objets `BDPersonnel` et `bdClient`.

```
public class ControlCreerProfil {
...
    // TODO Pour les tests
    public String visualiserBDUtilisateur() {
        return (bdPersonnel + "\n" + bdClient);
    }
}
```

Télécharger depuis Moodle le test à effectuer dans le paquetage TestCas.

```
public class TestCasCreerProfilGerant {

    public static void main(String[] args) {
        // Initialisation des objets metier
        BDClient bdClient = new BDClient();
        BDPersonnel bdPersonnel = new BDPersonnel();

        // Initialisation controleur du cas
        ControlCreerProfil controlCreerProfil = new ControlCreerProfil(
            bdClient, bdPersonnel);
        // Initialisation vue du cas
        BoundaryCreerProfilGerant boundaryCreerProfilGerant = new BoundaryCreerProfilGerant(
            controlCreerProfil);

        // Lancement du cas
        boundaryCreerProfilGerant.creerProfilGerant();

        // Verification de la bonne realisation du cas
        System.out.println("VERIFICATION");
        System.out.println(controlCreerProfil.visualiserBDUtilisateur());
    }
}
```