

Projet tutoré ASPE

GESTION DE SITE E-COMMERCE

Anthony REICHEN & Alex DA SILVA

IUT LYON 1 – DUT INFORMATIQUE ANNEE SPECIALE | 2021 - 2022

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 destinée 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 mêmes 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 présentes 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 framework 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

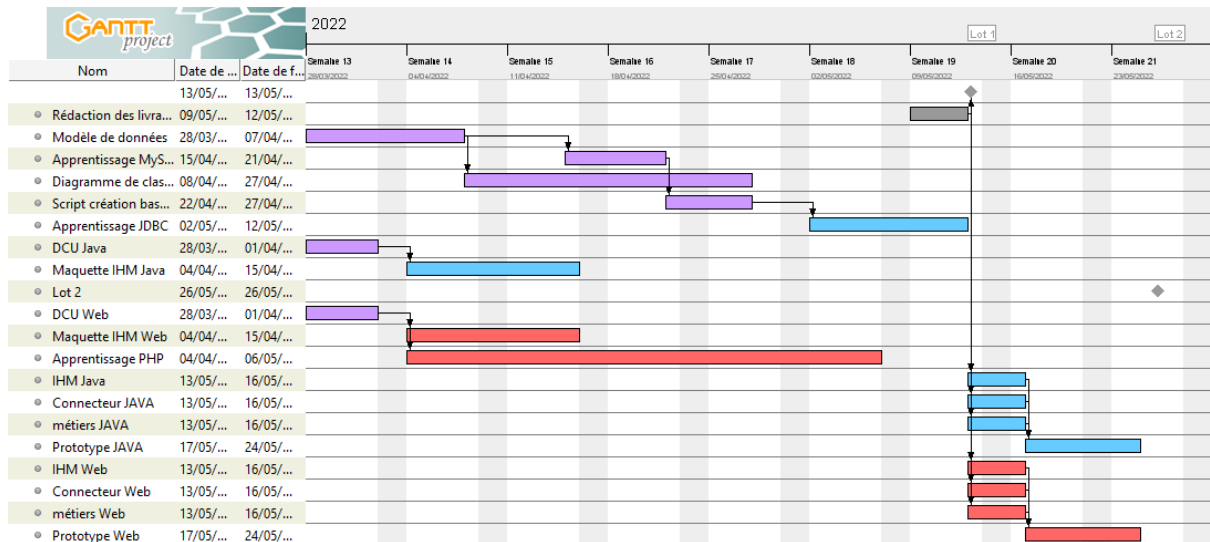


Figure : 1 Diagramme de Gantt

Diagramme de ressources

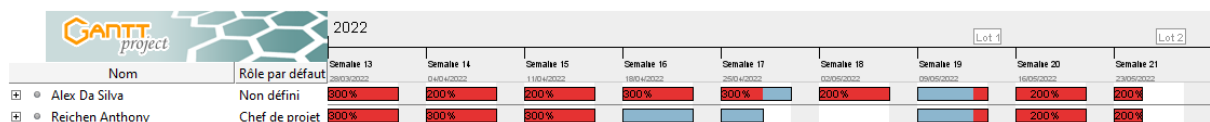


Figure : 2 Diagramme de ressources

Kanban

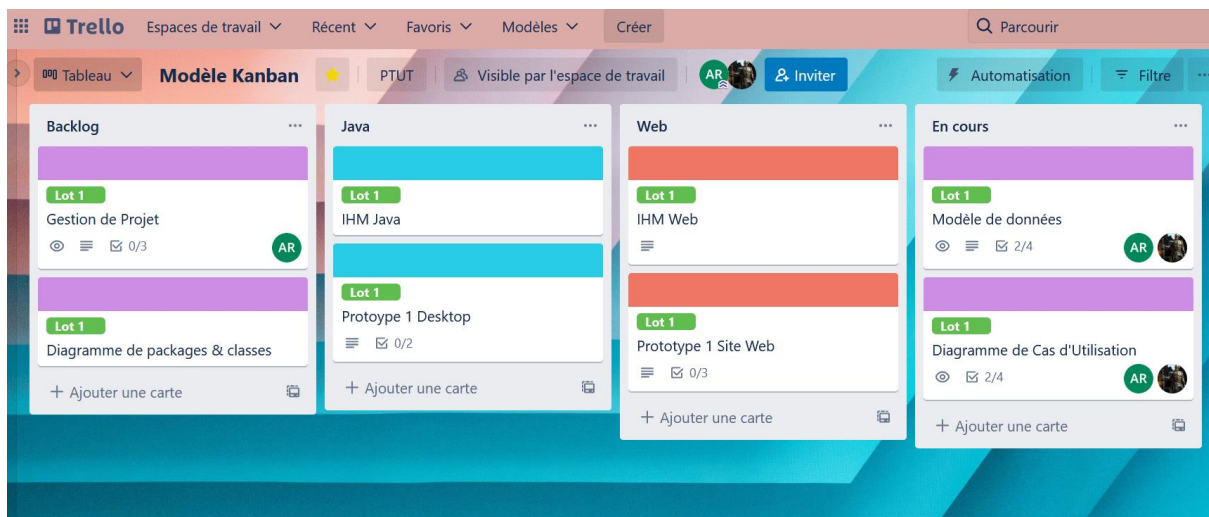


Figure 3 : Capture d'écran du Kanban (site Trello)

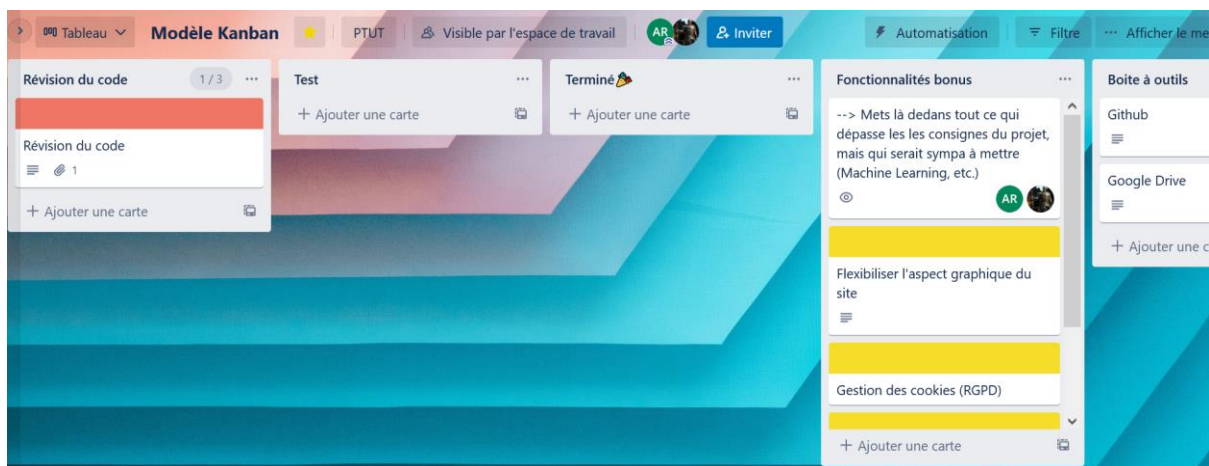


Figure 4 : Capture d'écran du Kanban (site Trello)

Maquettes d'interfaces homme-machine

Maquettes de l'application Java

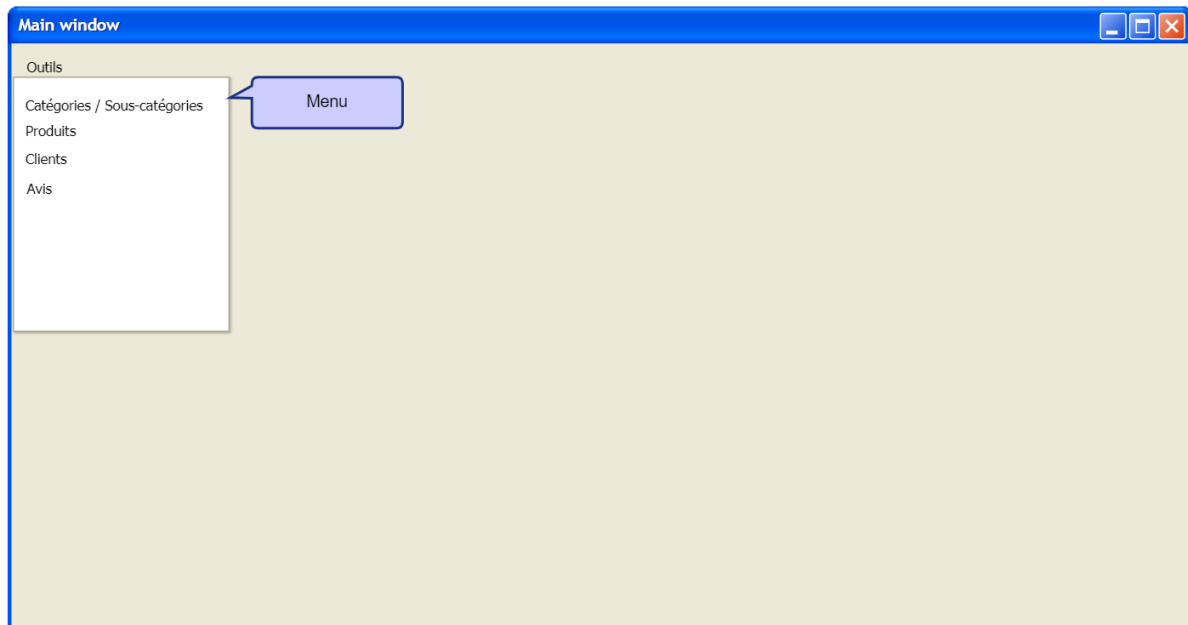


Figure 5 : menu de l'application

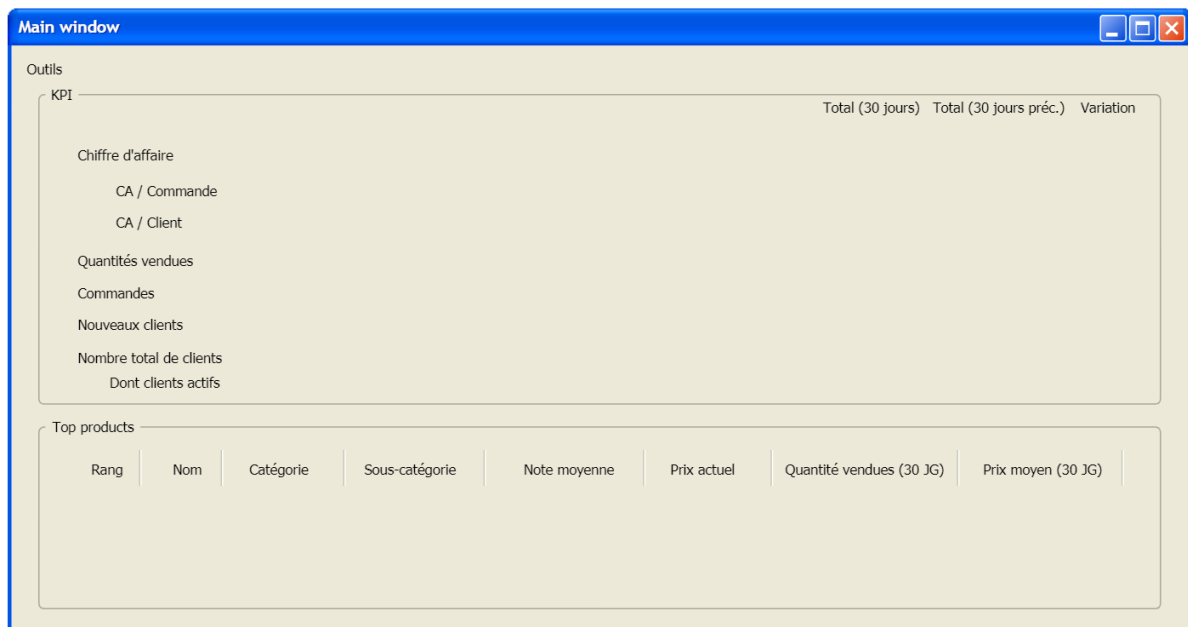


Figure 6 : page d'accueil de l'application

Ajout catégorie

Ajout de catégories et sous-catégories :

Catégories

Nom de la catégorie: Mobilier

Nom de la catégorie: Mobilier

+

Annuler Confirmer

Sous-catégories

Catégorie parente : Mobilier

Nom de la sous-catégorie: Literie

Nom de la sous-catégorie: Rangement

+

Annuler Confirmer

Figure 7 : ajout catégorie

Ajout produit

Ajout de produit

Informations du produit

Nom: Plumärh

Catégorie: Mobilier

Sous-catégorie: Literie

Prix: 14,99

Seuil Alerte: 50

Description: Text box

Group

Nom	Cat.	SCat.	Prix	Seuil Aler.	Desc.

Annuler Continuer la saisie Confirmer

Figure 8 : ajout produit

Client

Compte client

Informations du client

Nom: DA SILVA

Prénom: Alex

adresse mail: ads@wanadoo.fr

Prix: 14,99

Seuil Alerte: 50

Adresse: 23 rue du bourg 69100 VILLEURBANNE

Achats effectués

Commande	Prix total	Statut
Détails commande		

Annuler Continuer la saisie Confirmer

Figure 9 : gestion client

Maquettes du site web

Logo Catégorie Recherche btn Mon compte Panier

Hamburger Cat Promotions Meilleures ventes

Click
Menu apparaît par la gauche

No Image

Nom du produit
Note
Prix

Articles se répètent sur pls lignes

1...2...3...X

Pied de page

Figure 10 : Page d'accueil

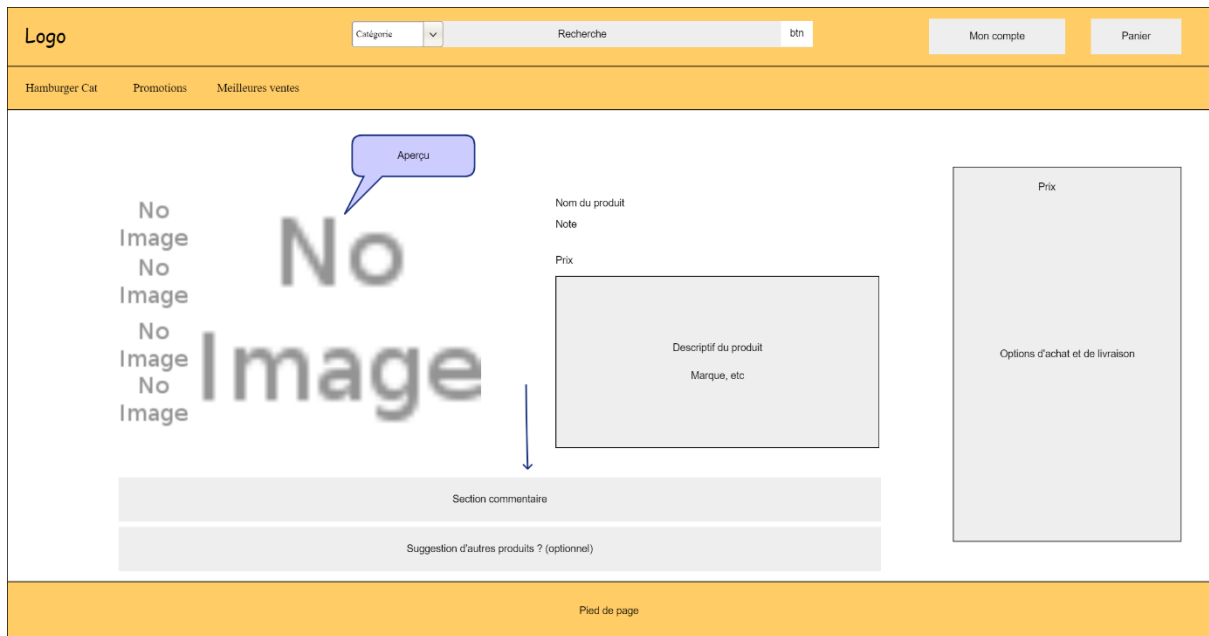


Figure 11 : Détails du produit

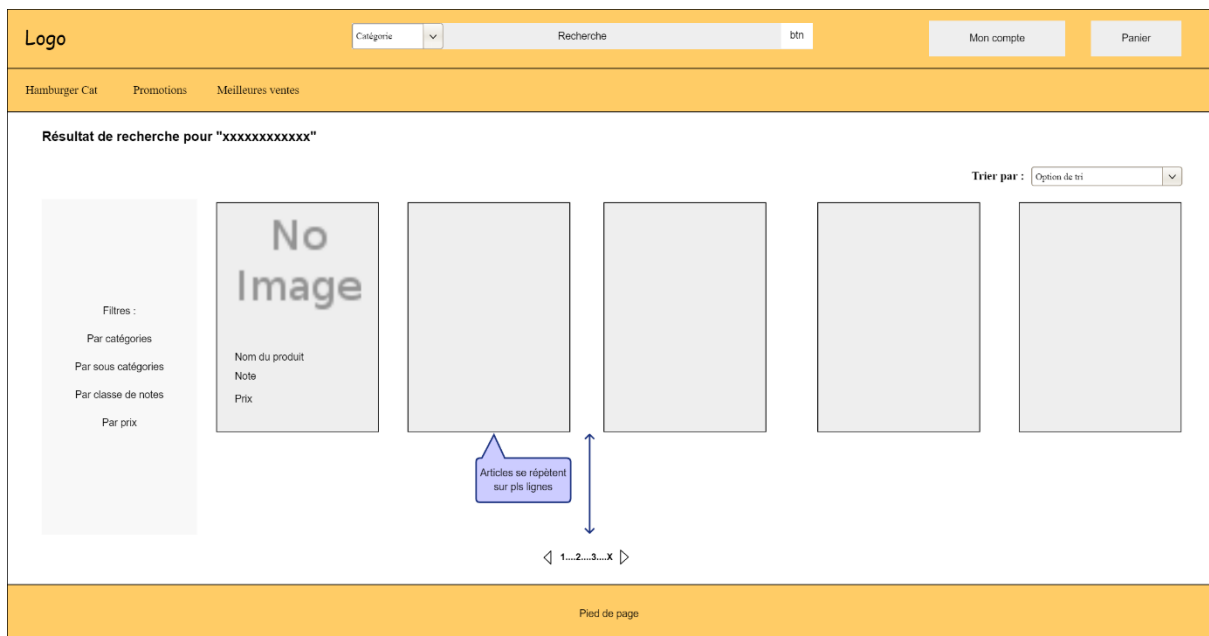


Figure 12 : Résultat de recherche

Logo

Catégorie

Recherche

btn

Mon compte

Panier

Hamburger Cat

Promotions

Meilleures ventes

Nom :

Text box

Prénom :

Text box

Adresse mail :

Text box

Sujet :

Combo box

Champ à remplir si sujet "autre" sélectionné

Message :

Text box

Ces trois champs ne sont pas visible lorsque l'utilisateur est connecté

Pied de page

Figure 13 : Formulaire de contact

Logo

Catégorie

Recherche

btn

Mon compte

Panier

Hamburger Cat

Promotions

Meilleures ventes

Mes commandes

	Nom du produit	Prix unitaire	Quantité	Prix total	Numéro de commande	Statut de la commande	
<div> <div>100 x 100</div> <div> <div></div> <div>Pour chaque article</div> </div> </div>	Chaise de bureau	79,99 €	1	79,99 €	402-99014-34444	En cours de livraison	<div>Laisser un avis sur ce produit</div> <div>Signaler un problème avec la commande</div>

Pied de page

Figure 14 : Commandes

Logo

Catégorie ▼


Recherche btn

Mon compte

Panier

Hamburger CatPromotionsMeilleures ventes

Panier

	Nom du produit	Prix	Quantité		
	Chaise de bureau	79,99 €	<input type="text" value="1"/>	Commander l'article	Mettre dans la wishlist

Pour chaque article

Pied de page

Figure 15 : Panier

Modèle conceptuel de données



Diagramme de cas d'utilisation application desktop

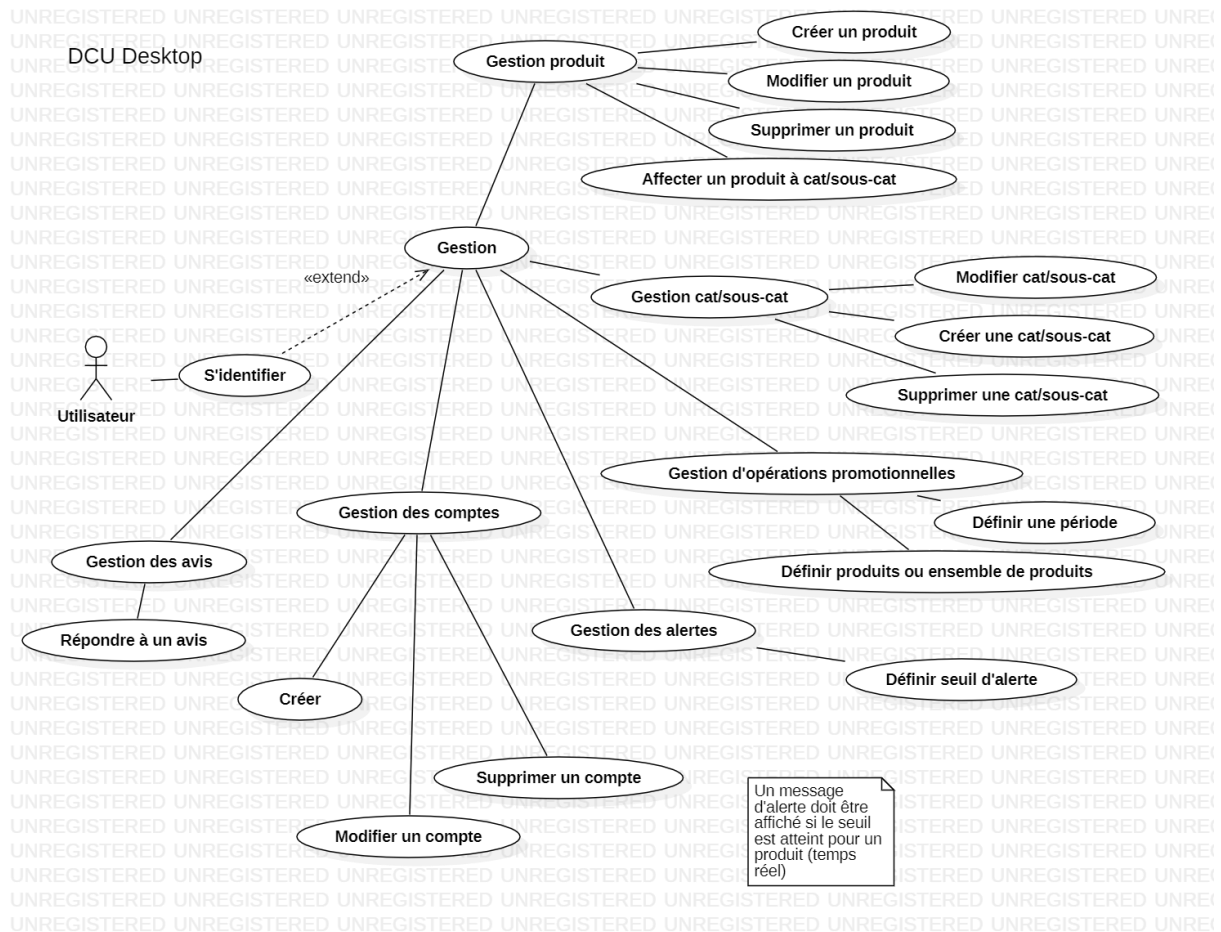


Figure 17 : DCU Desktop

Diagramme de cas d'utilisation site web

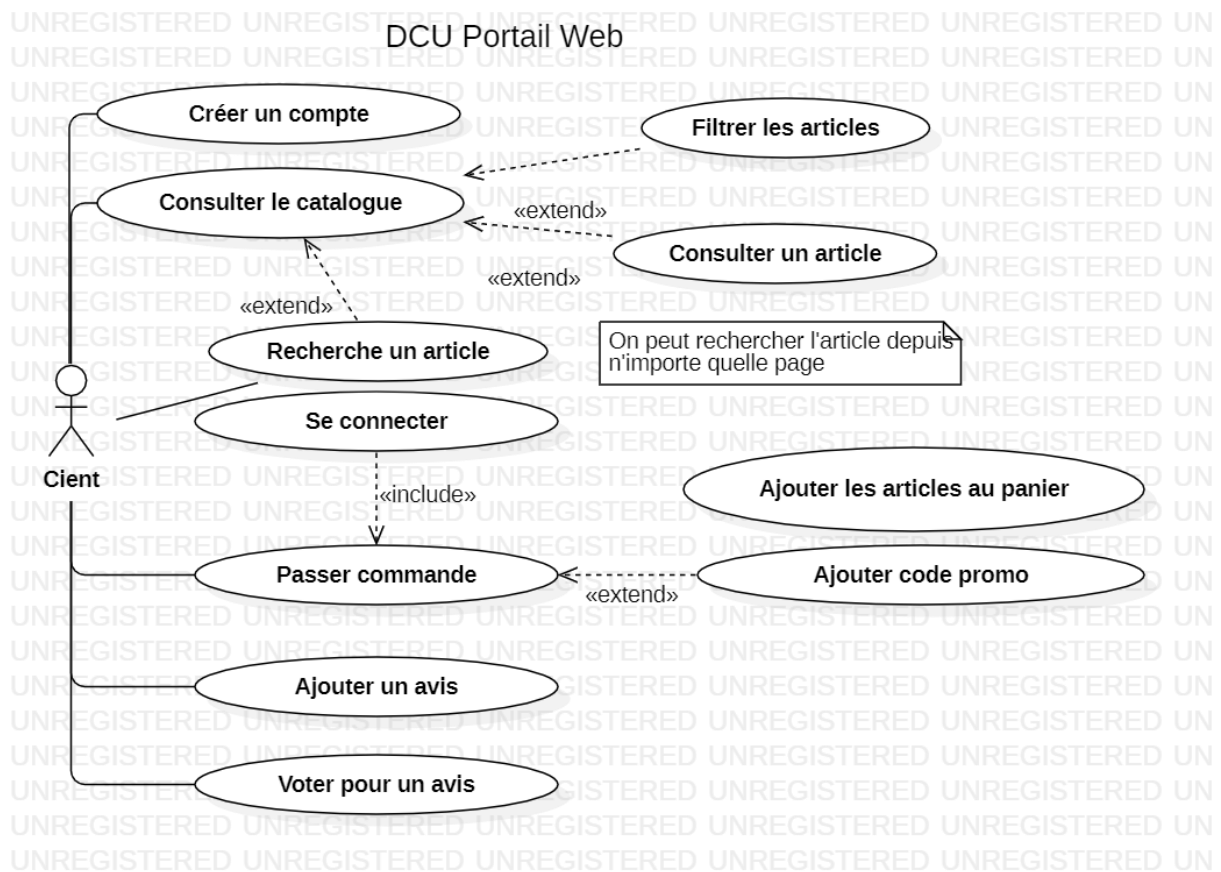


Figure 18 : DCU Portail Web


```

classDiagram
    class ModelCat {
        -id
        -libelle: String
    }
    class Categorie {
    }
    class Sous_categorie {
        -cat: Categorie
    }
    class Produit {
        -id: String
        -cat: Categorie
        -subcat: Sous_categorie
        -avis: ArrayList<avis>
        -prix: float
        -nom: String
        -description: String
        -tauxRemise: int
        -seuilAlerte: int
        -quantiteStock: int
        -photo: ArrayList<Photo>
        -initAvis()
        -setCategorie()
        -setSousCategorie()
    }
    class Avis {
        -produit
        -client
        -contenu: String
        -note: float
        -date: Date
    }
    class Client {
        -avis: ArrayList<Avis>
        -panier: Panier
        -id: String
        -nom: String
        -prenom: String
        -mail: String
        -motDePasse: String
        -dateCreationCompte: Date
        -adresse: ArrayList<Adresse>
    }
    class Adresse {
        -libellé: String
        -rue: String
        -complementRue: String
        -codePostal: String
        -ville: String
        -pays: String
    }
    class Transaction {
        -date: Date
        -id: String
        -adresseLivraison: String
        -produits: ArrayList<ElementTransaction>
        -reductionSupplementaire: int
    }
    class ElementTransaction {
        -prixAchatBrut: float
        -tauxRemise: int
    }
    class Panier {
    }
    class EnsembleProduits {
        -produits: ArrayList<ElementPanier>
        -getPrixTotal()
    }
    class ElementPanier {
        +produit: Produit
        +quantité: int
        +dateAjour: date
    }
    class Wishlist {
    }
    class Photo {
        -cheminRelatif: String
    }

    ModelCat <|-- Categorie
    ModelCat <|-- Sous_categorie
    Categorie o-- Sous_categorie
    Categorie o-- Produit
    Sous_categorie o-- Produit
    Produit o-- Avis
    Produit o-- Photo
    Client o-- Avis
    Client o-- Panier
    Client o-- Adresse
    Client o-- Transaction
    Avis o-- Produit
    Avis o-- Client
    Transaction o-- Adresse
    Transaction o-- ElementTransaction
    EnsembleProduits <|-- Panier
    EnsembleProduits <|-- Wishlist
    EnsembleProduits <|-- ElementPanier
    
```

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
Choix de la méthode Agile	1
Organisation des cycles de développement.....	1
Cycle 1 : Modélisation des cas d'utilisation.....	1
Cycle 2 : Modélisation de la base de données	2
Cycle 3 : Modélisation des classes.....	2
Cycle 4 : Modélisation des IHM.....	2
Cycle 5 : Création de la base de données	3
Cycle 6 : Application JAVA.....	3
Cycle 7 : Application PHP	4
Annexes :	5
Méthodologie de gestion de projet	5
Diagramme de Gantt.....	5
Diagramme de ressources.....	5
Kanban	6
Maquettes d'interfaces homme-machine.....	7
Maquettes de l'application Java	7
Maquettes du site web	9
Modélisation.....	13
Modèle conceptuel de données.....	13
Diagramme de cas d'utilisation application desktop	14
Diagramme de cas d'utilisation site web	15
Diagramme de classe	16
Création de la base de données	17
Script de création de la base de données	17
Script d'insertion d'un jeu de données test	23