Anthony REICHEN & Alex DA SILVA

IUT Lyon 1 – Dut informatique année spéciale | 2021 - 2022

Projet tutoré ASPE

Gestion de site e-commerce

# Méthodologie de gestion de projet :

Méthode retenue : Agile avec usage de Kanban.

Outils utilisés : GanttProject & Trello.

## Choix de la méthode Agile

Pour ce projet tuteuré, nous avons opté pour une méthode agile. Ce choix nous a été dicté par des impératifs de flexibilité, nécessaire pour pallier aux fortes incertitudes entourant le projet.

En effet, le projet nécessite des compétences techniques que nous n’avons pas encore acquises, et pour lesquelles l’IUT ne nous a pas fourni de cours correspondant. Par conséquent, une grande partie du temps alloué au PTUT a dû être dévolu à se documenter sur ces sujets (PHP, IHM, JDBC, MySQL, Modèle MVC, etc.).

## Organisation des cycles de développement

Nous avons choisi de prendre en compte les points forts de chaque membre du binôme pour décider de la personne qui sera responsable du développement de chaque fonctionnalité.

Les étapes de conception (cycle 1 à 3), ont été réalisées entièrement en binôme. Nous avons débuté par l’élaboration du diagramme des cas d’utilisation à partir de l’analyse du cahier des charges fournis. A l’aide du DCU, nous avons découpé le projet en fonctionnalités à inscrire sur le tableau Kanban, qui seront ensuite librement répartis entre les membres du binôme.

A partir du cycle 4, nous avons appliqué la méthode Kanban. Chaque membre du binôme peut choisir une fonctionnalité à implémenter. Des échanges ont lieu au début et à la fin de chaque cycle entre les deux membres du binôme.

### Cycle 1 : Modélisation des cas d’utilisation

La première étape de conception a été l’élaboration du Diagramme de Cas d’Utilisation (DCU), à partir du cahier des charges fournies, en UML. Nous avons disjoint les applications desktop et web car elles n’ont pas les mêmes finalités. En effet, l’application desktop est destiné à l’administrateur du site, alors que le site web est destiné aux clients.

Ressources : Alex, Anthony (responsable)

### Cycle 2 : Modélisation de la base de données

Dans un second temps, nous avons déduit du cahier des charges et des DCU les données nécessaires aux applications. La base de données est commune aux applications web et desktop. Le schéma de données correspond au modèle « entité/association », et a été réalisé sous StarUML.

Ressources : Alex, Anthony (responsable)

### Cycle 3 : Modélisation des classes

La modélisation des classes a été réalisée sous StarUML. Les classes métiers sont les même pour les deux applications. Certaines méthodes ne seront probablement pas implémentées en PHP.

Ressources : Alex (responsable), Anthony

### Cycle 4 : Modélisation des IHM

Les maquettes ont été réalisées sous Pencil.

#### Sous-cycle 4a : IHM desktop

Les interfaces de l’application java sont inspirées de celles présente dans le cours de Java avancé du second semestre.

Ressources : Anthony

#### Sous-cycle 4b : IHM Web

Les interfaces du site web sont inspirée du design général du site Amazon.

Ressources : Anthony

### Cycle 5 : Création de la base de données

Ressources : Alex

### Cycle 6 : Application JAVA

#### Sous-cycle 6a : Interface Homme-Machine

L’implémentation des IHM de l’application desktop a été réalisée avec la bibliothèque Swing de JAVA. Ce choix a été très fortement suggéré par l’IUT, puisque le cours correspondant est basé exclusivement sur cette bibliothèque. De plus, ce cours faisant l’objet d’une évaluation, y compris sur les travaux pratiques ayant un sujet précis, la conception et l’implémentation de l’IHM ne pouvaient pas être libres.

En effet, le cours d’IHM Java devait être destiné à implémenter l’IHM du projet tutoré, alors même qu’un rendu **individuel** de TP, dont le sujet est imposé, est exigé. Le projet tutoré étant mené en binôme, il y a là un paradoxe évident qui conduit non pas à mutualiser le temps consacré au PTUT entre deux matières, mais à multiplier par deux le travail exigé. Face à cette situation, nous avons donc décidé d’implémenter strictement le sujet de TP pour l’IHM de l’application desktop, au détriment de notre vision de départ.

Ressources : Anthony

#### Sous-cycle 6b : Fonctionnalités métiers

(Réalisation en cours)

Ressources : Alex, Anthony

#### Sous-cycle 6c : Connexion à la base de données

Un temps non négligeable a été alloué à la collecte d’information et à l’apprentissage de cette fonctionnalité. Ce sujet n’a jamais été vu en cours et ne le sera apparemment pas faute de temps. Toutefois, il est indispensable pour la réalisation de l’application.

(Réalisation en cours)

Ressources : Alex

### Cycle 7 : Application PHP

#### Sous-cycle 7a : Interface Homme-Machine

Nous avons choisi d’utiliser le framework Bootstrap (Réalisation en cours).

Ressources : Anthony

#### Sous-cycle 7b : Fonctionnalités métiers

Nous avons opté pour l’utilisation du frameword Symfony (Réalisation en cours)

Ressources : Alex, Anthony

#### Sous-cycle 7c : Connexion à la base de données

(Réalisation en cours)

Ressources : Alex

# Annexes :

## Méthodologie de gestion de projet

### Diagramme de Gantt

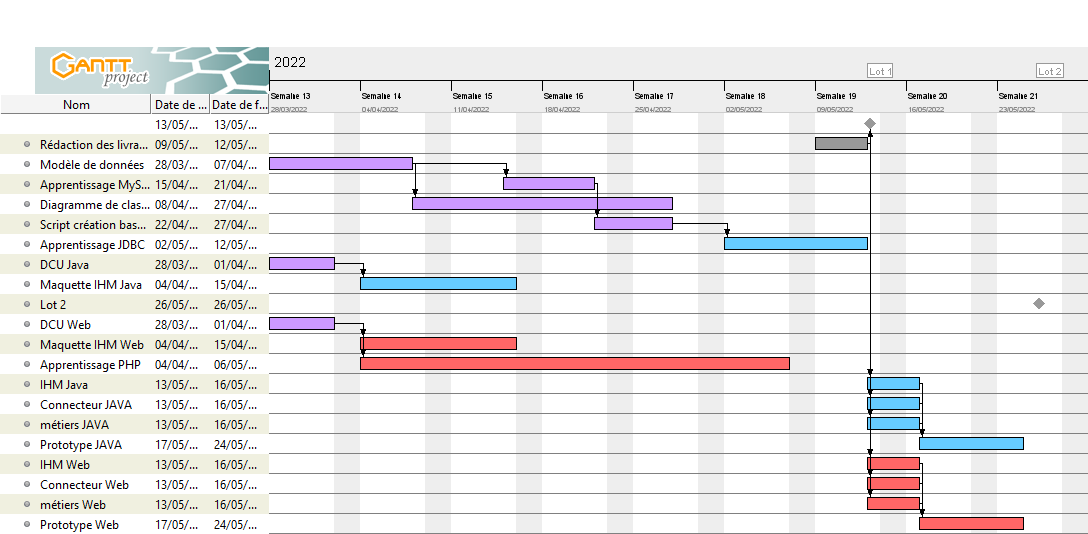


Figure : 1 Diagramme de Gantt

### Diagramme de ressources

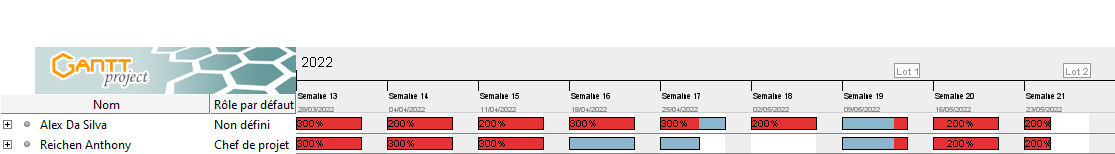


Figure : 2 Diagramme de ressources

## Kanban

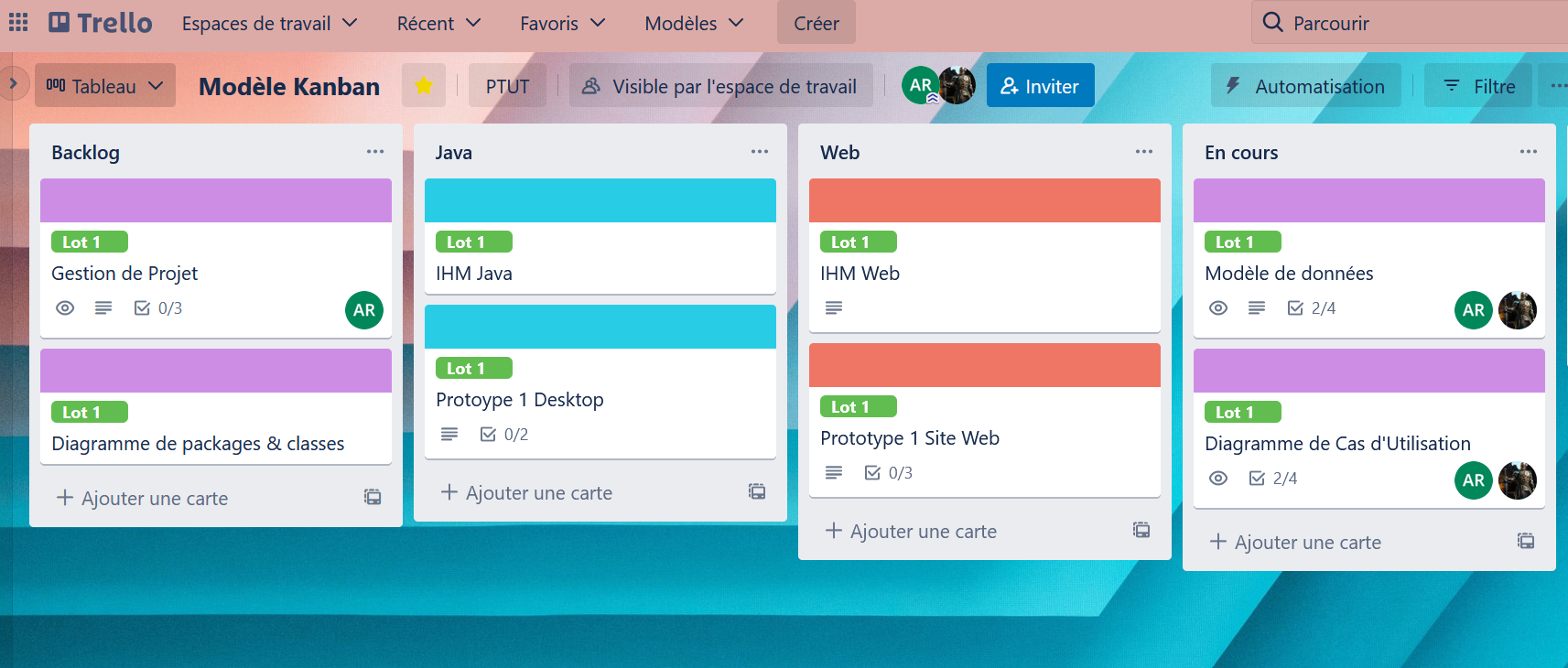


Figure 3 : Capture d'écran du Kanban (site Trello)

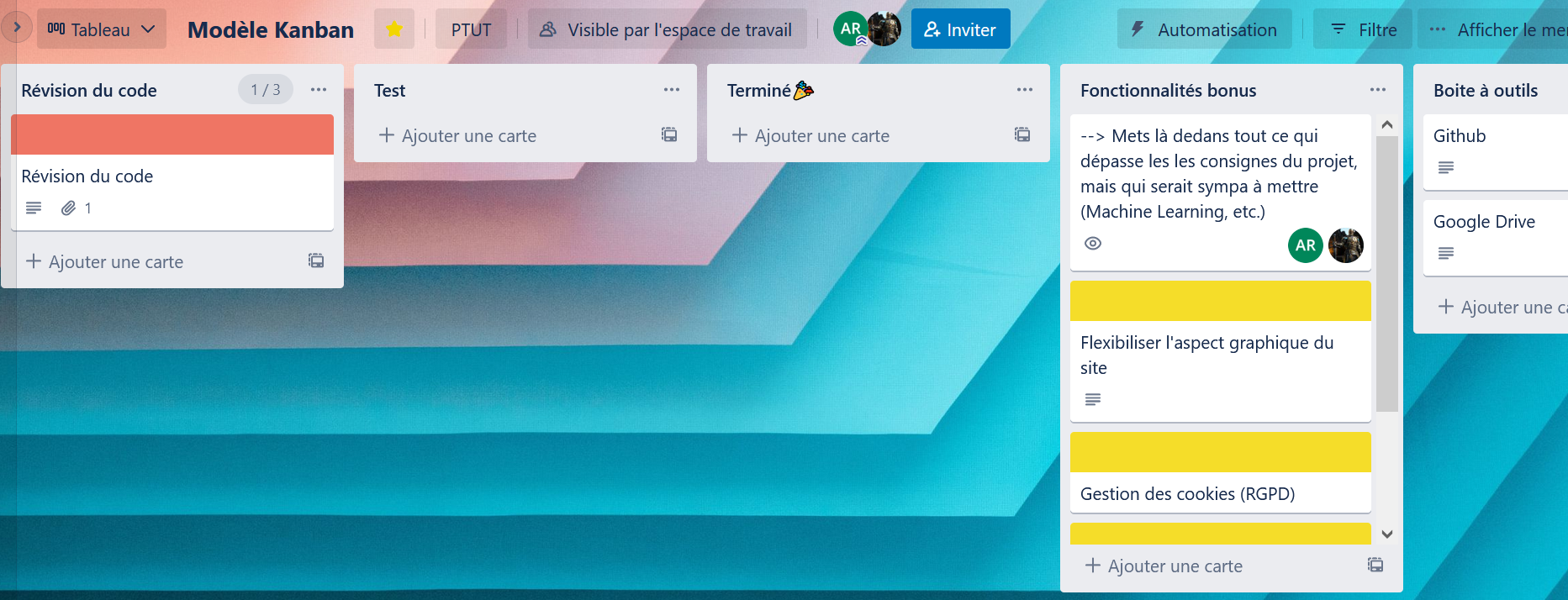


Figure 4 : Capture d'écran du Kanban (site Trello)

## Maquettes d’interfaces homme-machine

### Maquettes de l’application Java



Figure 5 : menu de l'application

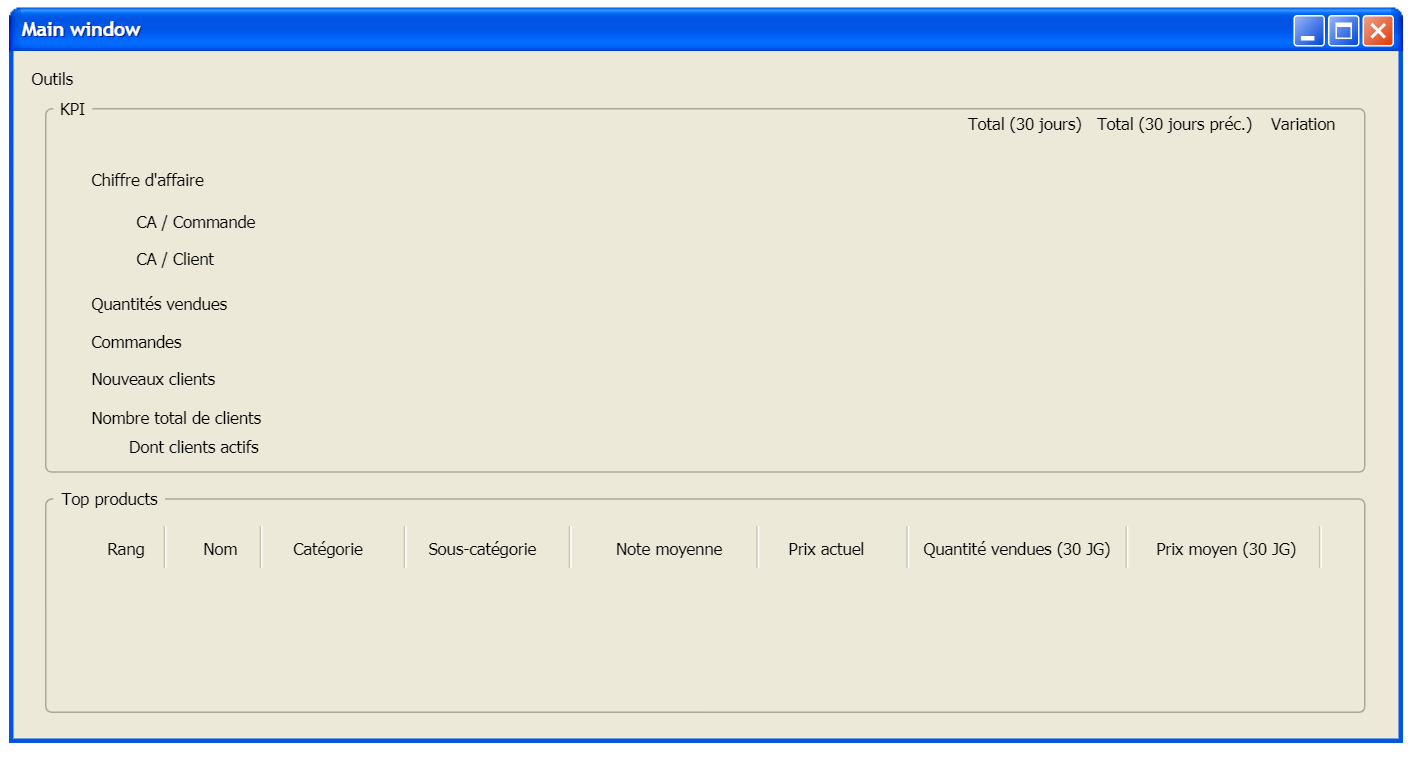


Figure 6 : page d'accueil de l'application

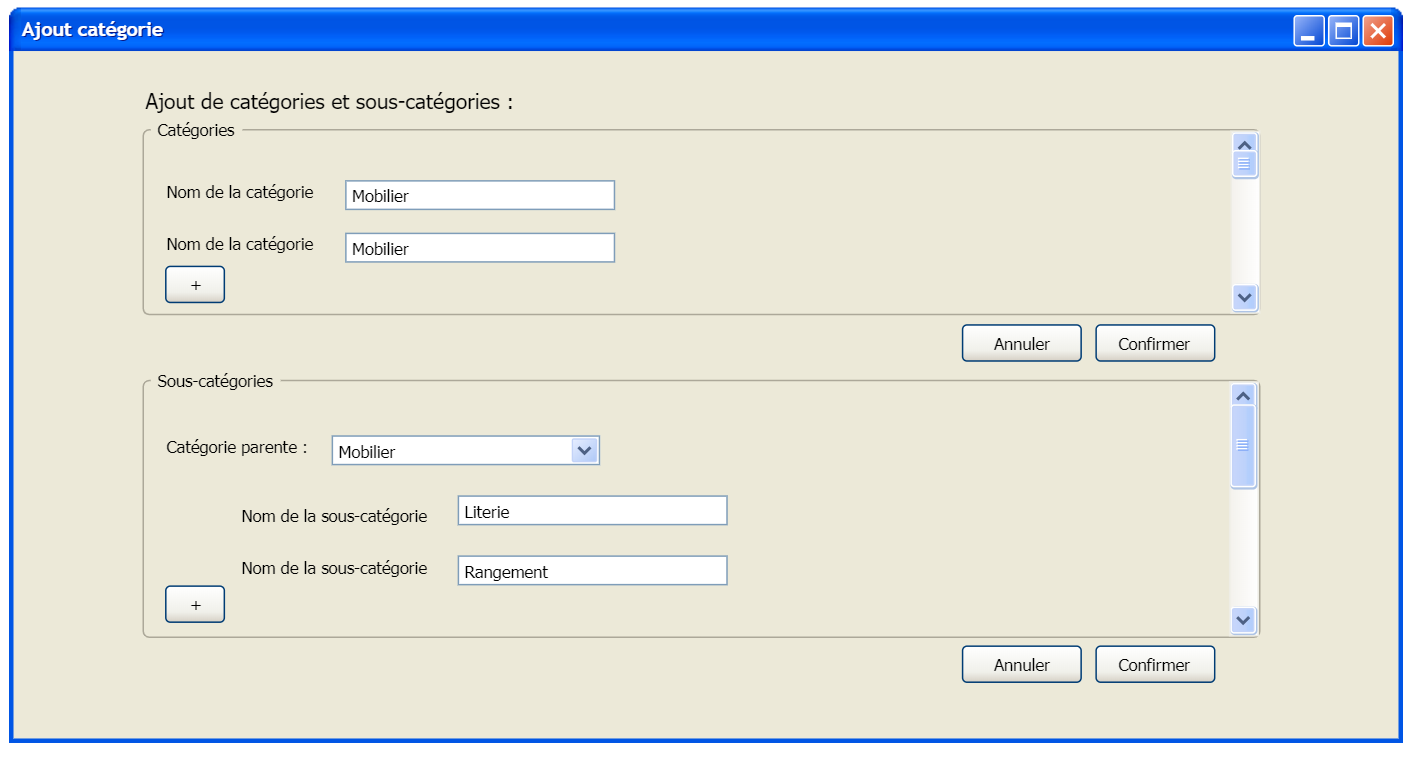


Figure 7 : ajout catégorie

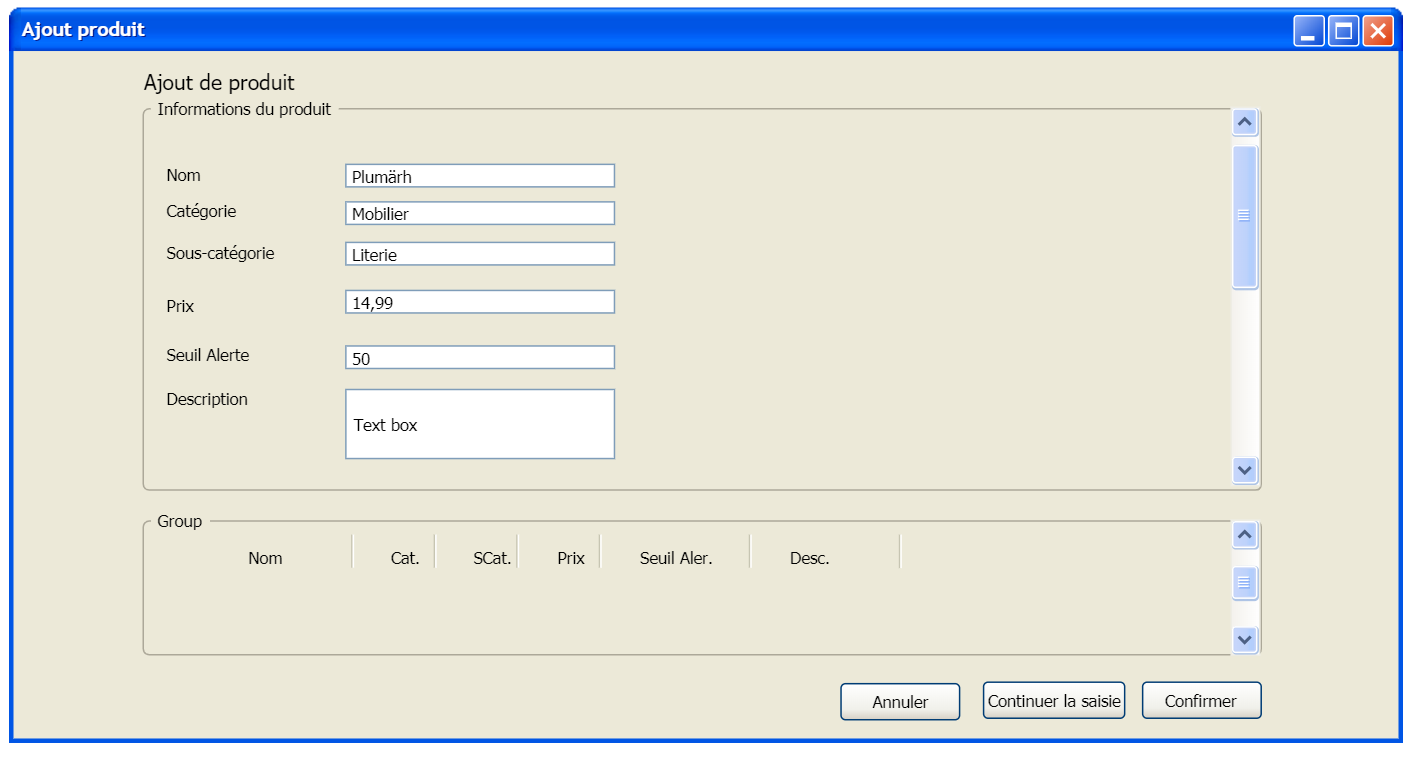


Figure 8 : ajout produit

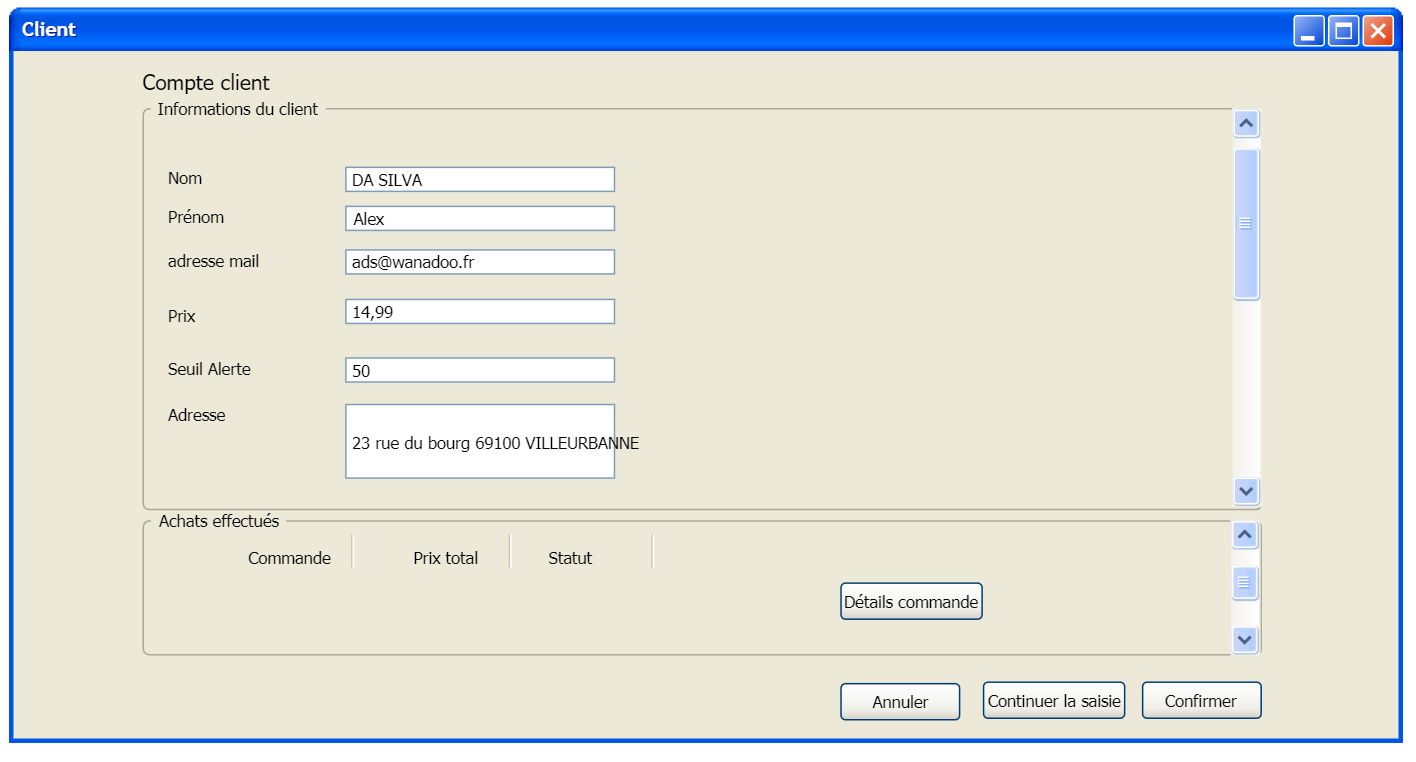


Figure 9 : gestion client

### Maquettes du site web

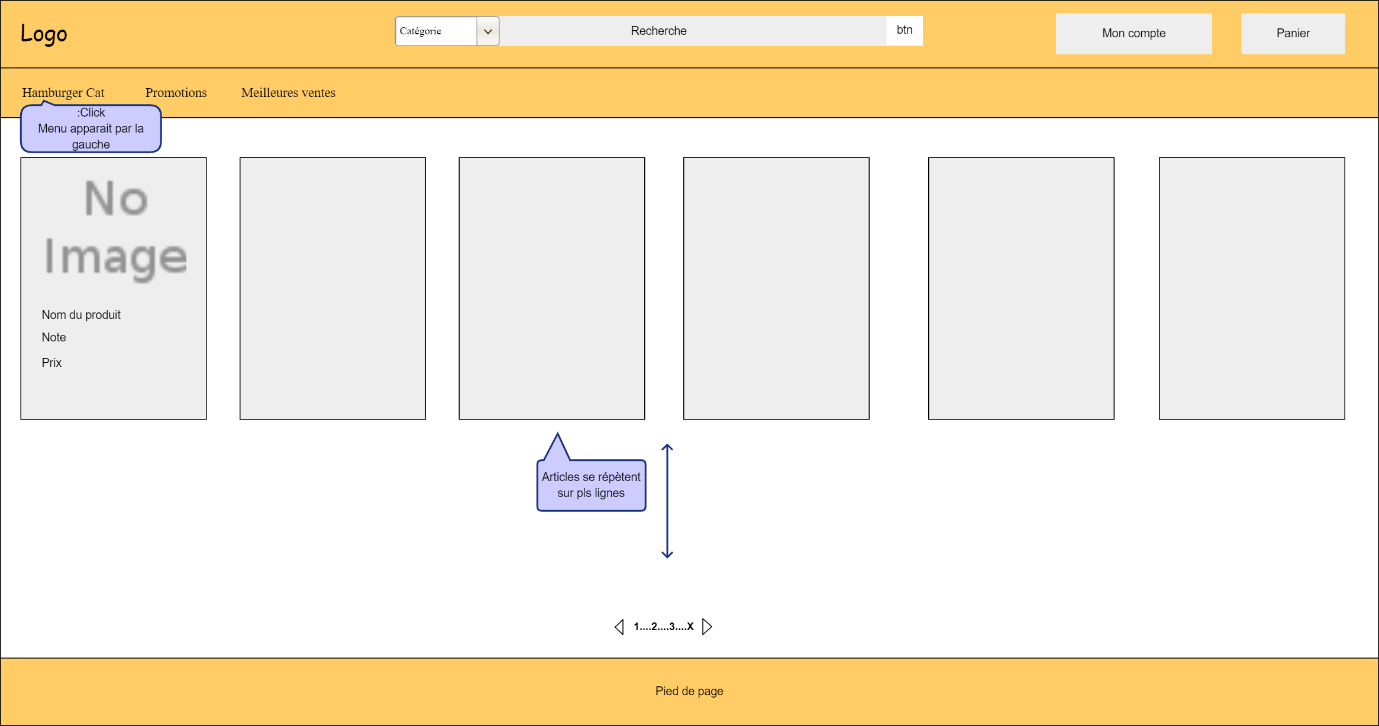


Figure 10 : Page d'accueil

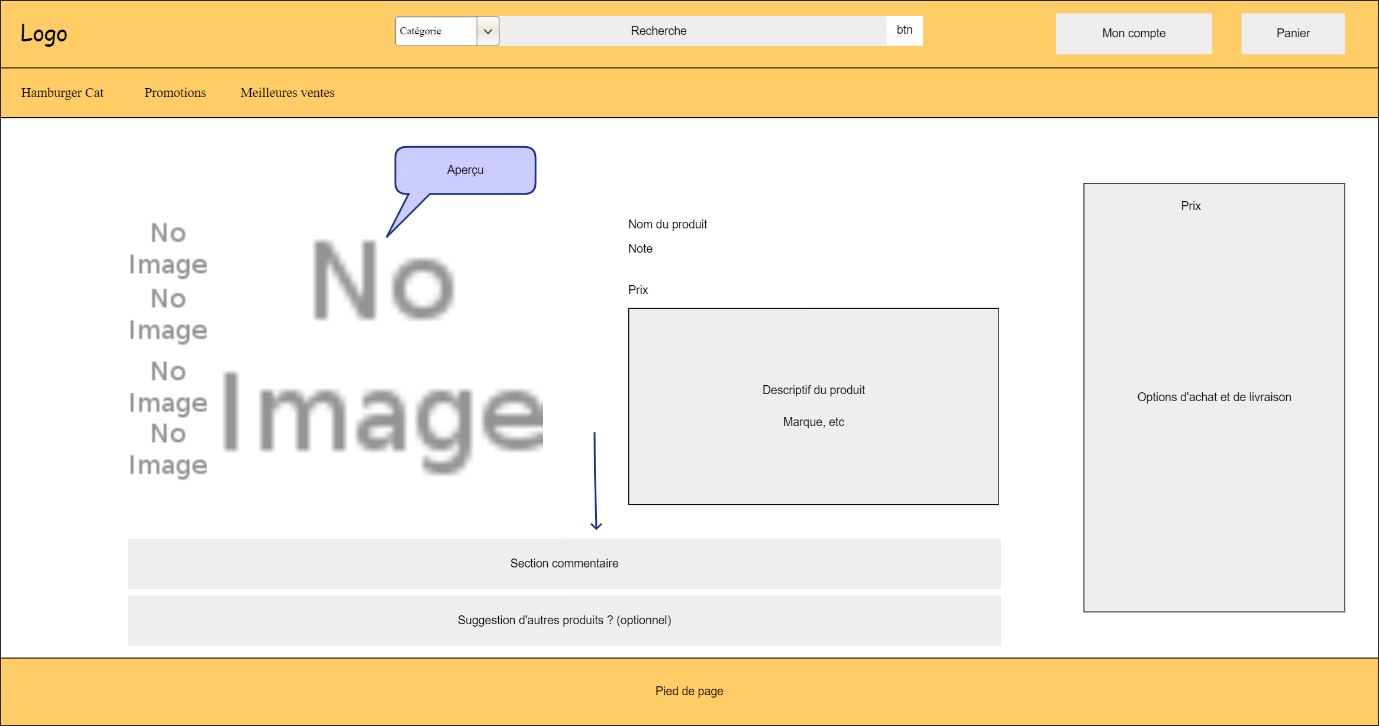


Figure 11 : Détails du produit

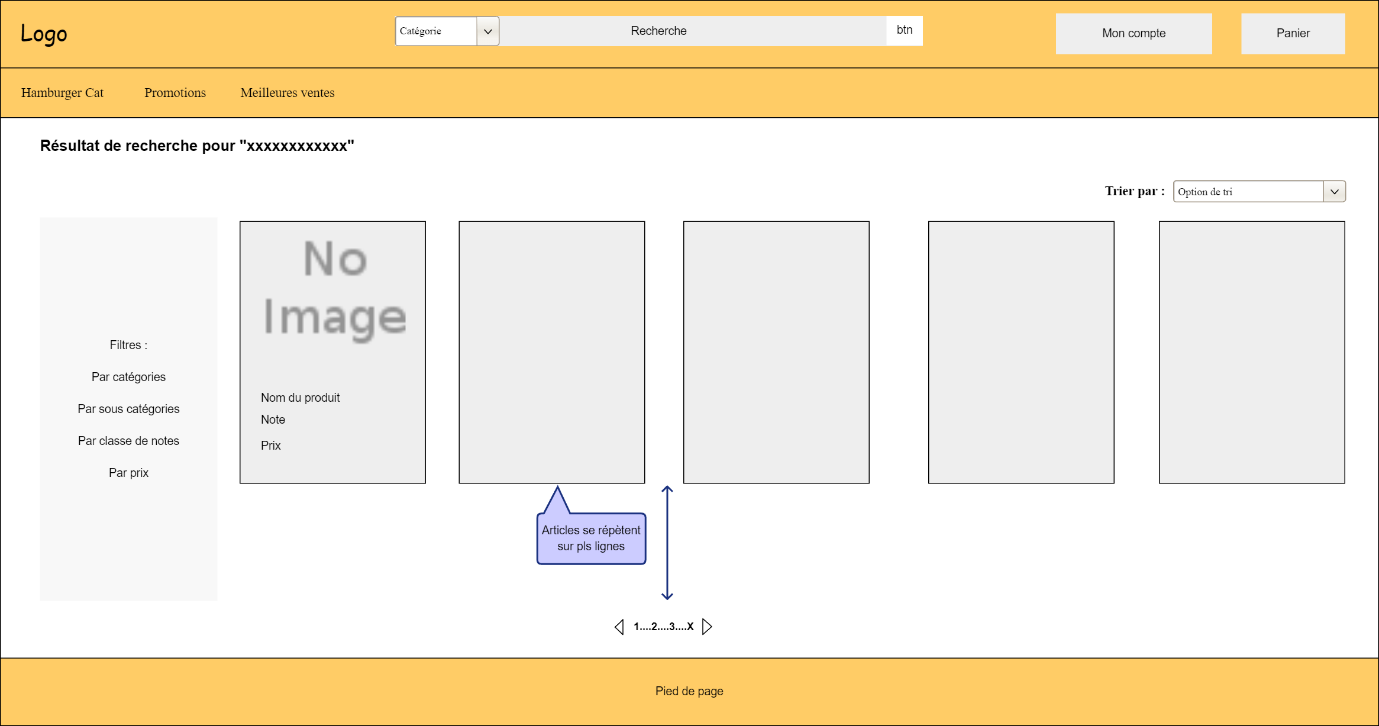


Figure 12 : Résultat de recherche

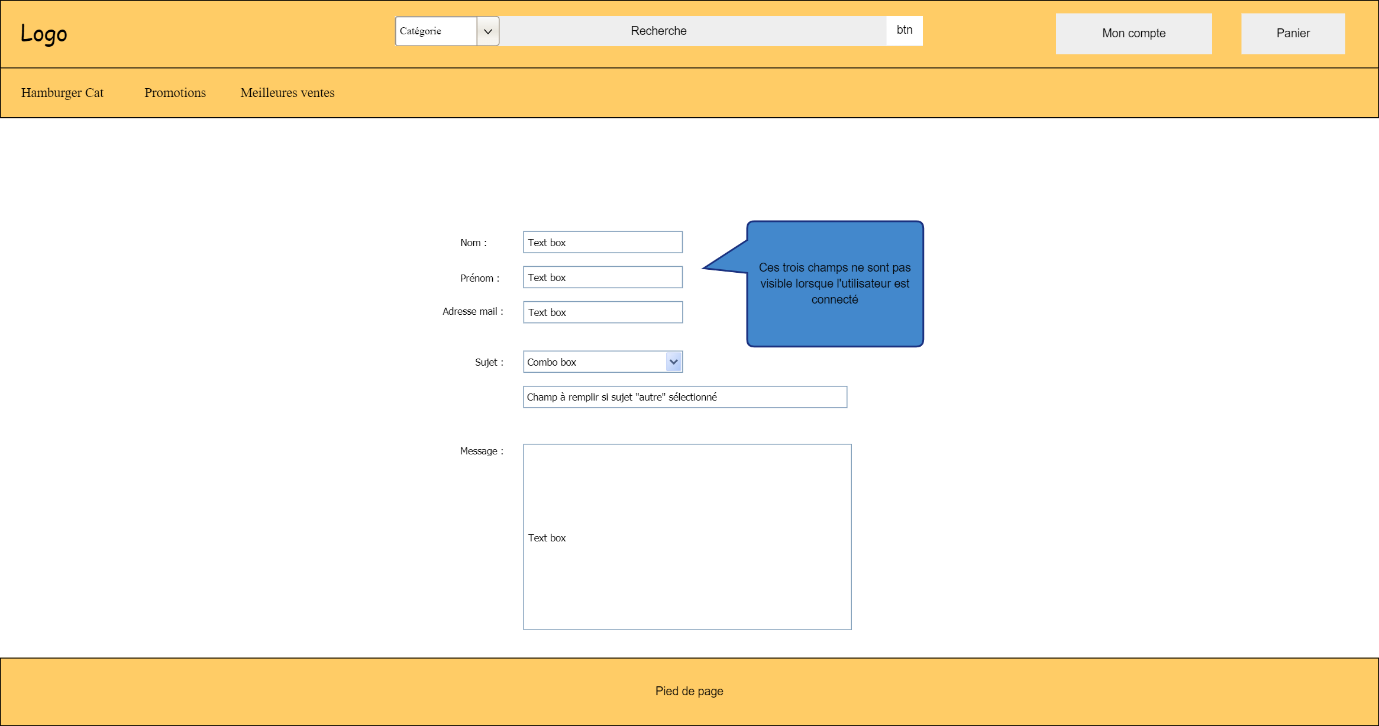


Figure 13 : Formulaire de contact

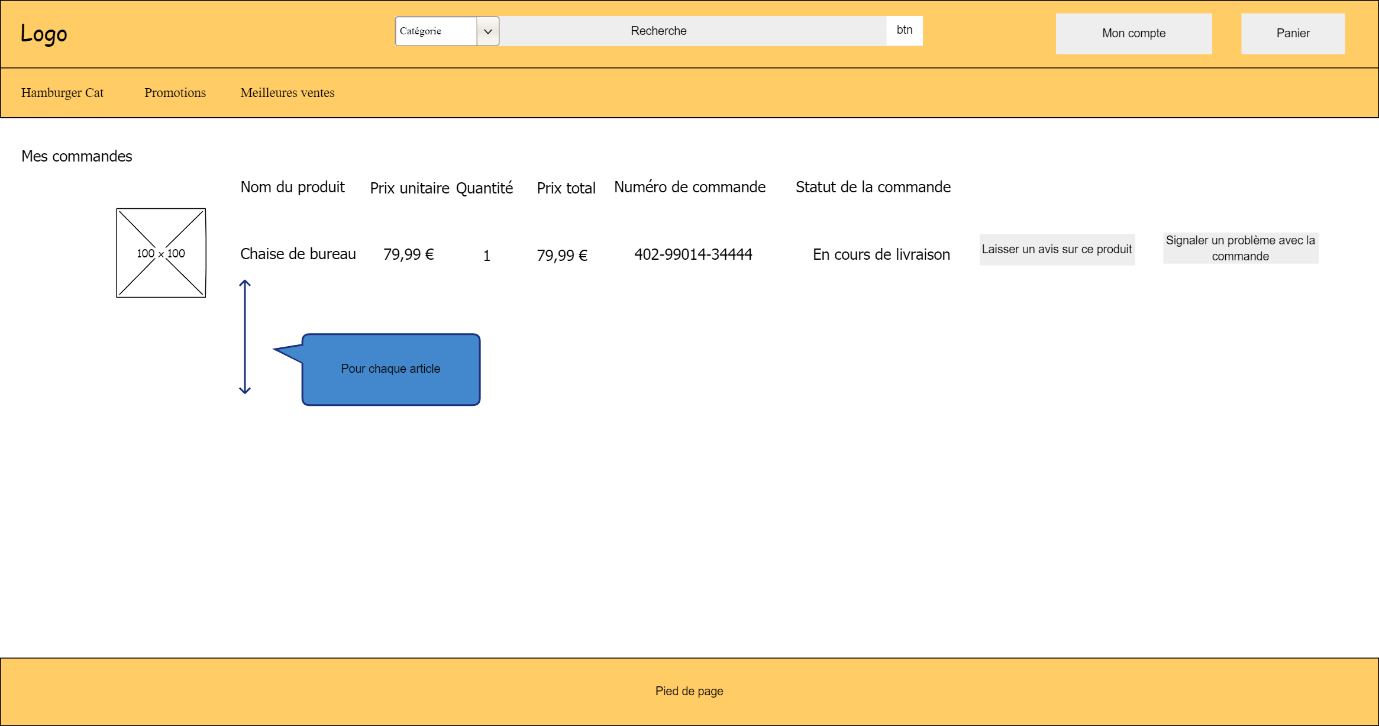


Figure 14 : Commandes

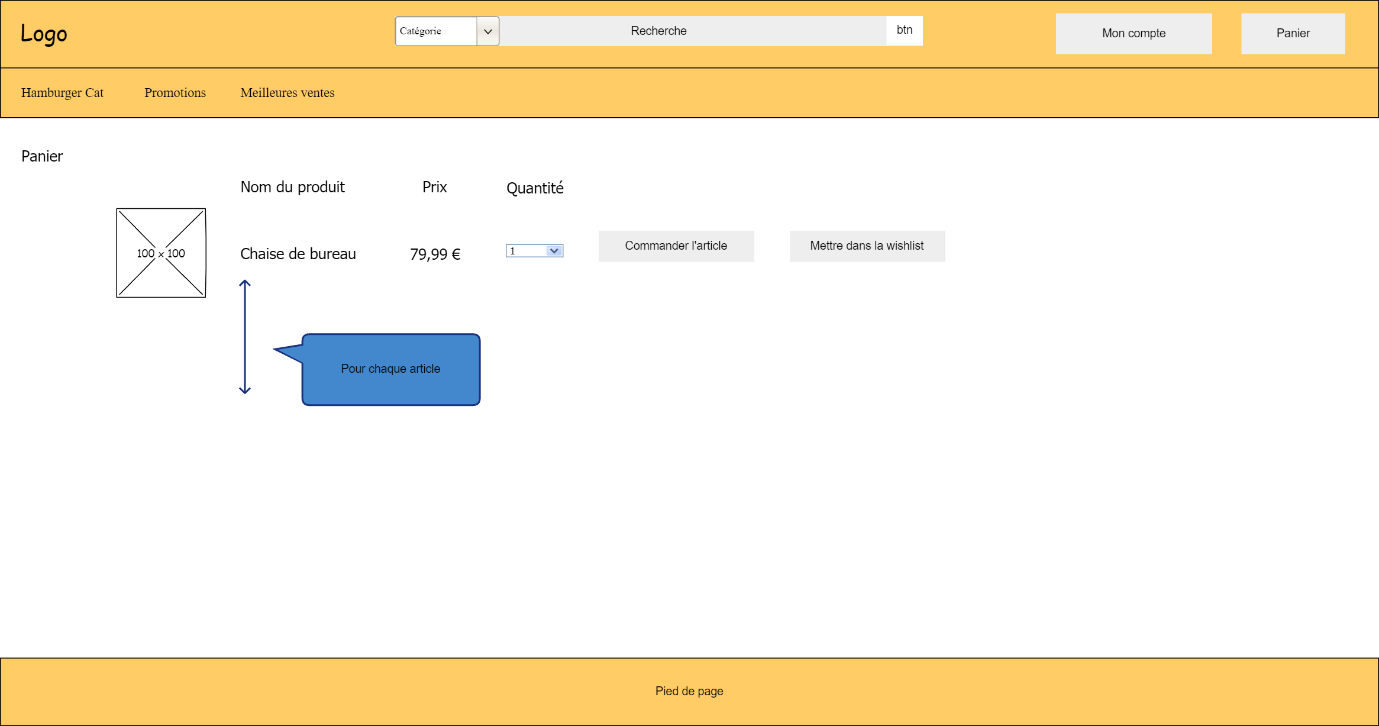


Figure 15 : Panier

## Modélisation

### Modèle conceptuel de données

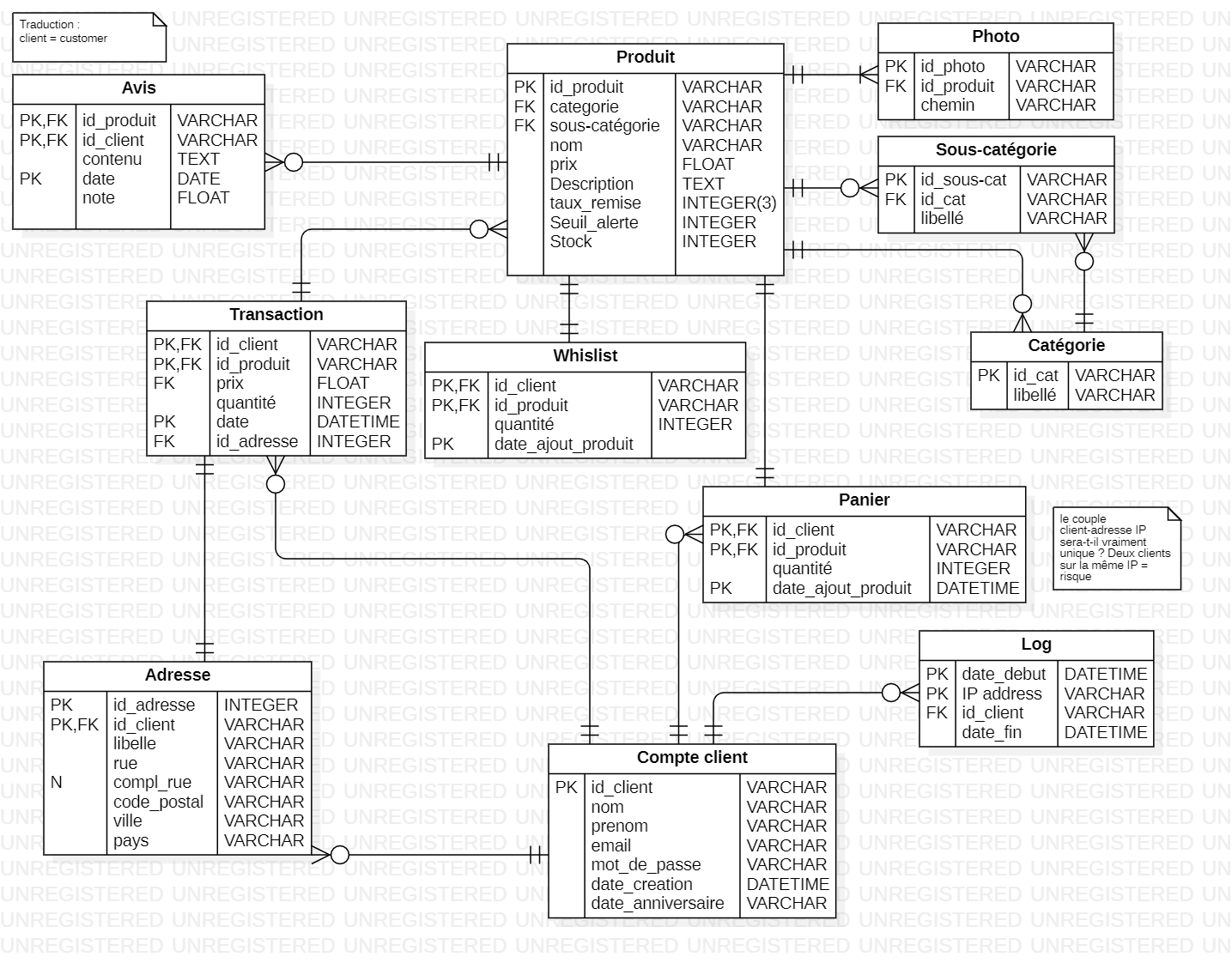


Figure 16 : Modèle conceptuel de données

### Diagramme de cas d’utilisation application desktop

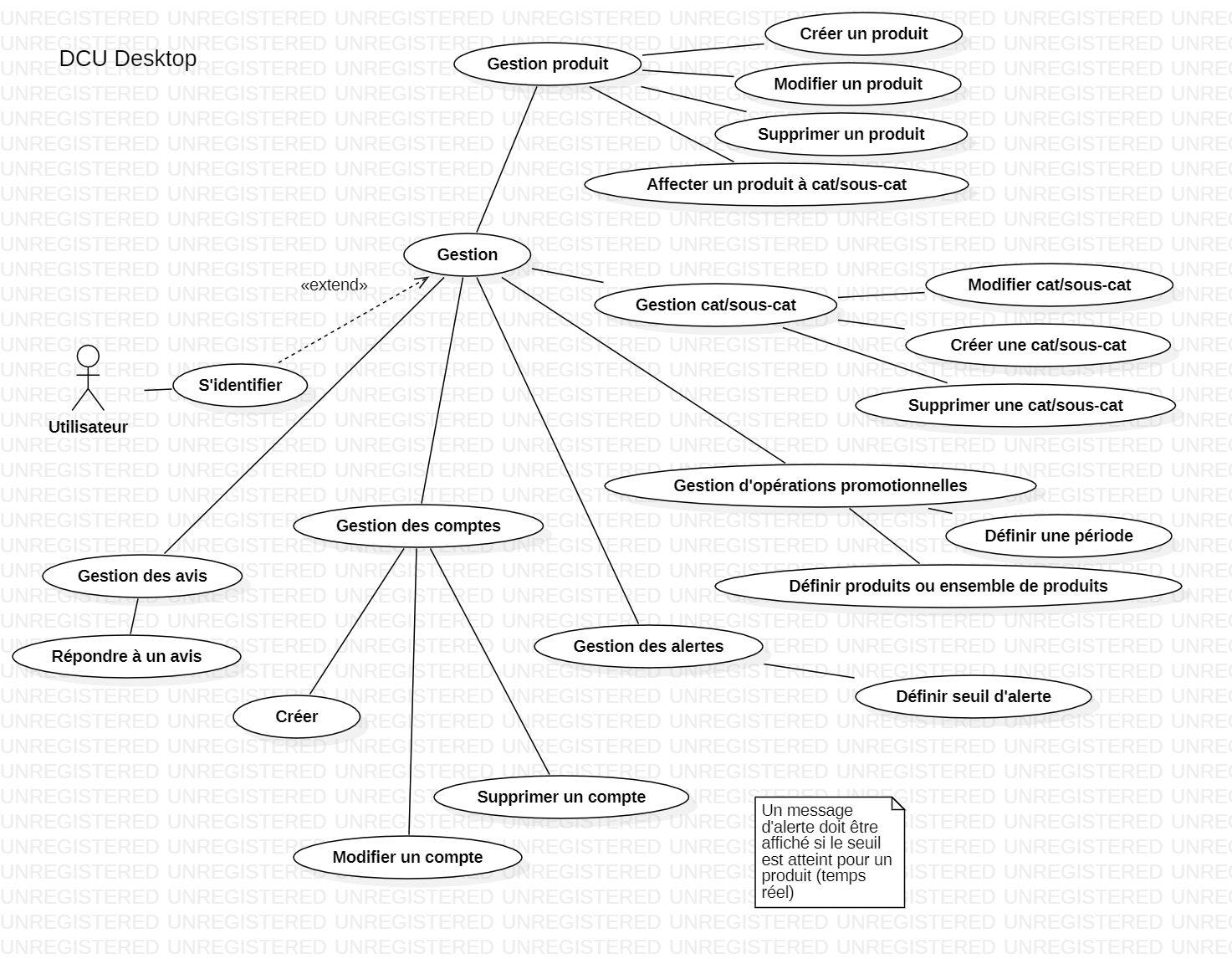


Figure 17 : DCU Desktop

### Diagramme de cas d’utilisation site web

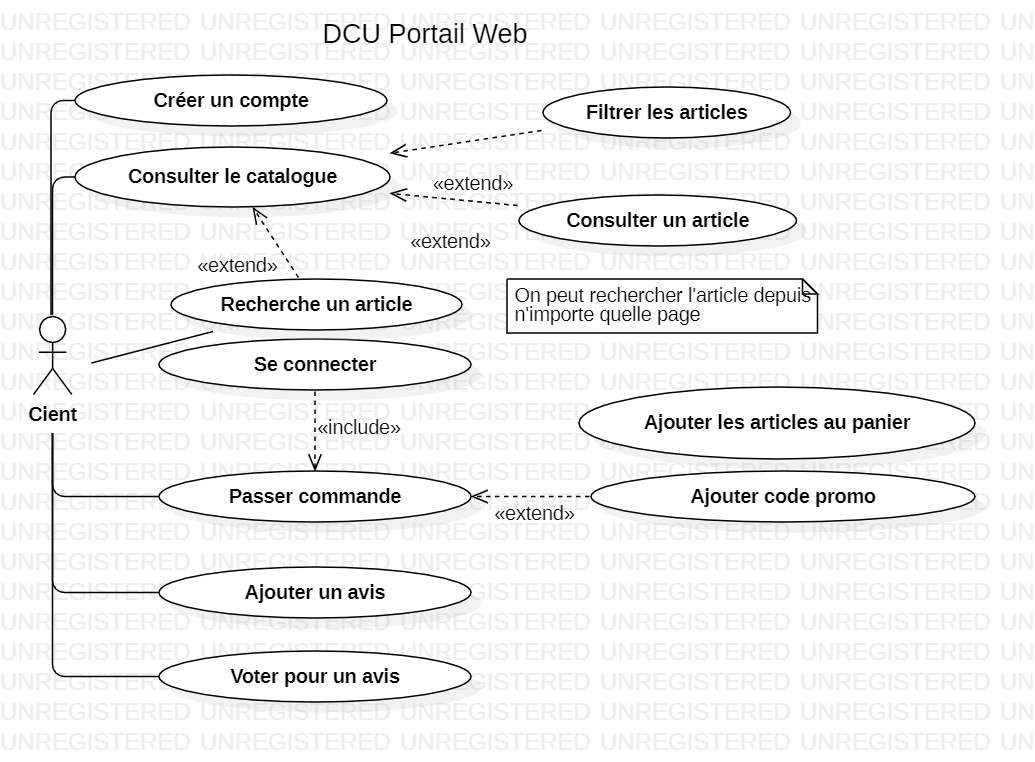
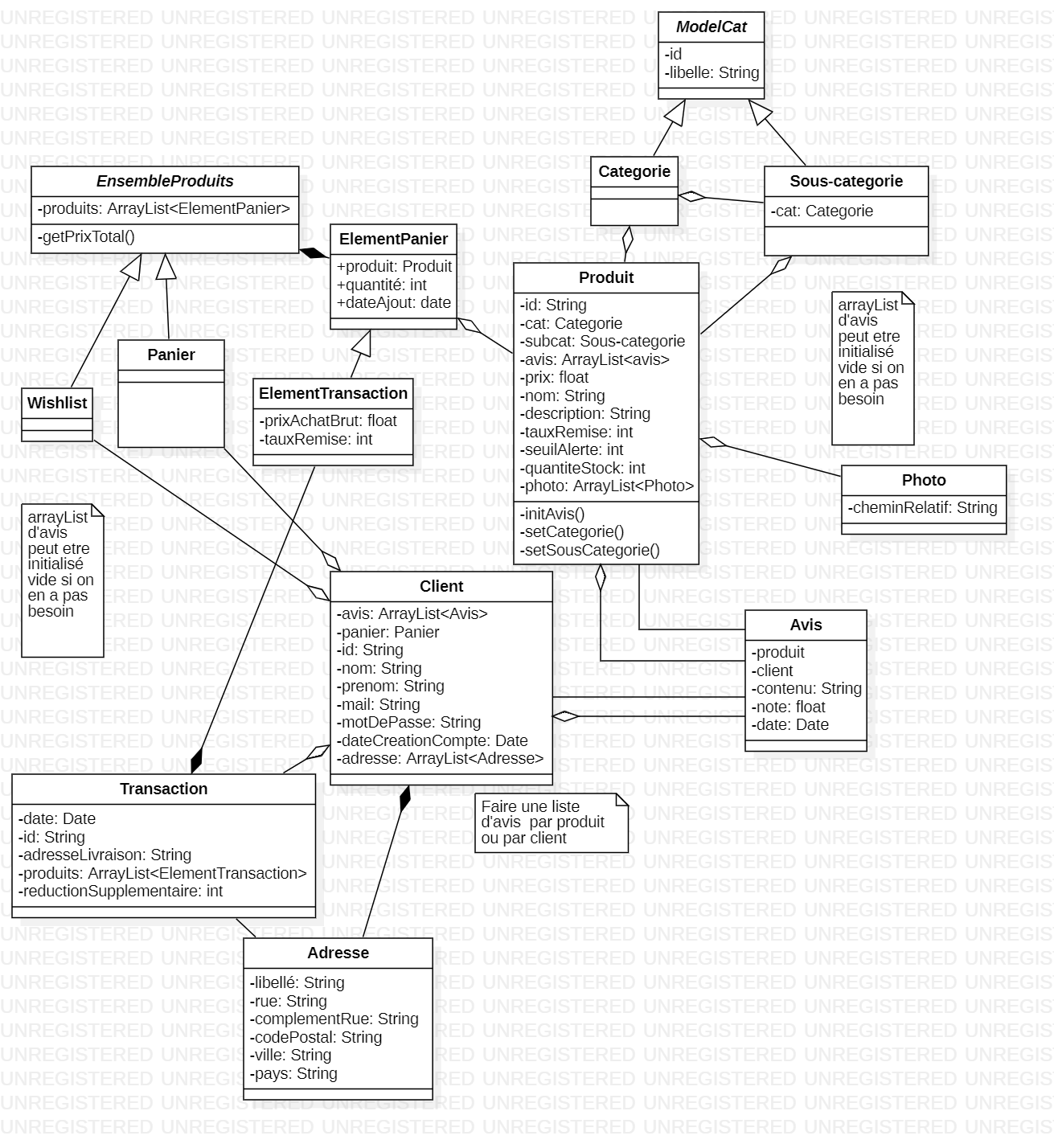


Figure 18 : DCU Portail Web

## Diagramme de classe



## Création de la base de données

### Script de création de la base de données

CREATE TABLE reviews (

PK\_FK\_product\_ID VARCHAR,

PK\_FK\_customer\_ID VARCHAR,

PK\_review\_date DATE,

review\_content TEXT,

review\_rating FLOAT NOT NULL,

CONSTRAINT PK\_review PRIMARY KEY (PK\_FK\_product\_ID, PK\_FK\_customer\_ID, PK\_date),

CONSTRAINT FK\_review\_product\_ID FOREIGN KEY (PK\_FK\_product\_ID) REFERENCES products(PK\_product\_ID),

CONSTRAINT FK\_review\_customer\_ID FOREIGN KEY (PK\_FK\_customer\_ID) REFERENCES customers(PK\_customer\_ID),

CONSTRAINT CHK\_review\_date CHECK (PK\_review\_date BETWEEN 1 AND 10)

);

CREATE TABLE adresses (

PK\_adress\_number int,

PK\_FK\_customer\_ID VARCHAR,

adress\_wording VARCHAR,

adress\_street VARCHAR NOT NULL,

adress\_street\_complement VARCHAR,

adress\_postal\_code VARCHAR NOT NULL ,

adress\_city VARCHAR NOT NULL,

adress\_country VARCHAR NOT NULL,

CONSTRAINT PK\_adress PRIMARY KEY (PK\_adress\_number, PK\_FK\_customer\_ID),

CONSTRAINT FK\_adress\_customer\_ID FOREIGN KEY (PK\_FK\_customer\_ID) REFERENCES customers(PK\_customer\_ID)

);

CREATE TABLE cart\_elements (

PK\_FK\_customer\_ID VARCHAR,

PK\_FK\_product\_ID VARCHAR,

cart\_element\_quantity INT NOT NULL,

CONSTRAINT PK\_cart PRIMARY KEY (PK\_FK\_customer\_ID, PK\_FK\_product\_ID),

CONSTRAINT FK\_cart\_element\_customer\_ID FOREIGN KEY (PK\_FK\_customer\_ID) REFERENCES customers(PK\_customer\_ID),

CONSTRAINT PK\_cart\_\_element\_product\_ID FOREIGN KEY (PK\_FK\_product\_ID) REFERENCES products(PK\_product\_ID),

CONSTRAINT CHK\_cart\_element\_quantity CHECK (cart\_element\_quantity > 0)

);

CREATE TABLE wishlist\_elements (

PK\_FK\_customer\_ID VARCHAR,

PK\_FK\_product\_ID VARCHAR,

wishlist\_element\_quantity INT NOT NULL,

CONSTRAINT PK\_wishlist PRIMARY KEY (PK\_FK\_customer\_ID, PK\_FK\_product\_ID),

CONSTRAINT FK\_whislist\_element\_customer\_ID FOREIGN KEY (PK\_FK\_customer\_ID) REFERENCES customers(PK\_customer\_ID),

CONSTRAINT FK\_wishlist\_\_element\_product\_ID FOREIGN KEY (PK\_FK\_product\_ID) REFERENCES products(PK\_product\_ID),

CONSTRAINT CHK\_wishlist\_element\_quantity CHECK (wishlist\_element\_quantity > 0)

);

CREATE TABLE transactions (

PK\_transation\_ID VARCHAR,

FK\_customer\_ID VARCHAR NOT NULL,

FK\_delivery\_adress VARCHAR NOT NULL,

transaction\_datetime DATE NOT NULL,

CONSTRAINT PK\_transaction PRIMARY KEY (PK\_transaction\_ID),

CONSTRAINT FK\_transaction\_customer\_ID FOREIGN KEY (FK\_customer\_ID) REFERENCES customers(PK\_customer\_ID),

CONSTRAINT FK\_transaction\_delivery\_adress FOREIGN KEY (FK\_delivery\_adress) REFERENCES adresses(PK\_adress\_ID),

CONSTRAINT UNIQUE\_transaction\_date\_customer UNIQUE (FK\_customer\_ID, transaction\_datetime)

);

CREATE TABLE transaction\_elements (

PK\_FK\_transaction\_ID VARCHAR,

PK\_FK\_product\_ID VARCHAR,

transaction\_element\_price FLOAT NOT NULL,

transaction\_element\_quantity INT NOT NULL,

CONSTRAINT PK\_transaction\_element PRIMARY KEY (PK\_FK\_transaction\_ID, PK\_FK\_product\_ID),

CONSTRAINT FK\_transaction\_element\_transaction\_ID FOREIGN KEY (PK\_FK\_transaction\_ID) REFERENCES transactions(PK\_transaction\_ID),

CONSTRAINT FK\_transaction\_\_element\_product\_ID FOREIGN KEY (PK\_FK\_product\_ID) REFERENCES products(PK\_product\_ID),

CONSTRAINT CHK\_transaction\_element\_quantity CHECK (transaction\_element\_quantity > 0),

CONSTRAINT CHK\_transaction\_element\_price CHECK (transaction\_element\_price >= 0)

);

CREATE TABLE logs (

PK\_ip\_adress VARCHAR,

PK\_FK\_customer\_ID VARCHAR,

log\_session\_starting TIME NOT NULL,

log\_session\_ending TIME NOT NULL,

CONSTRAINT PK\_log PRIMARY KEY (PK\_ip\_adress, PK\_FK\_customer\_ID),

CONSTRAINT FK\_log\_customer\_id FOREIGN KEY (PK\_FK\_customer\_ID) REFERENCES customers(PK\_customer\_ID),

CONSTRAINT CHK\_log\_session CHECK (log\_session\_starting < log\_session\_ending)

);

/\*===============================================================================================================\*/

CREATE TABLE customers (

PK\_customer\_ID CHAR(15),

customer\_lastName VARCHAR(255) NOT NULL,

customer\_firstName VARCHAR(255) NOT NULL,

customer\_email VARCHAR(255) NOT NULL,

customer\_passwordHash VARCHAR(255) NOT NULL,

customer\_accountCreationDate DATE NOT NULL,

customer\_birthday DATE NOT NULL,

CONSTRAINT C\_PK\_customer PRIMARY KEY (PK\_customer\_ID),

CONSTRAINT C\_UC\_customer\_email UNIQUE (customer\_email)

);

CREATE TABLE categories (

PK\_cat\_ID CHAR(15),

cat\_wording VARCHAR(255) NOT NULL,

CONSTRAINT C\_PK\_category PRIMARY KEY (PK\_cat\_ID)

);

CREATE TABLE subcategories (

PK\_subcat\_ID CHAR(15),

FK\_subcat\_cat\_ID CHAR(15),

subcat\_wording VARCHAR(255) NOT NULL,

CONSTRAINT C\_PK\_subcategory PRIMARY KEY (PK\_subcat\_ID),

CONSTRAINT C\_FK\_subcategory\_cat\_ID FOREIGN KEY (FK\_subcat\_cat\_ID) REFERENCES categories(PK\_cat\_ID)

);

CREATE TABLE products (

PK\_product\_ID CHAR(15),

FK\_product\_cat\_ID CHAR(15) NOT NULL,

FK\_product\_subcat\_ID CHAR(15),

product\_wording VARCHAR(255) NOT NULL,

product\_price FLOAT NOT NULL,

product\_description TEXT,

product\_discountRate INT NOT NULL,

product\_alertThreshold INT NOT NULL,

product\_stock INT NOT NULL,

CONSTRAINT C\_PK\_product PRIMARY KEY (PK\_product\_ID),

CONSTRAINT C\_FK\_product\_cat\_ID FOREIGN KEY (FK\_product\_cat\_ID) REFERENCES categories(PK\_cat\_ID),

CONSTRAINT C\_FK\_product\_subcat\_ID FOREIGN KEY (FK\_product\_subcat\_ID) REFERENCES subcategories(PK\_subcat\_ID)

);

CREATE TABLE pictures (

PK\_picture\_ID CHAR(10),

FK\_picture\_product\_ID CHAR(10),

picture\_path VARCHAR(255) NOT NULL,

CONSTRAINT C\_PK\_picture PRIMARY KEY (PK\_picture\_ID),

CONSTRAINT C\_FK\_picture\_product\_ID FOREIGN KEY (FK\_picture\_product\_ID) REFERENCES products(PK\_product\_ID)

);

### Script d’insertion d’un jeu de données test

INSERT INTO customers VALUES ('CST100000000000','DA SILVA', 'Alex', 'alexdasilva@gmail.com', 'azerty', '2022-04-01', '2002-09-04');

INSERT INTO customers VALUES ('CST100000000001', 'REICHEN', 'Anthony', 'anthonyreichen@gmail.com', '1234', '2022-04-02', '1993-04-01');

INSERT INTO customers VALUES ('CST100000000002', 'CHEFFE', 'Carlos', 'carloscheffe@gmail.com', '0000', '2022-05-10', '1999-02-02');

INSERT INTO categories VALUES ('CAT000000100000', 'Ordinateur');

INSERT INTO categories VALUES ('CAT000000100001', 'Accessoires');

INSERT INTO categories VALUES ('CAT000000100002', 'Meubles');

INSERT INTO subcategories VALUES ('SBC000000100000', 'CAT000000100000', 'PC Portable');

INSERT INTO subcategories VALUES ('SBC000000100001', 'CAT000000100001', 'Souris');

INSERT INTO subcategories VALUES ('SBC000000100002', 'CAT000000100002', 'Chaise Gaming');

INSERT INTO products VALUES ('PRD000100000000', 'CAT000000100000', 'SBC000000100000', 'Asus ROG Zephyrus', 2995.94,

'Un PC bien mais très très cher !', 0, 15, 1);

INSERT INTO products VALUES ('PRD000100000001', 'CAT000000100001', 'SBC000000100001', 'Razer Pro Tournament', 69.99,

NULL, 10, 19, 34);

INSERT INTO products VALUES ('PRD000100000002', 'CAT000000100002', 'SBC000000100002', 'Cloud Gaming Super Chair', 230.5,

'Pour ne pas avoir mal au dos', 0, 10, 18);

Table des matières

[Méthodologie de gestion de projet : 1](#_Toc103291456)

[Choix de la méthode Agile 1](#_Toc103291457)

[Organisation des cycles de développement 1](#_Toc103291458)

[Cycle 1 : Modélisation des cas d’utilisation 1](#_Toc103291459)

[Cycle 2 : Modélisation de la base de données 2](#_Toc103291460)

[Cycle 3 : Modélisation des classes 2](#_Toc103291461)

[Cycle 4 : Modélisation des IHM 2](#_Toc103291462)

[Cycle 5 : Création de la base de données 3](#_Toc103291463)

[Cycle 6 : Application JAVA 3](#_Toc103291464)

[Cycle 7 : Application PHP 4](#_Toc103291465)

[Annexes : 5](#_Toc103291466)

[Méthodologie de gestion de projet 5](#_Toc103291467)

[Diagramme de Gantt 5](#_Toc103291468)

[Diagramme de ressources 5](#_Toc103291469)

[Kanban 6](#_Toc103291470)

[Maquettes d’interfaces homme-machine 7](#_Toc103291471)

[Maquettes de l’application Java 7](#_Toc103291472)

[Maquettes du site web 9](#_Toc103291473)

[Modélisation 13](#_Toc103291474)

[Modèle conceptuel de données 13](#_Toc103291475)

[Diagramme de cas d’utilisation application desktop 14](#_Toc103291476)

[Diagramme de cas d’utilisation site web 15](#_Toc103291477)

[Diagramme de classe 16](#_Toc103291478)

[Création de la base de données 17](#_Toc103291479)

[Script de création de la base de données 17](#_Toc103291480)

[Script d’insertion d’un jeu de données test 23](#_Toc103291481)