

Spécification, Analyse et conception Gestion des menus d'un restaurant

01/08/2020



Quan ZHANG
Yajuan LUO

Table des matières

Contents

1	Description	3
2	Requêtes	4
2.1	Mise à jour	5
2.2	Suppression	6
2.3	consultation	6
3	Trigger	9
4	Maquettes d'écran	12
4.1	System de commande	12
4.2	System de gestion	14
5	ATTENTION	16

1 Description

Afin de répondre aux exigences de la base de données, nous devons non seulement développer **un système de gestion des menus de restaurant**, mais également mettre en place **un système de commande**.

Système de commande:

- Le numéro de commande est généré aléatoirement après la saisie du numéro de table.
- Montrer le menu aux clients.
- Ajouter des plats au panier.
- Modifier les informations de commande dans le panier.
- Cliquer sur **Confirmer** pour envoyer la commande au système de gestion.

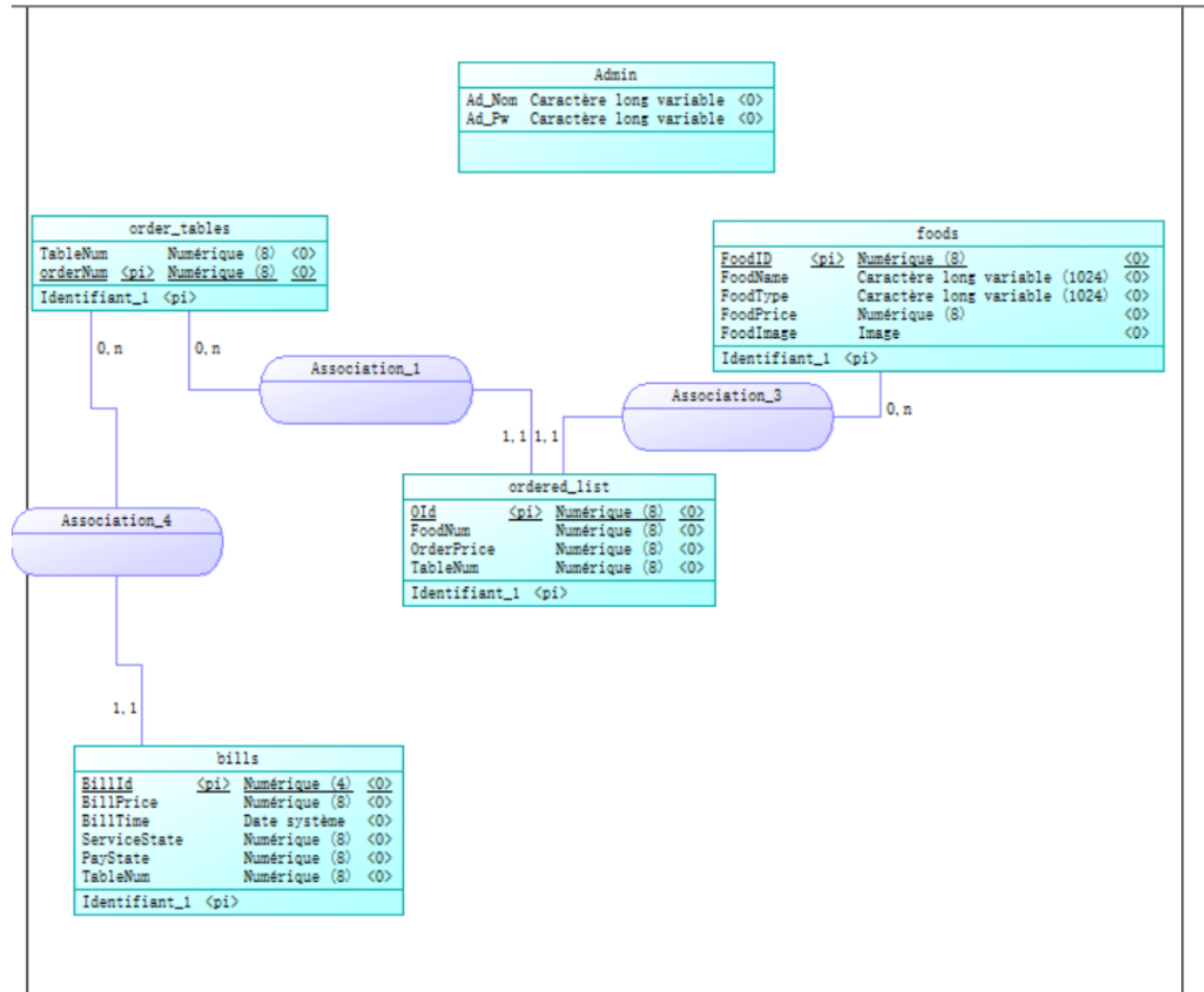
Système de gestion des menus de restaurant

- Ajouter un plat (classification, nom, prix et image)
- Afficher les informations de commande
- Cliquer sur **Terminé** et changer le statut de commande.
- Cliquer sur **Payé** et supprimer cette commande.

Nous avons simulé la situation de restauration de McDonald's. Lors de la commande, saisir le numéro de la marque et générer aléatoirement le numéro de commande. On va enregistrer tous les commandes qui sont dans le panier dans 'ordered_list'. Après confirmation de la commande, on va enregistrer dans 'bills'.

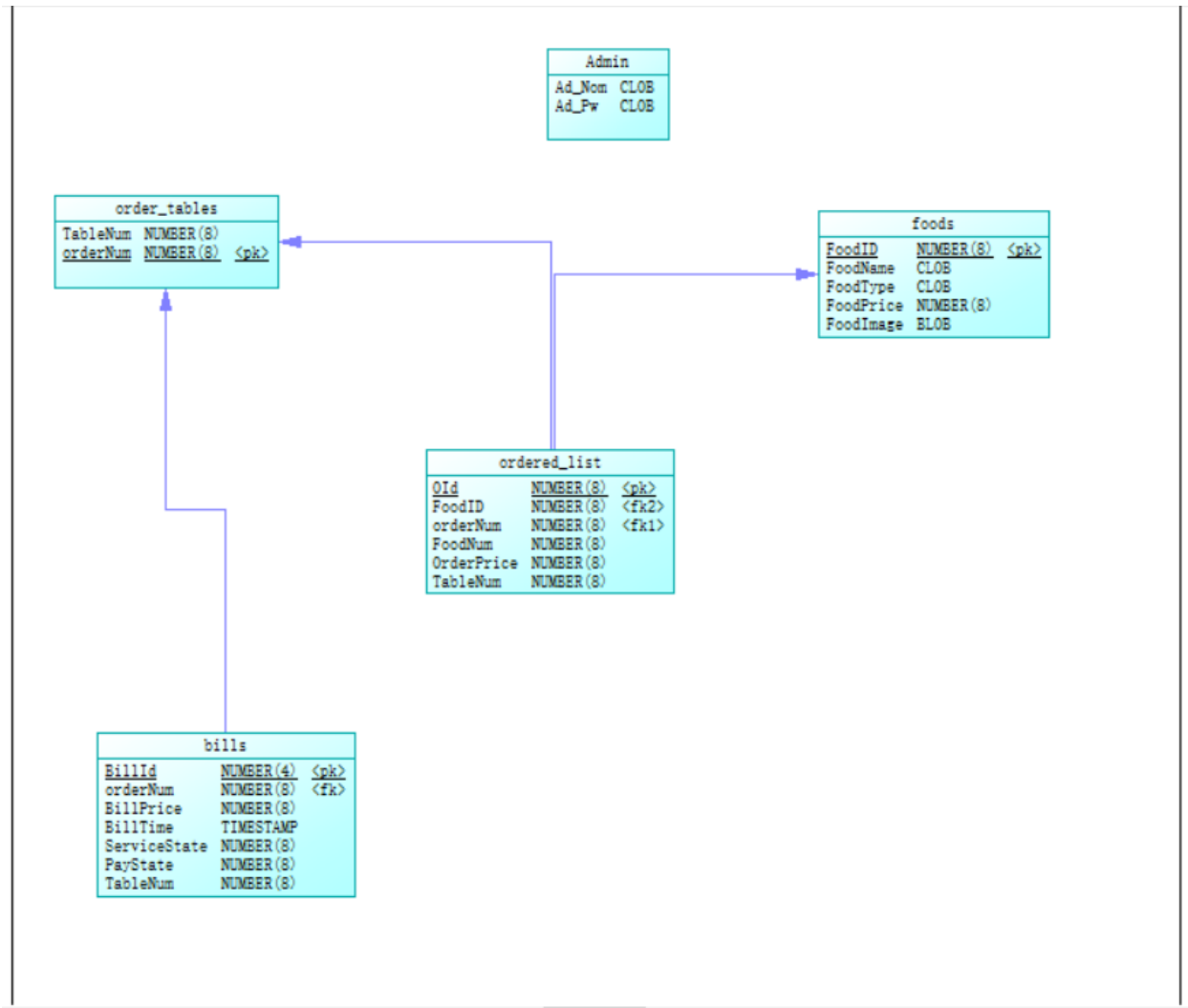
2 Requêtes

Avant de le faire, on va designer un Modèle Entité-Association (version 1) qui est satisfait au moins 3NF. En considérant le cas réel. Bien que les règles ne soient pas respectées, des problèmes de redondance des données surviendront. Une redondance appropriée des données peut souvent améliorer les performances du système. Il convient au projet sous haute concurrence.



Après la conversion et la fusion, on peut obtenir les 5 entité:

- Admin(Ad_Nom, Ad_Pw)
- foods(FoodId, FoodName, FoodType, FoodPrice, FoodImage)
- order_tables(TableNum, OrderNum)
- ordered_list(OId, FoodId, OrderNum, TableNum, FoodNum, OrderPrice)
- bill(BillId, OrderNum,TableNum, BillPrice, BillTime, ServiceState, PayState)



2.1 Mise à jour

- La modification des informations du plat dans tableau 'food'.

```
1 UPDATE "C##ORCL"."foods" SET "FoodPrice" = '4' WHERE "FoodID" = '6';
```

- La modification la quantité d'un plat demandé par le client.

```
1 UPDATE "C##ORCL"."ordered_list" SET "FoodNum" = '2', "OrderPrice" = '8'
2 WHERE "OID" = '1';
```

Afin de créer 2 requêtes impliquant 2 tables, 2 requêtes impliquant plus de 2 tables, on va créer 3 Triggers dans chapitre 3:

- La modification de "FoodID".

```
1 UPDATE "C##ORCL"."foods" SET "FoodID" = '3' WHERE "FoodID" = 1;
```

- La modification de "FoodID" en cherchant le nom de plat.

```
1 UPDATE "C##ORCL"."foods" SET "FoodID" = '1' WHERE "FoodName" = 'glaces';
```

- La modification de "orderNum".

```
1 UPDATE "C##ORCL"."order_tables" SET "orderNum" = '1112' WHERE "orderNum" = '1111';
```

- La modification de "orderNum" en cherchant une table.

```
1 UPDATE "C##ORCL"."order_tables" SET "orderNum" = '1111' WHERE "TableNum" = '111';
```

2.2 Suppression

- La suppression d'un admin.

```
1 DELETE FROM "C##ORCL"."Admin" WHERE "Ad_Nom" = '1010';
```

- La suppression d'un order en cherchant "OID".

```
1 DELETE FROM "C##ORCL"."ordered_list" WHERE "OID" = '1';
```

- La suppression d'un order dans 'ordered_list' en supprimant dans "foods" en cherchant "FoodID".(Pour le faire, grace a **on delete cascade** avec foreign key).

```
1 DELETE FROM "C##ORCL"."foods" WHERE "FoodID" = '4';
```

- La suppression d'un order dans 'ordered_list' en supprimant dans "foods" en cherchant "FoodName".(Pour le faire, grace a **on delete cascade** avec foreign key).

```
1 DELETE FROM "C##ORCL"."foods" WHERE "FoodName" = 'cola';
```

- La suppression d'un bill et un order dans 'ordered_list' et 'bills' en supprimant dans "order_tables" en cherchant "orderNum".(Pour le faire, grace a **on delete cascade** avec foreign key).

```
1 DELETE FROM "C##ORCL"."order_tables" WHERE "orderNum" = '1111';
```

- La suppression d'un bill et un order dans 'ordered_list' et 'bills' en supprimant dans "order_tables" en cherchant "TableNum".(Pour le faire, grace a **on delete cascade** avec foreign key).

```
1 DELETE FROM "C##ORCL"."order_tables" WHERE "TableNum" = '111';
```

2.3 consultation

- Pour connecter au system de gestion, on doit comparer password dans 'admin'.

```
1 select "Ad_Pw" from "C##ORCL"."Admin" WHERE "Ad_Nom"='111';
```

- Pour afficher les infos d'un plat, on doit chercher dans 'food'.

```
1 select * from "C##ORCL"."foods" WHERE "FoodID" = '1';
```

- Chercher tous les commandes correspondant à TableNum.

```
1 select * from "C##ORCL"."bills" WHERE "TableNum" = '111';
```

- Rechercher le montant total des commandes générées par chaque table.(sum(BillPrice) et group by TableNum).

```
1 select sum("BillPrice") from "C##ORCL"."bills" group by "TableNum";
```

- Rechercher des commandes par temps (from bill order by BillTime).

```
1 select * from "C##ORCL"."bills" order by "BillTime";
```

- Jointure interne entre 'foods' et 'ordered_list' nous permet d'afficher le plat le plus vendu.

```
1 select "FoodName" from "ordered_list" inner join "foods" on
2 "ordered_list"."FoodID" = "foods"."FoodID"
3 WHERE "FoodNum" in
4 (
5 select max("FoodNum") from "ordered_list"
6 );
```

- Jointure interne entre 'foods' et 'ordered_list' nous permet d'afficher le type de plat le plus vendu.

```
1 select "FoodType" from "ordered_list" inner join "foods" on
2 "ordered_list"."FoodID" = "foods"."FoodID"
3 WHERE "FoodNum" in
4 (
5 select max("FoodNum") from "ordered_list"
6 );
```

- Jointure interne entre 'foods' et 'ordered_list' nous permet d'afficher l'image de plat le plus vendu.

```
1 select "FoodImage" from "ordered_list" inner join "foods" on
2 "ordered_list"."FoodID" = "foods"."FoodID"
3 WHERE "FoodNum" in
4 (
5 select max("FoodNum") from "ordered_list"
6 );
```

- Jointure interne entre 'bill' et 'ordered_list' nous permet d'afficher ID de plats en cours de préparer.

```
1 select "FoodID" from "ordered_list" inner join "bills" on
2 "ordered_list"."orderNum" = "bills"."orderNum" WHERE "ServiceState"='0';
```

- Jointure externe entre 'bill' et 'ordered_list' nous permet d'afficher les orders qui n'ont pas encore confirmé (group by OrderNum).

```
1 select count("ordered_list"."TableNum") from
2 "ordered_list" left outer join "bills"
3 on "ordered_list"."orderNum" = "bills"."orderNum"
4 WHERE "PayState" is null
5 group by "ordered_list"."orderNum";
```

- Jointure interne permis 'foods', 'bill' et 'ordered_list' nous permet d'afficher nom de plats en cours de préparer.

```
1 select "FoodName" from "ordered_list"
2 inner join "foods" on "ordered_list"."FoodID" = "foods"."FoodID"
3 inner join "bills" on "ordered_list"."orderNum" = "bills"."orderNum"
4 where "ServiceState" = '0';
```

- Jointure interne permis 'foods', 'bill' et 'ordered_list' nous permet d'afficher type de plats en cours de préparer.

```
1 select "FoodType" from "ordered_list"
2 inner join "foods" on "ordered_list"."FoodID" = "foods"."FoodID"
3 inner join "bills" on "ordered_list"."orderNum" = "bills"."orderNum"
4 where "ServiceState" = '0';
```

- Jointure interne permis 'foods', 'bill' et 'ordered_list' nous permet d'afficher Image de plats en cours de préparer.

```
1 select "FoodImage" from "ordered_list"
2 inner join "foods" on "ordered_list"."FoodID" = "foods"."FoodID"
3 inner join "bills" on "ordered_list"."orderNum" = "bills"."orderNum"
4 where "ServiceState" = '0';
```

- Jointure interne permis 'foods', 'bill' et 'ordered_list' nous permet d'afficher noms de plats qui n'ont pas encore payé.

```
1 select "FoodName" from "ordered_list"
2 inner join "foods" on "ordered_list"."FoodID" = "foods"."FoodID"
3 inner join "bills" on "ordered_list"."orderNum" = "bills"."orderNum"
4 where "PayState" = '0';
```

- Jointure externe permis 'foods', 'bill' et 'ordered_list' nous permet d'afficher le revenu total.

```
1 select sum("bills"."BillPrice") from "ordered_list"
2 left outer join "bills" on "ordered_list"."orderNum" = "bills"."orderNum"
3 left outer join "foods" on "ordered_list"."FoodID" = "foods"."FoodID"
4 group by "foods"."FoodID";
```


3 Trigger

Ajouter un **DELETE CASCADE DEFERRABLE** pour Mise à jour dans plusieurs tables:

```
1  /*
2  1
3  */
4  ALTER
5  TABLE "ordered_list" DROP
6  CONSTRAINT FK_ORDERED__ASSOCIATI_FOODS;
7
8  ALTER
9  TABLE "ordered_list" ADD
10 CONSTRAINT FK_ORDERED__ASSOCIATI_FOODS FOREIGN
11 KEY ("FoodID")
12 references "foods" ("FoodID") ON
13 DELETE CASCADE DEFERRABLE;
14
15 CREATE
16 OR REPLACE TRIGGER TGR_TB_ORDERED_UPDATE
17 AFTER
18 UPDATE OF "FoodID" ON
19 "foods"
20 FOR
21 EACH ROW
22 BEGIN
23 IF :OLD."FoodID"<>:NEW."FoodID" THEN
24 UPDATE
25 "ordered_list" SET
26 "FoodID"=:NEW."FoodID" WHERE
27 "FoodID"=:OLD."FoodID";
28 END
29 IF;
30 END;
31
32 /*
33 2
34 */
35 ALTER
36 TABLE "bills" DROP
37 CONSTRAINT FK_BILLS_ASSOCIATI_ORDER_TA;
38
39 ALTER
40 TABLE "bills" ADD
41 CONSTRAINT FK_BILLS_ASSOCIATI_ORDER_TA foreign key ("orderNum")
42 references "order_tables" ("orderNum") ON
43 DELETE CASCADE DEFERRABLE;
44
45 CREATE
46 OR REPLACE TRIGGER TGR_TB_BILLS_UPDATE
```

```

47 AFTER
48 UPDATE OF "orderNum" ON
49 "order_tables"
50 FOR
51 EACH ROW
52 BEGIN
53 IF :OLD."orderNum"<>:NEW."orderNum" THEN
54 UPDATE
55 "bills" SET
56 "orderNum"=:NEW."orderNum" WHERE
57 "orderNum"=:OLD."orderNum";
58 END
59 IF;
60 END;
61
62
63
64 /*
65 3
66 */
67 ALTER
68 TABLE "ordered_list" DROP
69 CONSTRAINT FK_ORDERED__ASSOCIATI_ORDER_TA;
70
71 ALTER
72 TABLE "ordered_list" ADD
73 CONSTRAINT FK_ORDERED__ASSOCIATI_ORDER_TA foreign key ("orderNum")
74 references "order_tables" ("orderNum") ON
75 DELETE CASCADE DEFERRABLE;
76
77 CREATE
78 OR REPLACE TRIGGER TGR_TB_ORDEREDT_UPDATE
79 AFTER
80 UPDATE OF "orderNum" ON
81 "order_tables"
82 FOR
83 EACH ROW
84 BEGIN
85 IF :OLD."orderNum"<>:NEW."orderNum" THEN
86 UPDATE
87 "ordered_list" SET
88 "orderNum"=:NEW."orderNum" WHERE
89 "orderNum"=:OLD."orderNum";
90 END
91 IF;
92 END;

```

Afin de réaliser la fonction **auto_increment** qui existe dans Mysql, on va utiliser triggers pour 'Old', 'FoodId' et 'BillId'.

D'abord, on crée une sequence:

```
1 CREATE SEQUENCE oid_Sequence
2 INCREMENT BY 1
3 START WITH 1
4 NOMAXVALUE
5 NOCYCLE ;
```

Créer un trigger:

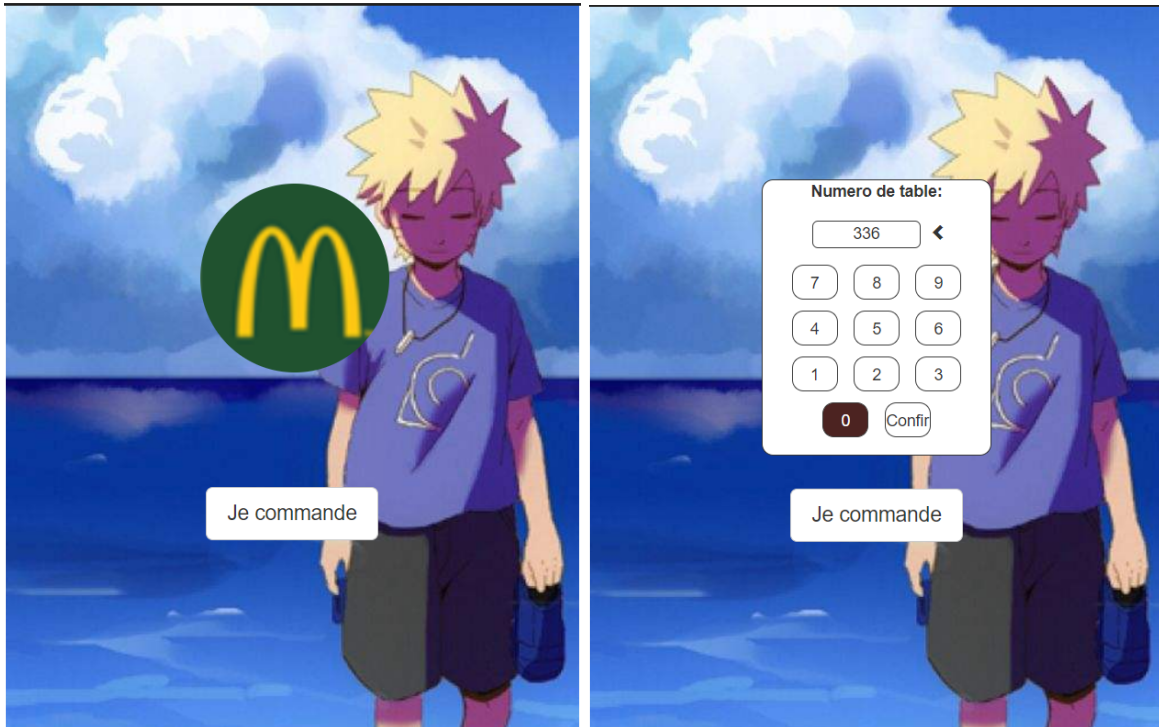
```
1 CREATE TRIGGER oid_increase BEFORE
2 insert ON "ordered_list" FOR EACH ROW
3 begin
4 select oid_SEQUENCE.nextval into:New."OId" from dual;
5 end;
```

Pour 'BillId' et 'FoodId', c'est pareil.

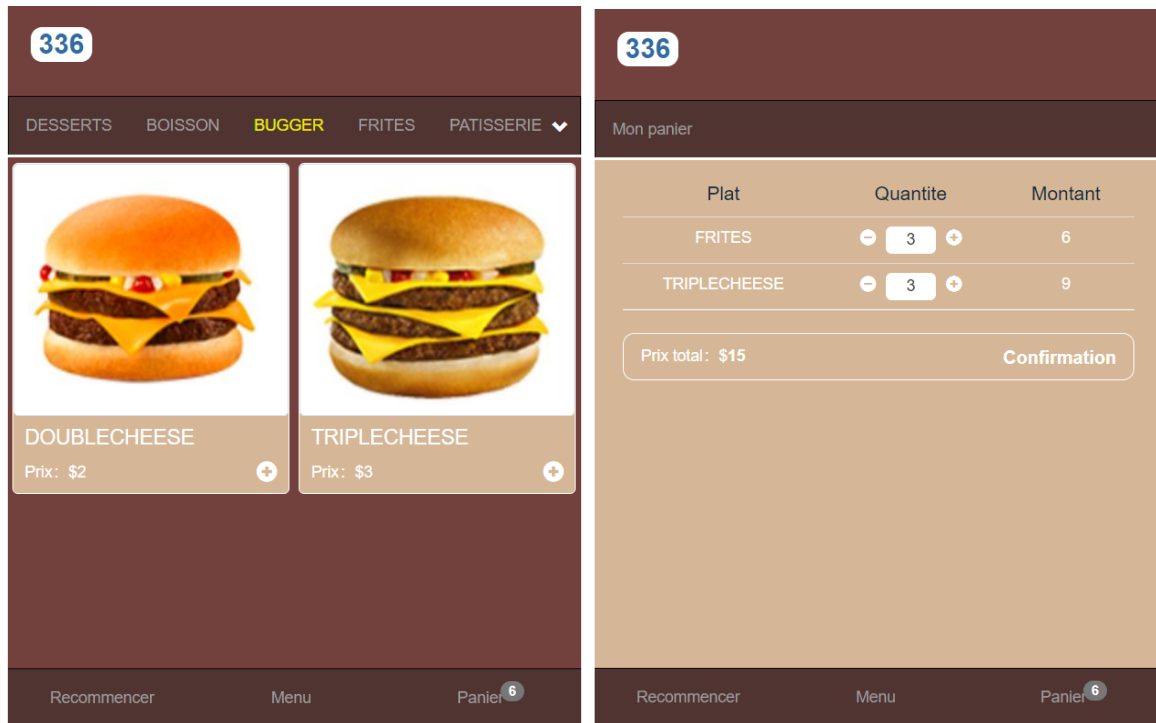
4 Maquettes d'écran

4.1 System de commande

Cliquer sur 'Je commande' puis saisir un numéro de chevalet (Comme chez Mcdonald's):



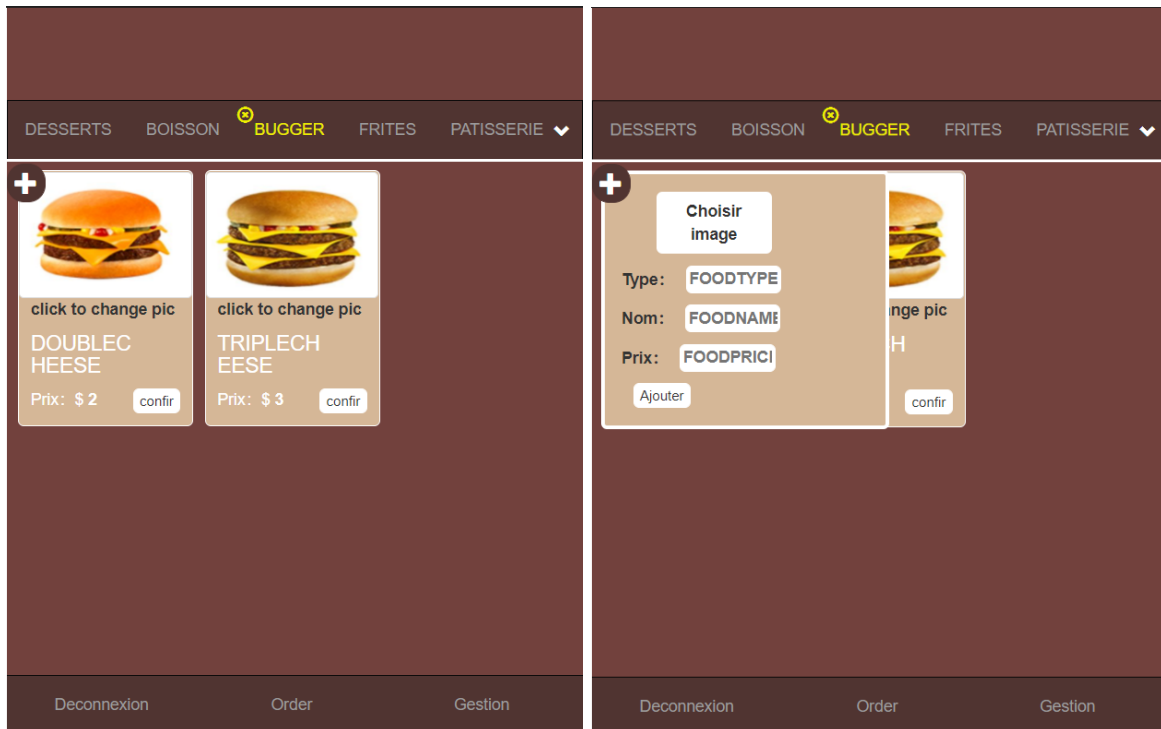
En suite, on peut choisir n'importe quel type de plat ou bien plat qu'on veut. Cliquer sur panier, on va voir tous les infos de nos commandes.



Lorsqu'on clique sur 'confirmation', il va saisir dans table 'BILLS':

	BILLID	TABLENUM	ORDERNUM	BILLPRICE	BILLTIME	SERVICESTATE	PAYSTATE
1	21	336	6384	17	2020-08-11 10:22:41	0	0
2	22	336	6384	17	2020-08-12 12:23:12	0	0
3	20	9966	8381	10	2012-03-22 23:00:00	1	1
4	18	9966	8381	10	2012-03-22 23:00:00	1	1
5	19	336	6384	17	1597183811	0	0

Cliquer sur 'Gestion', on peut saisir un nouveau produit, modifier les infos d'un produit ou bien supprimer un type de plat (les plats correspondants vont etre supprimés).



5 ATTENTION

POUR LES NOMS DE TABLE ET LES NOMS D'ATTRIBUTS, ON DOIT UTILISER TOUS LES LETTRES MAJUSCULES.

EN PLUS, ON VA SUPPRIMER TOUS LES "" QUI SONT AVEC LES TABLES ET LES NOMS D'ATTRIBUTS!!!!!!!!!!!!!!

Pour faciliter de saisir BILLTIME, on change la type:

```
DELETE FROM BILLS WHERE BILLID = '1';
DELETE FROM BILLS WHERE BILLID = '2';
DELETE FROM BILLS WHERE BILLID = '3';
DELETE FROM BILLS WHERE BILLID = '4';
DELETE FROM BILLS WHERE BILLID = '5';
ALTER TABLE BILLS modify (BILLTIME VARCHAR2(30));
INSERT INTO BILLS (TABLENUM, ORDERNUM, BILLPRICE, BILLTIME, SERVICESTATE, PAYSTATE) VALUES ('111', '1111', '4', '2012-03-22 23:00:00', '1', '1');
INSERT INTO BILLS (TABLENUM, ORDERNUM, BILLPRICE, BILLTIME, SERVICESTATE, PAYSTATE) VALUES ('222', '2222', '4', '2012-03-22 23:00:00', '1', '1');
INSERT INTO BILLS (TABLENUM, ORDERNUM, BILLPRICE, BILLTIME, SERVICESTATE, PAYSTATE) VALUES ('333', '3333', '4', '2012-03-22 23:00:00', '1', '1');
INSERT INTO BILLS (TABLENUM, ORDERNUM, BILLPRICE, BILLTIME, SERVICESTATE, PAYSTATE) VALUES ('444', '4444', '6', '2012-03-22 23:00:00', '1', '1');
INSERT INTO BILLS (TABLENUM, ORDERNUM, BILLPRICE, BILLTIME, SERVICESTATE, PAYSTATE) VALUES ('555', '5555', '10', '2012-03-22 23:00:00', '1', '1');
```