



Projet de Fin De Module : E-Epicier

Réalisé Par :

M'SAFRI Kaouthar
CHALKHOUNE Oumaima
ANNAKI Oussama

M. EL Encadré Par :

AACAHAK Lotfi

Année universitaire : 2022/2023

À nos parents,

“Progress is impossible without change.”

Walt Disney

REMERCIEMENTS :

En préambule à ce rapport, on souhaite adresser nos remerciements les plus sincères à toutes les personnes qui ont contribué à son élaboration.

On tient à remercier M. EL AACHAK Lotfi, notre encadrant du projet pour ses précieux conseils qui nous ont guidé tout au long de la rédaction de ce rapport.

On tient à remercier nos très chers parents, qui ont été toujours là pour nous, « Vous avez tout sacrifié pour vos enfants n'épargnant ni santé ni efforts. Vous nous avez donné un magnifique modèle de labeur et de persévérance. On est redevable d'une éducation dont on est fier »

Nos vifs remerciements aussi à Nos amis, collègues pour leur encouragement.

Nous adressons nos remerciements également à toute personne qui a de près ou de loin participé à collaborer à l'existence de ce travail.

Un grand merci à tous !

Table des figures :

Figure 1 : diagramme de bête à cornes	12
Figure 2 : lien entre les diagrammes UML	17
Figure 3 : diagramme d'utilisation de l'épicier.....	20
Figure 4: diagramme de cas d'utilisation de client.....	21
Figure 5 : diagramme de classe.....	22
Figure 6 : Home page.....	26
Figure 7 : guide d'utilisation	27
Figure 8 : guide d'utilisation	27
Figure 9 : guide d'utilisation	28
Figure 10 : guide d'utilisation	28
Figure 11 : formulaire d'inscription d'épicier.....	29
Figure 12 : formulaire de créer une épicerie	30
Figure 13 : consultation des épiceries	31
Figure 14 : formulaire d'enregistrement d'un client	31
Figure 15: consultation des clients.....	32
Figure 16 : formulaire d'ajouter un produit.....	33
Figure 17 : consultation des produits	33
Figure 18 : formulaire de création d'un nouveau crédit	34
Figure 19 : consultation des crédits	34

Table des matières :

REMERCIEMENTS :	5
Introduction Générale :	10
Présentation du projet :	11
I. Situation initiale :	11
II. Problématique et objectifs :	11
1. Problématique :	11
2. Objectifs :	12
3. Besoins fonctionnels :	13
a) La gestion de confidentialité :	13
b) La gestion des crédits et articles par l'épicier :	13
c) La consultation des crédits par client :	13
4. Besoins non fonctionnels :	14
b) Ergonomie :	14
c) Accessibilité :	14
5. Méthodologie de développement :	14
b) Phases du Processus Unifié :	15
c) Activités du processus unifié :	15
6. Langage de modélisation :	16
b) Liens entre les diagrammes :	17
Conception :	18
I. Les acteurs du système :	18
1. Définition :	18
2. Identification des acteurs :	18

b)	L'épicier :	18
c)	Le client :	18
3.	Identification des cas d'utilisation :	19
4.	Cas d'utilisation :	19
5.	Diagramme de cas d'utilisations :	20
b)	Diagramme de cas d'utilisation d'épicier :	20
c)	Diagramme de cas d'utilisation du client :	21
6.	Réalisation du diagramme de classe	22
b)	Diagramme de classes :	22
	Bilan :	22
	Étude technique et implémentation :	23
I.	Application web :	23
1.	Avantages d'une application web :	23
2.	Outils de développement :	23
b)	Django v4.1.7 :	24
c)	Bootstrap v5.3.0 :	24
d)	MYSQL version de serveur 10.4.27-MariaDB - mariadb.org binary distribution :	24
e)	XAMPP control panel v3.3.0 :	24
f)	PYTHON v3.11.2 :	25
g)	VSCODE v1.75.1 :	25
h)	App Diagram :	25
II.	Représentation des interfaces de l'application :	26
	Page home :	26
	Interface d'épicier :	28
	Interface de gestion des épiciers :	29

Interface de gestion des clients :	31
Interface de gestion des produits :	32
Interface client :	35
Conclusion :	37
Conclusion générale.....	38
Documentations :	39

Introduction Générale :

De nos jours, le monde connaît une avance technologique importante dans multiples secteurs et cela et grâce aux sciences informatiques qui étudient les techniques du traitement automatique de l'information, elles jouent un rôle considérable dans le développement des sociétés et plusieurs autres établissements.

Avant l'apparition de l'ordinateur, on enregistrait toutes informations ou bien données manuellement, généralement sur des supports en papier, ce qui engendrait beaucoup de difficultés en matière de stockage, organisation et accessibilité.

Ainsi, à l'air de la digitale et avec le développement démesuré des méthodes et périphériques de stockage, l'ordinateur reste le moyen le plus sûr pour le traitement et la sauvegarde de l'information.

Les épiciers, grands ou petits, ils font une partie des établissements que l'informatique et la digitalisation pourrait beaucoup les aider. On effet, la diversité des produits et marque ainsi que la demande de crédit de la part des clients à leurs épiciers, nécessite une organisation et une certaine rigueur dans la « gestion des données », or à ce jour, la manière de gérer manuellement est encore dominante.

Cahier, ou bien feuille, ou même sur un bout de papier, l'épicier enregistre les achats des clients pour être payé ultérieurement, cela engendre un grand risque de perte et d'inexactitude mettant ainsi la fidélité du client au péril, d'où la nécessité d'introduire l'usage de l'informatique dans les épiciers.

Vu cet état de fait, nôtre projet de fin de module a pour objectif de concevoir et mettre en œuvre une web application interactive et conviviale, facile à intégrer dans l'environnement de travail des épiciers, assurant ainsi une bonne gestion et regroupement des données ainsi une certaine visibilité aux clients pour prévisualiser leurs crédits.

Cette web application vise essentiellement à diminuer les risques de pertes de données et à améliorer le suivi des crédits des clients.

I. Situation initiale :

La majorité des épiciers qu'on connaît accordent des crédits par carnet à leurs clients. L'épicer du quartier accepte d'accorder des facilités de paiement pour ses clients fréquents avec un délai de paiement non précisé.

Le carnet de crédit chez l'épicer a toujours existé dans la société marocaine. Même avec la prolifération des grandes surfaces et des supermarchés, ce procédé n'a jamais disparu.

II. Problématique et objectifs :

Après avoir fait l'étude de marché, nous allons analyser des problèmes rencontrés par les épiciers puis nous allons tenter d'apporter des solutions.

1. Problématique :

La gestion des crédits dans une épicerie se fait manuellement, en effet pour ajouter un nouveau crédit il faut qu'il cherche sur ses carnets et ses papiers pour trouver le client correspondant à ce crédit, ce qui engendre plusieurs problèmes tels que :

- Une perte de temps dans la recherche des clients.
- Les erreurs de calcul engendrées par la gestion manuelle.
- Utilisation de plusieurs documents ce qui entraîne une mauvaise organisation de ces derniers ou même perte dans certains cas.
- Confusion des produits achetés avec les produits enregistrés.

Dans l'intention de résoudre ces problèmes, nous proposons de développer une application web permettant aux épiciers d'atteindre les objectifs présentés dans le point suivant.

2. Objectifs :

La phase d'analyse a pour objectif de décrire de manière précise, concise, correcte et compréhensible les besoins et les exigences du client. Il s'agit de livrer des spécifications pour permettre la conception de la solution. La phase d'analyse permet de s'accorder sur « ce que doit faire l'application web »

Les épiciers veulent un logiciel qui va pouvoir exécuter les tâches précédentes.

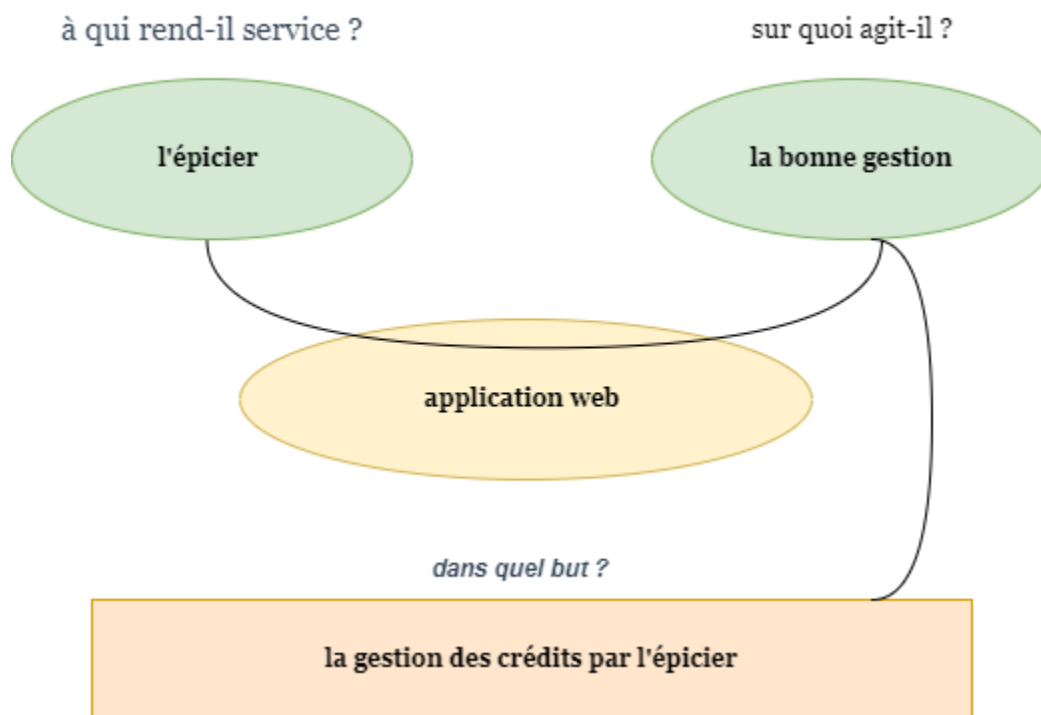


Figure 1 : diagramme de bête à cornes

3. Besoins fonctionnels :

Les besoins fonctionnels se rapportent aux fonctionnalités que l'application doit offrir pour satisfaire les épicier.

Les fonctionnalités que doit intégrer l'application à développer sont :

a) La gestion de confidentialité :

Le système permet de gérer les droits d'accès de chaque utilisateur en associant un ID et un mot de passe pour chaque utilisateur (l'épicier et le client).

b) La gestion des crédits et articles par l'épicier :

Cette opération consiste à consulter à tout temps le compte épicier pour constater la situation d'épicerie, les crédits des clients et leurs informations. L'épicier enregistre tous les achats par crédits en précisant l'/les article(s), le prix, la quantité la date de l'achat et le délai de paiement. Également, l'épicier aura la possibilité de préciser la disponibilité des articles.

c) La consultation des crédits par client :

Tout client peut consulter son compte pour constater ses crédits avec une précision du délai de paiement, la quantité achetée, la date d'achat, en addition il aura accès à vérifier la disponibilité ou bien l'existence d'un produit souhaiter.

4. Besoins non fonctionnels :

a) Authentification par un login et mot de passe :

L'application contient deux interfaces, la première interface correspond à l'épicier, ce dernier peut soit créer un compte ainsi commencer son expérience, soit s'authentifier avec un ID et un mot de passe. La deuxième interface concerne le client, il peut consulter à partir de l'ID qui sera fourni par l'épicier.

b) Ergonomie :

Le système devra offrir aux utilisateurs (l'épicier et le client) une interface qui soit le plus riche possible afin de limiter le nombre d'écrans. Par ailleurs, l'interactivité devra être adaptée.

c) Accessibilité :

L'application doit être mobile c'est à dire que l'épicier où le client peut accéder à cette dernière en dehors de l'épicerie.

5. Méthodologie de développement :

Selon la nature de notre projet, nous avons jugés que PU (processus unifié) serait le plus adapté pour sa réalisation.

a) Définition du Processus Unifié :

PU ou UP (Unified Process), est une méthode orientées objets pour le développement des logiciels, c'est une méthodes générique itérative et incrémental, piloté par les cas d'utilisation centré sur l'architecture.

Le processus unifié fournit un cadre au développement logiciel pour la construction de systèmes orientés objet.

b) Phases du Processus Unifié :

La méthode UP se base sur quatre phases :

- Analyse des besoins :
Établir une vision globale du projet où on spécifie les besoins et on étudie la faisabilité du projet.
- Élaboration :
On reprend les éléments de l'analyse des besoins et on développe une architecture de référence, les risques et la plupart des besoins sont identifiés.
- Construction :
Finaliser l'analyse, la conception, l'implémentation et les tests puis transformer l'architecture de référence en produit exécutable tout en veillant à respecter son intégrité.
- Transition :
Livraison du produit au client afin d'effectuer des essais pour détecter d'éventuelles anomalies.

c) Activités du processus unifié :

Chaque phase est constituée d'une succession d'activités. Les activités de la méthode UP sont :

- Expression des besoins :
Compréhension et expression des besoins et des exigences du client qu'elle soit fonctionnelle ou non fonctionnelle.
- Analyse et Conception :
Permet d'acquérir une compréhension approfondie des contraintes liées aux outils de réalisation en prenant en compte le choix d'architecture technique retenu pour le développement et l'exploitation système.
- Implémentation :
On implémente le système sous forme de composants, bibliothèques et de fichiers. Elle a pour objectif de planifier l'intégration.
- Tests :
Permettent de vérifier les résultats de l'implémentation de toutes les exigences et de s'assurer de la bonne intégration de tous les composants dans le logiciel.

6. Langage de modélisation :

Sachant que le processus unifié exige l'utilisation d'UML, notre modélisation de la solution se fera en utilisant des diagrammes UML.

a) Unified Modeling Language (UML) :

UML (Unified Modeling Language) est un langage de modélisation objet né de la fusion de trois langages de modélisation objet (Booch, OMT, OOSE) qui ont révolutionnés la modélisation dans les années 90.

UML est dit universel car il est indépendant des langages de programmation, des domaines d'application et aussi du processus de développement adopté.

Il permet de représenter un système sous forme de schémas. Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de la solution.

Sa véritable force repose sur un méta modèle qui normalise la sémantique des concepts, qu'il véhicule. Notez qu'UML est ouvert et n'est la propriété de personne.

Diagramme	Objectifs	Type
Diagramme de classes	1. Point central de la modélisation du système pour décrire ce que le système doit faire (analyse) et avec quoi il va le faire (conception). 2. Représentation de la structure statique du système d'information. 3. Modélisation des classes et de leurs relations.	Statique
Diagramme de cas d'utilisation	1. Décrire la manière dont une organisation ou un système externe doivent interagir avec le système. 2. décrire ce que doit faire le système. 3. Mettre en évidence les services rendus par le système.	Fonctionnel

b) Liens entre les diagrammes :

La figure suivante représente les liens entre les diagrammes à concevoir dans le chapitre suivant :

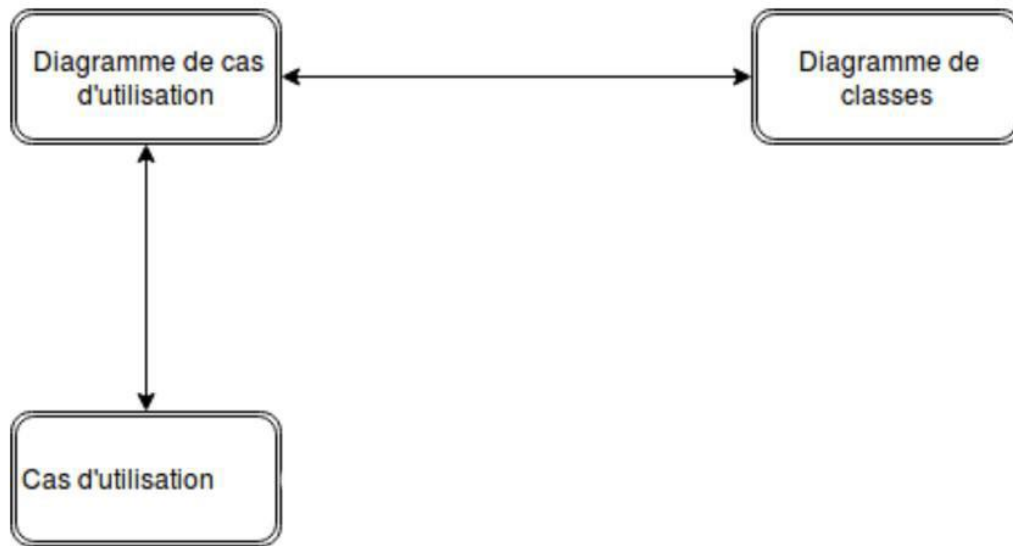


Figure 2 : lien entre les diagrammes UML

Conception :

Dans ce chapitre dédié à la conception, nous allons définir le rôle de chaque acteur qui interagit avec le système. Nous allons aussi modéliser leurs rôles sous forme de diagrammes de cas d'utilisation, puis nous définirons les cas d'utilisation et nous les modéliserons sous forme de diagramme de classe.

I. Les acteurs du système :

1. Définition :

Un acteur est une entité externe qui interagit avec le système (Utilisateur, dispositif matériel, ou autre système...). En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin (modification du système ou simple consultation). Les acteurs peuvent être classés hiérarchiquement

2. Identification des acteurs :

a) Les acteurs principaux :

Dans notre système le client et l'épicier ne jouent pas le même rôle et doivent avoir des diagrammes d'utilisation séparées.

b) L'épicier :

Le rôle d'épicier est de :

- Gérer le compte client /épicier
- Gérer les crédits.
- Gérer les produits
- Gérer l'épicerie

c) Le client :

Le rôle de client est limité :

- Consulter son compte.
- Consulter les crédits.
- Consulter les produits.

3. Identification des cas d'utilisation :

Dans l'application a développé, nous avons définis les cas d'utilisation suivant :

Acteur	Cas d'utilisation
Épicier / Client	Authentification
Épicier	Gérer le compte client (consulter, modifier, ajouter, supprimer). Gérer les crédits (consulter, modifier, ajouter, supprimer). Gérer l'épicerie (consulter, modifier, ajouter, supprimer). Gérer les produits (consulter, modifier, ajouter, supprimer).
Client	Consulter : le compte, les crédits et les produits.

4. Cas d'utilisation :

- Authentification : Cette opération permet à un utilisateur de s'authentifier avant d'accéder à l'application, elle permet aussi de s'assurer de l'identité de l'utilisateur.
- Gérer les comptes des clients : permet à l'épicier de créer un compte pour son client a fin qu'il peut le consulter, ainsi de modifier ses coordonnées, et consulter sa situation.
- Gérer les crédits : permet à l'épicier d'ajouter, supprimer, enregistrer ; consulter et modifier les crédits.
- Gérer les produits : permet à l'épicier d'ajouter, supprimer, enregistrer modifier et consulter les produits de son épicerie.
- Gérer l'épicerie : permet à l'épicier d'ajouter, supprimer, enregistrer modifier et consulter son épicerie.

5. Diagramme de cas d'utilisations :

a) Définition :

Les diagrammes de cas d'utilisation décrivent les utilisations requises d'un système, ou ce qu'un système est supposé faire. Les principaux concepts de ces diagrammes sont les acteurs, cas d'utilisation et sujets. Un sujet représente un système avec lequel les acteurs et autres sujets interagissent. Le comportement requis du sujet est décrit par les cas d'utilisation

b) Diagramme de cas d'utilisation d'épicier :

Le diagramme ci-dessous représente les cas d'utilisations identifiés pour l'épicier qui est l'épicier :

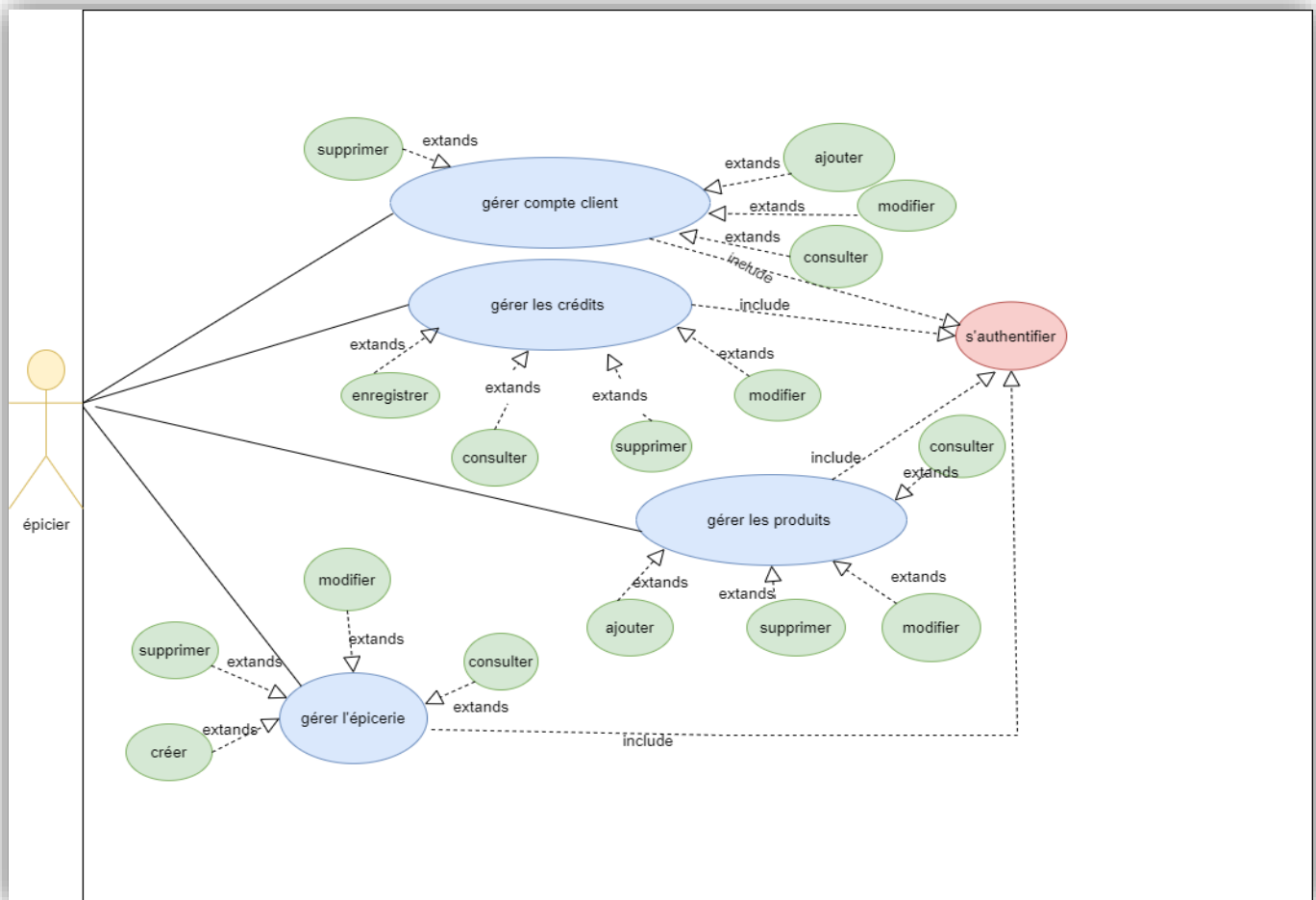


Figure 3 : diagramme d'utilisation de l'épicier

c) Diagramme de cas d'utilisation du client :

Le diagramme ci-dessous représente les cas d'utilisations identifiés pour le client :

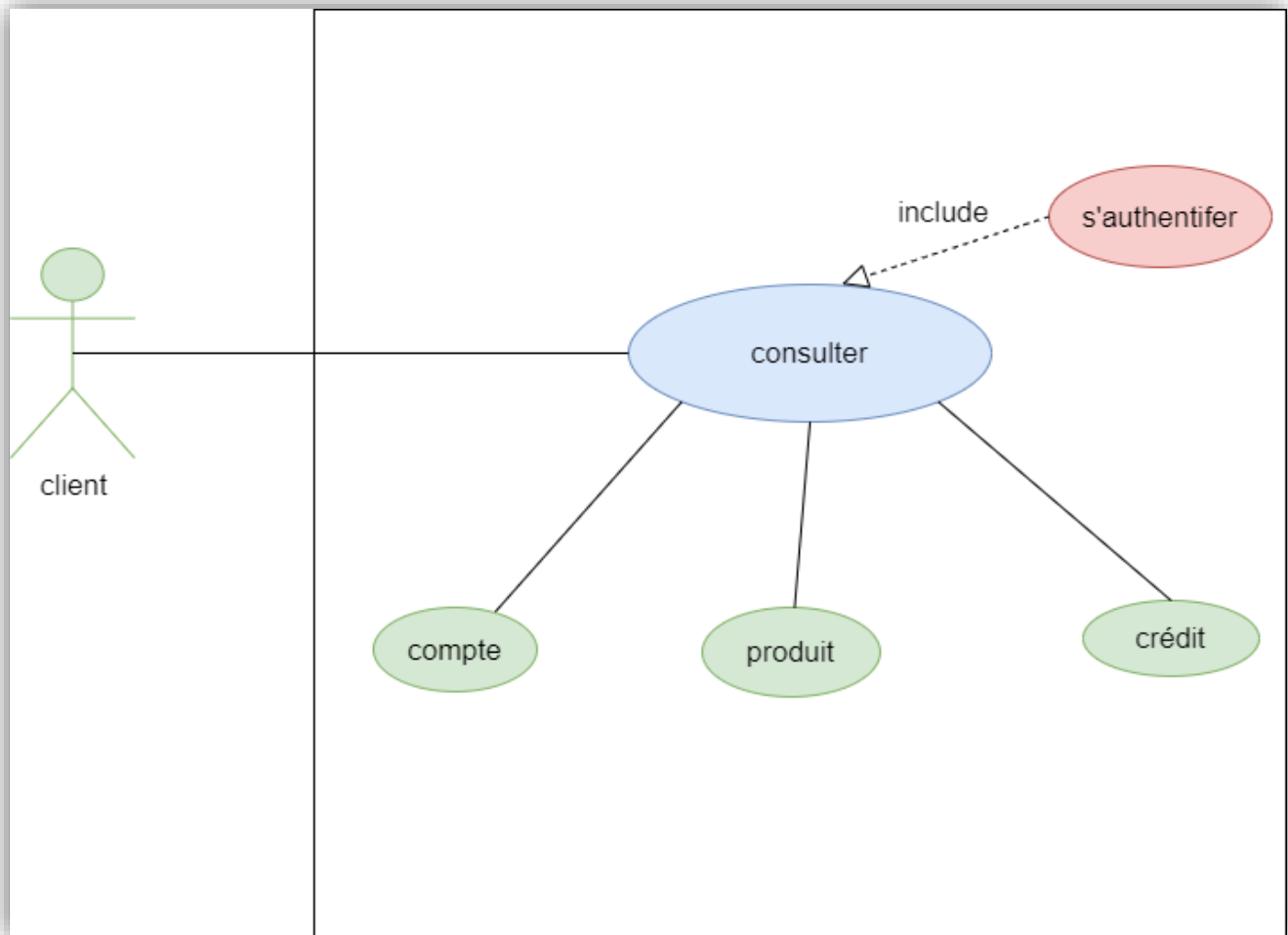


Figure 4: diagramme de cas d'utilisation de client

6. Réalisation du diagramme de classe

a) DEFINITION :

Le diagramme de classes est sans doute le diagramme le plus important à représenter pour les méthodes d'analyse orientées objet. En effet, il permet de spécifier QUI intervient à l'intérieur du système.

Un diagramme de classes fait abstraction des aspects dynamiques et temporels du système, il va permet de représenter une vue statique du système d'information. Il s'agit plutôt des relations entre les classes, des services rendus et utilisés par chacune d'elles et de l'articulation de l'ensemble.

b) Diagramme de classes :

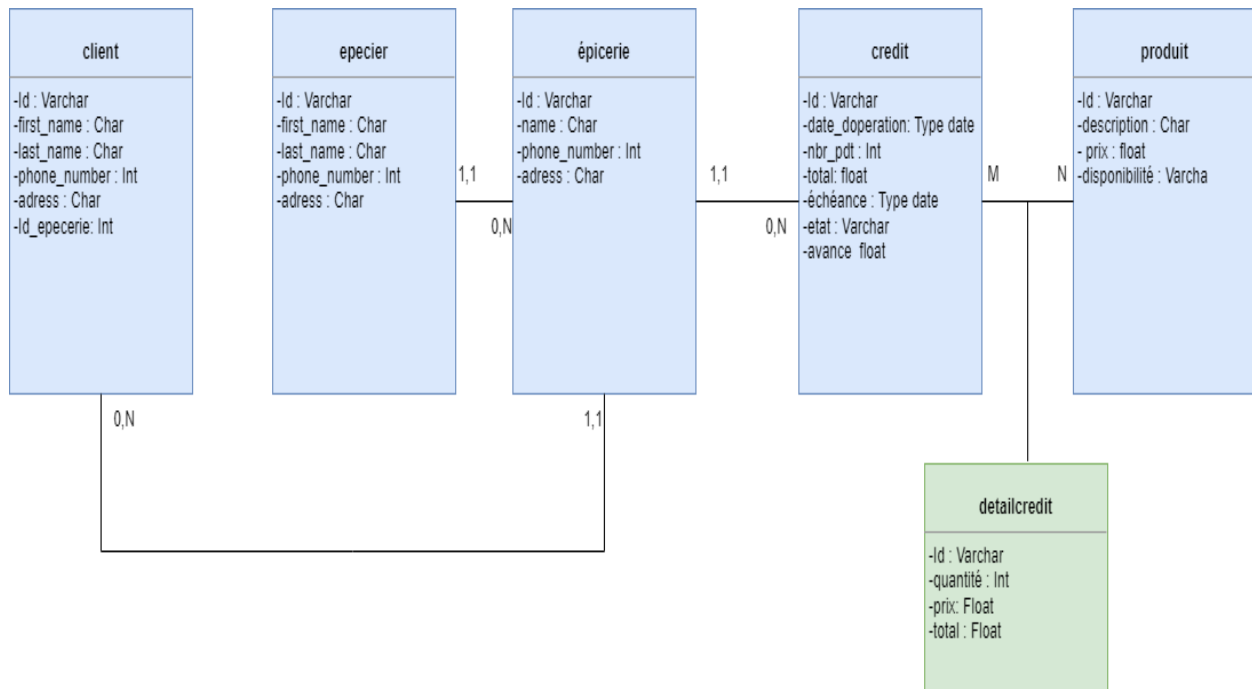


Figure 5 : diagramme de classe

Bilan :

Dans ce chapitre, nous avons modélisé notre application en apportant des réponses à nos questions de modélisation et de conception. En s'appuyant sur l'analyse des besoins de notre application. Nous avons pu là modéliser sous forme de diagrammes de cas d'utilisation, puis concevoir notre application sous forme de diagrammes de classe, reste à concevoir et a créé notre application

Dans ce chapitre dédié à l'étude technique et à l'implémentation, nous avons commencé à définir les outils de développement utiliser pour l'implémentation de notre application. Ensuite nous passerons à la présentation de l'application puis on finira par une conclusion.

I. Application web :

Une application web désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web. Contrairement à un logiciel traditionnel, l'utilisateur d'une application web n'a pas besoin de l'installer sur son ordinateur. Il lui suffit de se connecter à l'application à l'aide de son navigateur

1. Avantages d'une application web :

- Accès universel depuis n'importe quel type de poste : PC, portables, téléphone mobile, tablette
- Aucune incompatibilité de système d'exploitation (il suffit d'avoir un navigateur) ;
- Travailler depuis n'importe quel endroit de la planète ;
- Les données sont centralisées ;
- Les données sont disponibles 24h sur 24 et 7j sur 7 ;
- Aucun risque de perte de données.

2. Outils de développement :

a) Figma desktop 116.6.3 :

Figma est un outil de design graphique collaboratif basé sur le Cloud. Il fonctionne sur les MacOS, mais également sur les PC exécutant Windows ou Linux. C'est un outil qui vous propose un plan gratuit ou vous pouvez créer et stocker 3 projets actifs à la fois.

En effet, Figma est une application web d'édition graphique qui permet le partage en temps réel sur le même fichier, ce qui signifie que toutes les parties prenantes du projet peuvent interagir ensemble et travailler en collaboration sur un projet qui prend en compte chaque mise à jour et les modifications. C'est évidemment un gain de temps et un facteur d'efficacité.

Figma est pris en charge par une grande communauté de développeurs et de concepteurs qui partagent régulièrement des plug-ins, des fonctionnalités et des projets. Il est utilisé par de grandes entreprises comme Twitter et Zoom.

b) Django v4.1.7 :

Django est un framework Web avancé écrit en Python qui utilise le modèle architectural MVC (Model View Controller). Django a été créé dans un environnement de salle de rédaction en évolution rapide, et son objectif principal est de faciliter le développement de sites Web complexes, basés sur des bases de données. Ce framework Web a été initialement développé pour The World Company pour la gestion de certains de leurs sites orientés actualités. En juillet 2005, il a été rendu public sous une licence BSD.

c) Bootstrap v5.3.0 :

Bootstrap est un framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Ce framework est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran, et en priorité pour les smartphones. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore. On appelle ce type de framework un "Frontend Framework".

d) MYSQL version de serveur 10.4.27-MariaDB - mariadb.org binary distribution :

MySQL est une base de données relationnelle libre qui a vu le jour en 1995 et très employée sur le Web, souvent en association avec PHP (langage) et Apache (serveur web). MySQL fonctionne indifféremment sur tous les systèmes d'exploitation (Windows, Linux, Mac OS notamment).

Le principe d'une base de données relationnelle est d'enregistrer les informations dans des tables qui représentent des regroupements de données par sujets (table des produits, table d'utilisateur par exemple). Les tables sont reliées entre elles par des relations.

e) XAMPP control panel v3.3.0 :

XAMPP est un ensemble de logiciels qui permet de mettre en place facilement un serveur Web confidentiel, un serveur FTP et un serveur de messagerie électronique. Simple d'utilisation, il est à la portée d'un grand nombre de personnes puisqu'il ne demande aucune connaissance particulière.

XAMPP contient principalement les utilitaires suivants :

- Le logiciel serveur Apache
- MariaDB (fork de MYSQL) pour la gestion des bases de données
- PHP et PERL pour gérer les scripts et les contenus dynamiques

f) PYTHON v3.11.2 :

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages.

g) VSCODE v1.75.1 :

Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'auto complétion, la coloration syntaxique, le débogage, et les commandes git.

h) App Diagram :

Pour réaliser les diagrammes UML qui ont servis à modéliser notre application web, nous avons utilisé le site www.app.diagram.net.

II. Représentation des interfaces de l'application :

Page home :

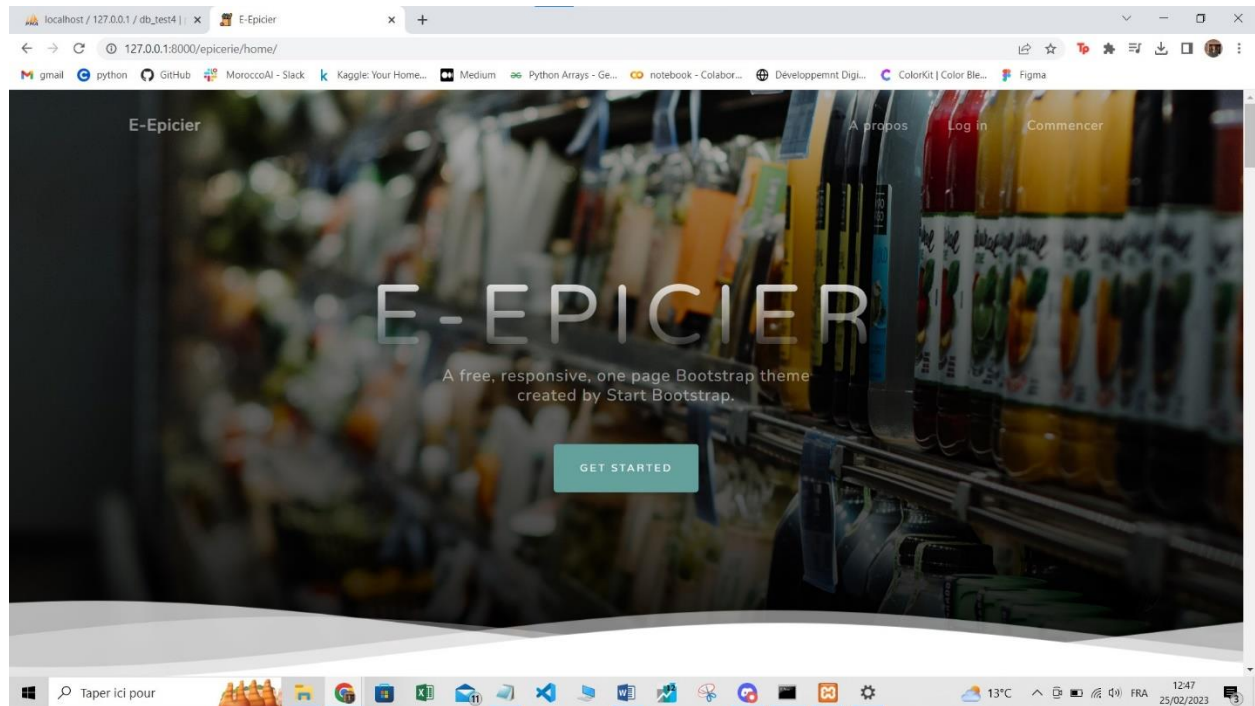


Figure 6 : Home page

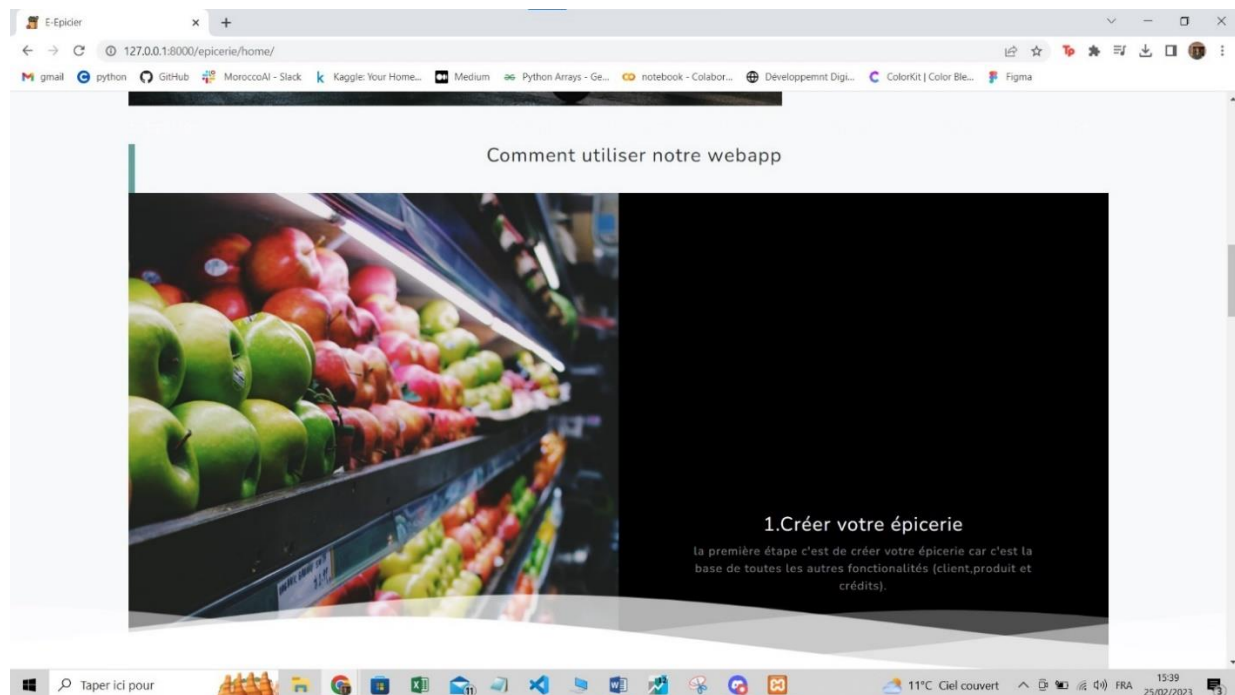


Figure 7 : guide d'utilisation

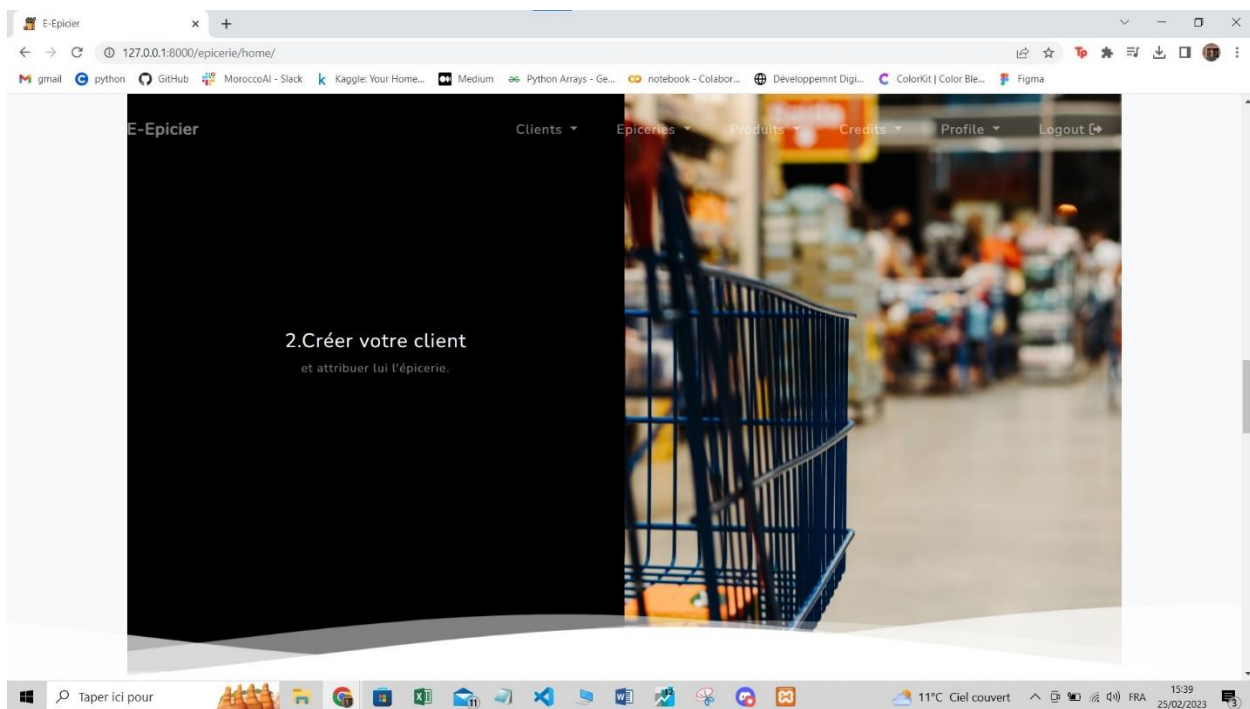


Figure 8 : guide d'utilisation

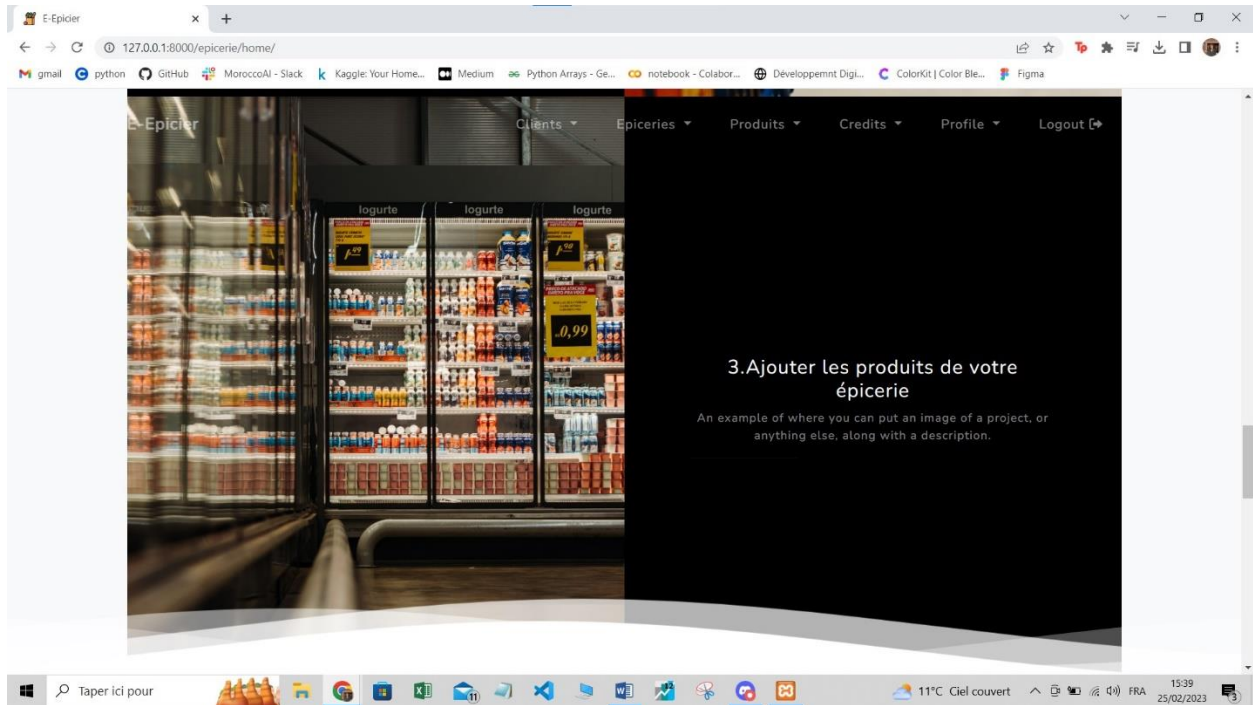


Figure 9 : guide d'utilisation

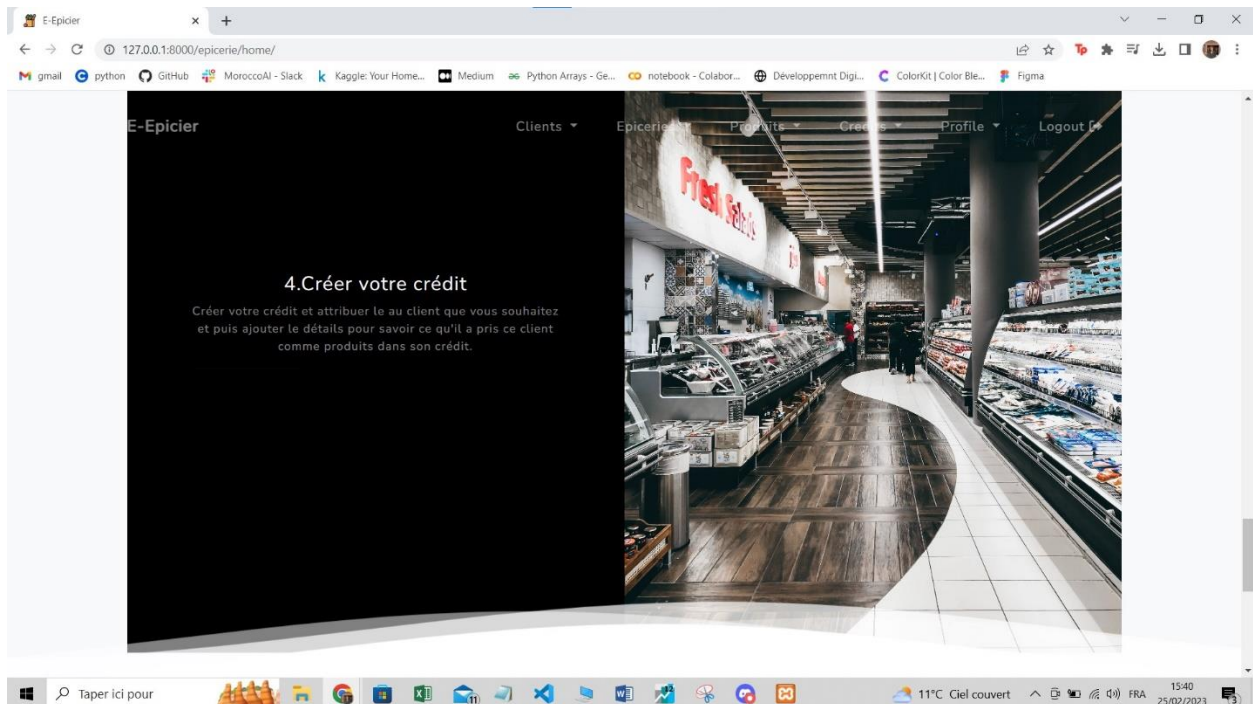


Figure 10 : guide d'utilisation

Interface d'épicier :

Cette interface présente le formulaire d'inscription de l'épicier dans notre application :

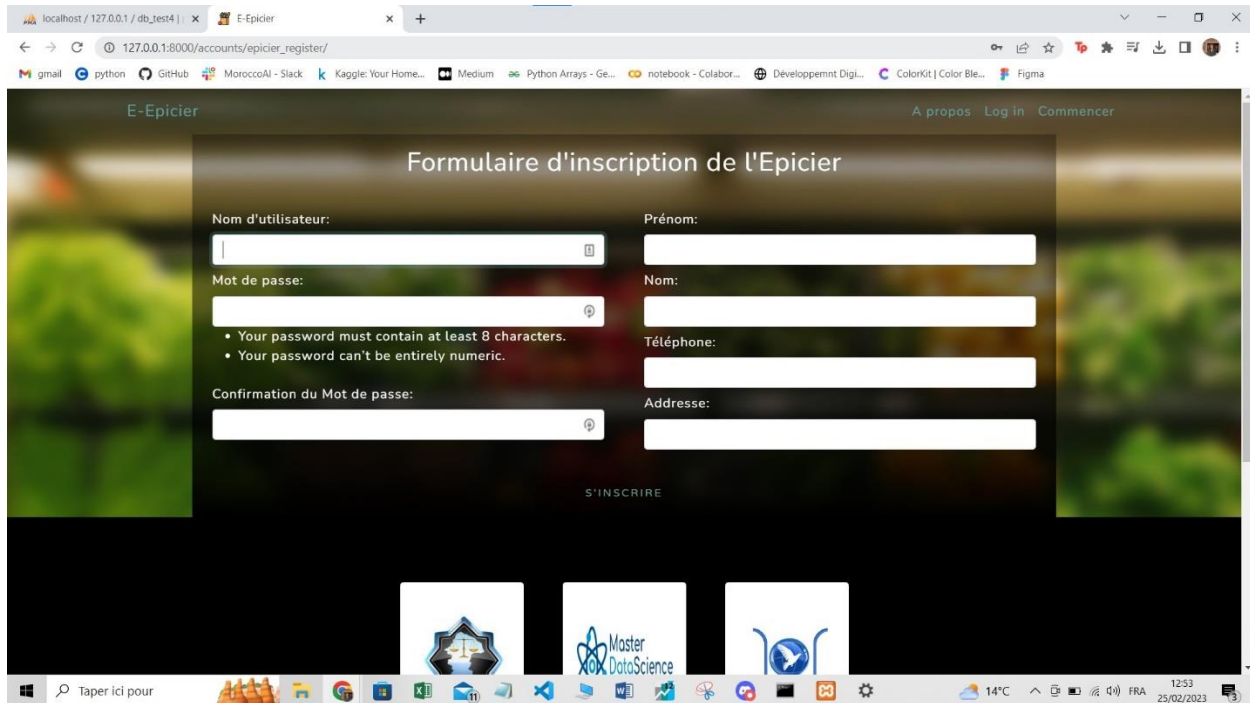
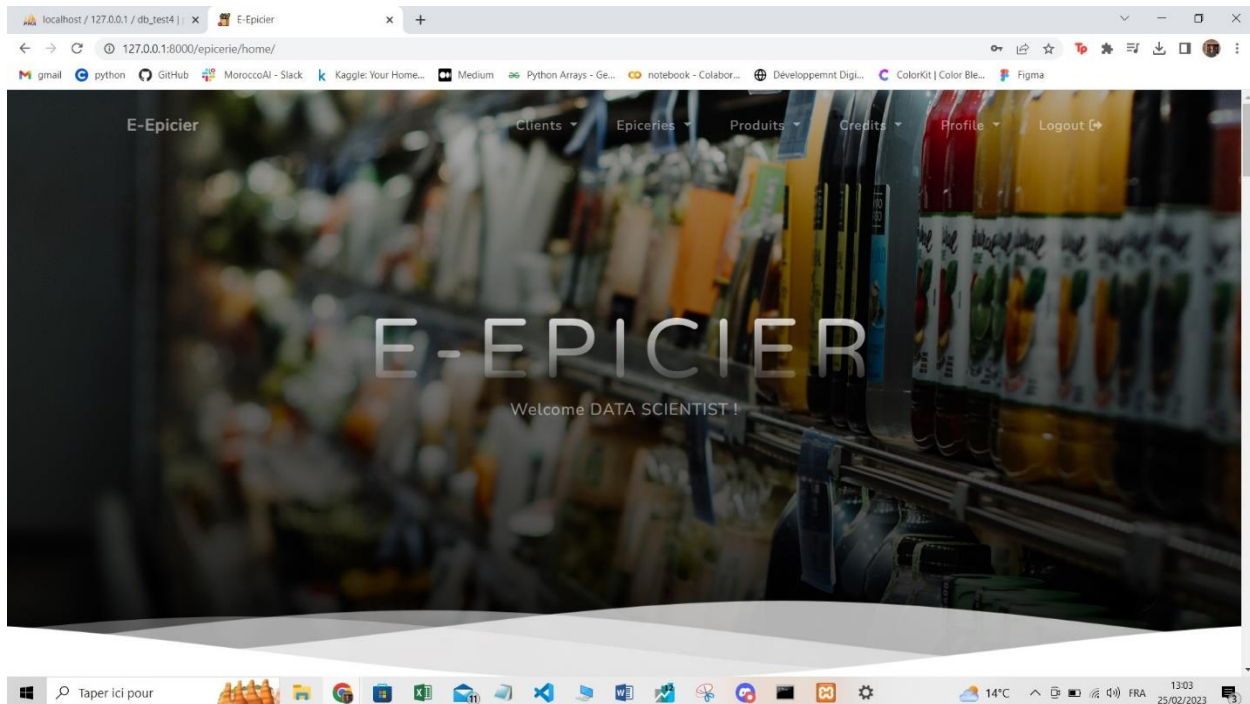


Figure 11 : formulaire d'inscription d'épicier

Après l'inscription, l'épicier aura la possibilité de créer ses épiceries, ses clients, ses produits et ses crédits ... il peut les modifier, les supprimer et les consulter à tout le temps.



Interface de gestion des épiciers :

Cette interface représente l'étape d'enregistrement d'une épicerie :

The screenshot displays a web browser window with the URL `127.0.0.1:8000/epicerie/create-epicerie/`. The page title is "E-Epicier". The navigation bar includes links for "Clients", "Epiceries", "Produits", "Credits", "Profile", and "Logout". The main content area is titled "Epicerie" and contains a registration form with the following fields:

- Nom_Epicerie:
- Adresse:
- Téléphone:

Below the form is a button labeled "ENREGISTRER". The footer features three logos: a scale of justice, "Master DataScience POUR L'ECONOMIE ET LA FINANCE", and a logo with a bird and waves. The Windows taskbar at the bottom shows the date and time as 16:02 on 25/02/2023.

Figure 12 : formulaire de créer une épicerie

Pour consulter l'ensemble des épiceries de cet épicier :

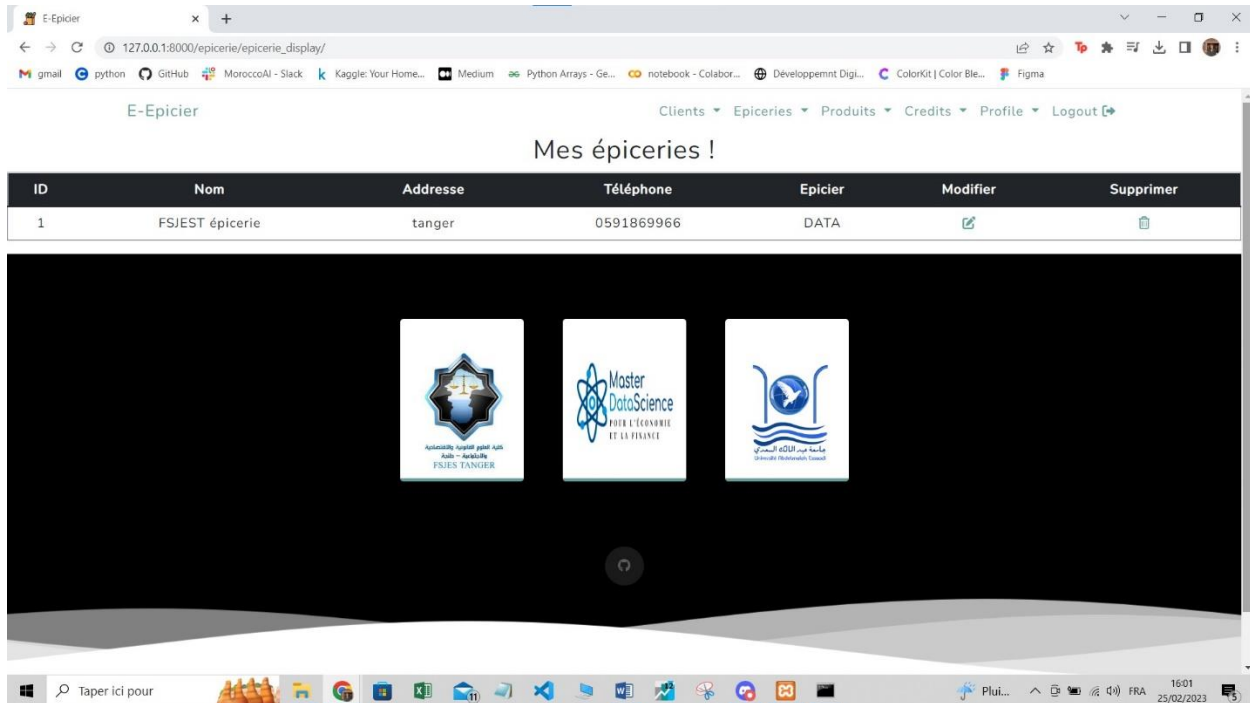


Figure 13 : consultation des épiceries

Interface de gestion des clients :

Formulaire d'inscription de client : Cette interface permet à l'épicier d'inscrire ses clients en demandant ces coordonnées :

The screenshot shows a web browser window with the URL 'localhost / 127.0.0.1 / db_test4 | : x +'. The page title is 'E-Epicier'. The main heading is 'Formulaire d'enregistrement d'un client'. The form contains the following fields:

- Nom d'utilisateur:
- Mot de passe:
 • Your password must contain at least 8 characters.
 • Your password can't be entirely numeric.
- Confirmation du Mot de passe:
- Epicier:
- Prénom:
- Nom:
- Téléphone:
- Adresse:

At the bottom of the form is a button labeled 'ENREGISTRER'.

Figure 14 : formulaire d'enregistrement d'un client

L'interface suivante permet à l'épicier de modifier, supprimer et consulter ses clients :

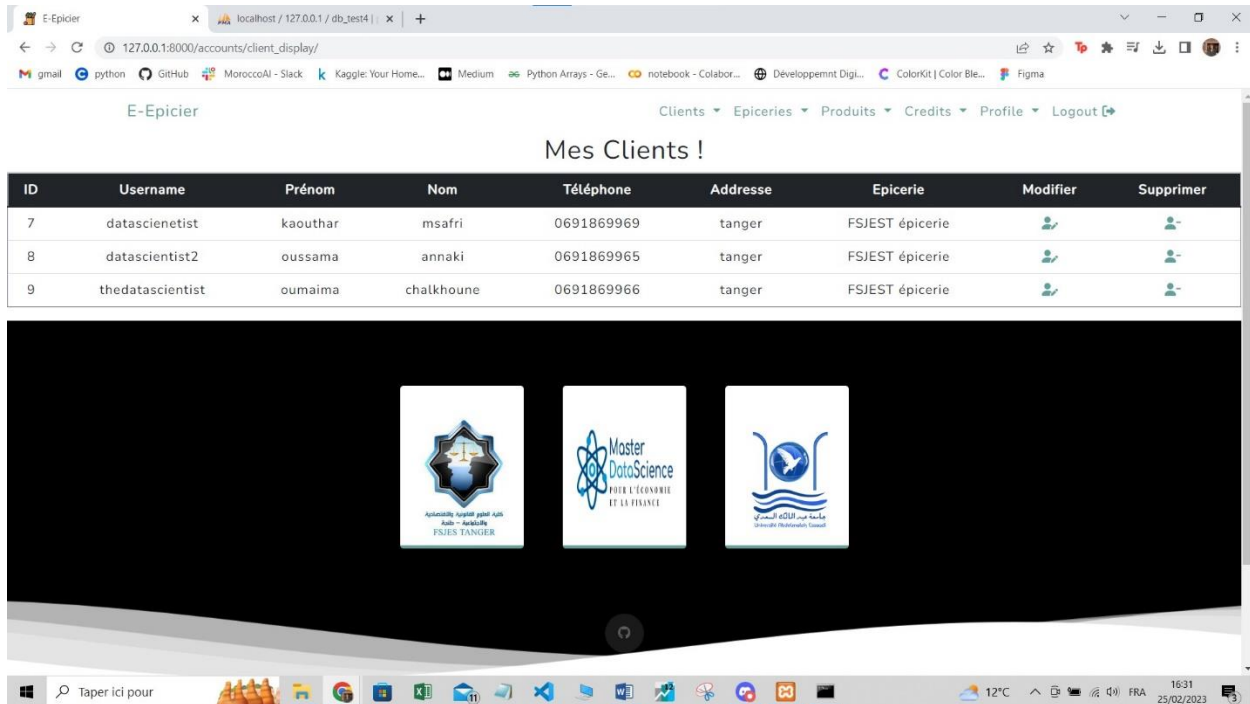


Figure 15: consultation des clients

Interface de gestion des produits :

Pour créer un nouveau produit :

Produit

Description:

Prix:

Disponibilité: ☐

Nom_Epicerie:

ENREGISTRER

Figure 16 : formulaire d'ajouter un produit

Pour consulter, modifier ou supprimer un produit :

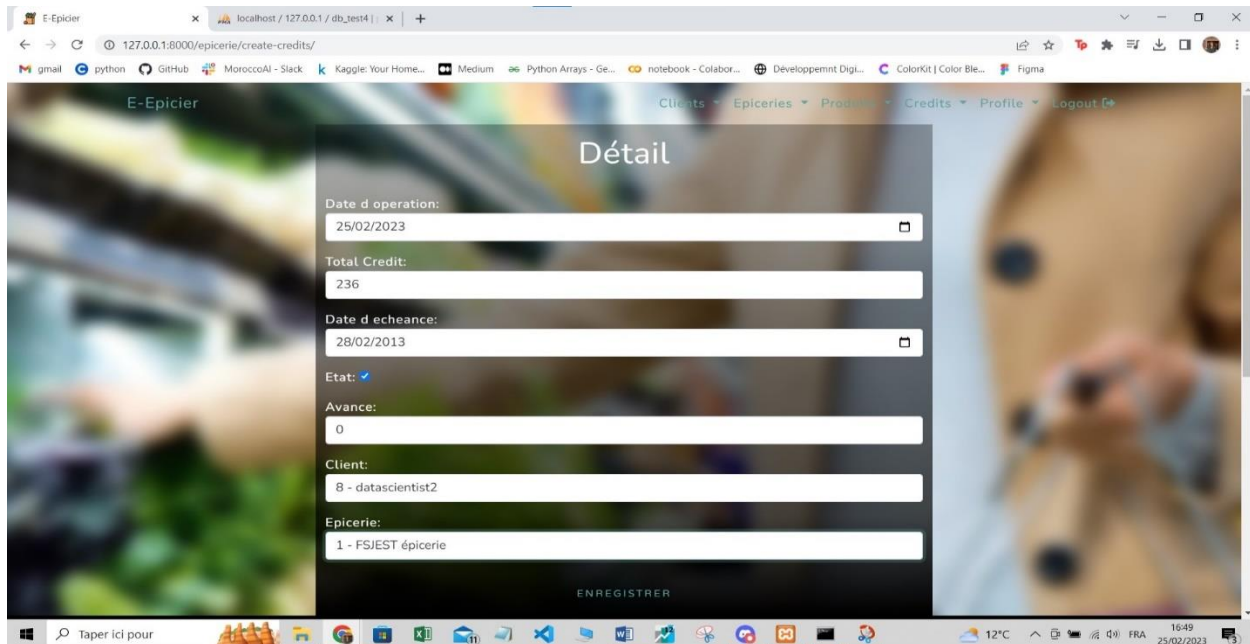
Mes Produits !

ID	Description	Prix	Disponibilité	Epicerie	Modifier	Supprimer
1	Farine	10.0	Disponible	FSJEST épicerie		
2	Purée de Fruits	15.0	Disponible	FSJEST épicerie		
3	Fruits Secs	20.0	Disponible	FSJEST épicerie		

Figure 17 : consultation des produits

Interface de gestion des crédits :

Cette interface permet à l'épicier d'inscrire ses crédits :



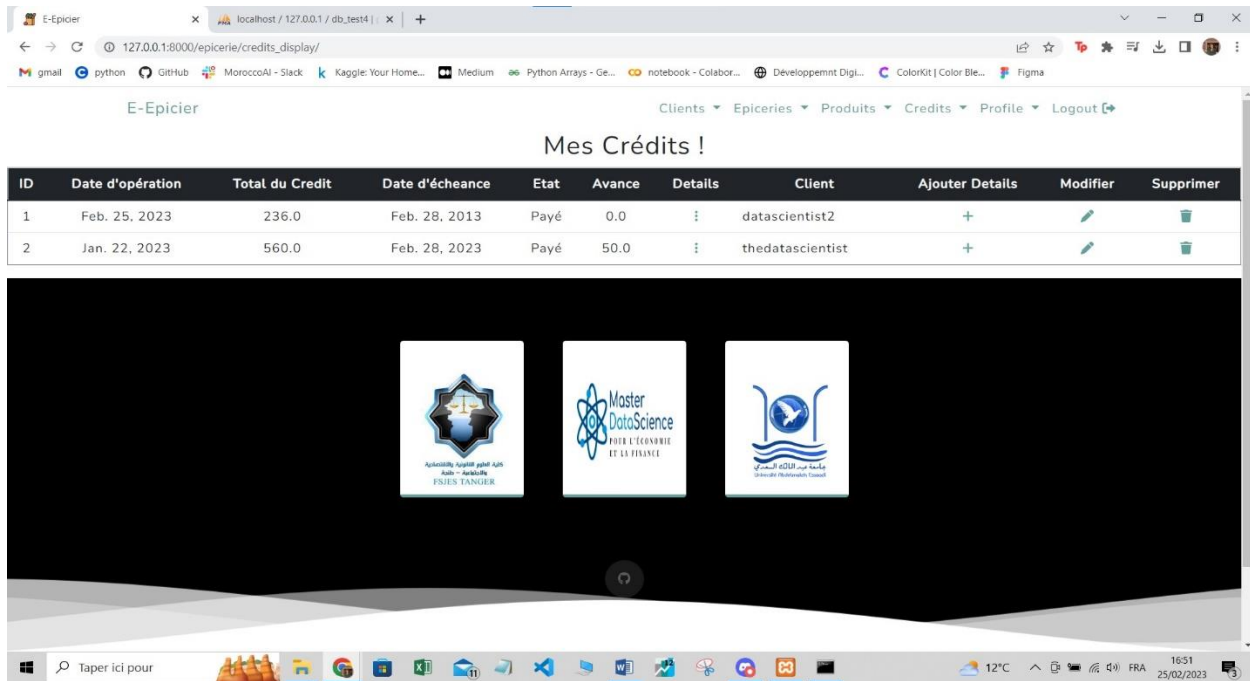
The screenshot shows the 'E-Epicier' web application interface. The main heading is 'Detail'. The form contains the following fields:

- Date d'operation: 25/02/2023
- Total Credit: 236
- Date d'echéance: 28/02/2013
- Etat: ☒
- Avance: 0
- Client: 8 - datascientist2
- Epicerie: 1 - FSJEST épicerie

At the bottom of the form is a button labeled 'ENREGISTRER'.

Figure 18 : formulaire de création d'un nouveau crédit

Et pour consulter, modifier ou supprimer les crédits :



The screenshot shows the 'E-Epicier' web application interface. The main heading is 'Mes Crédits !'. Below the heading is a table with the following columns: ID, Date d'opération, Total du Credit, Date d'échéance, Etat, Avance, Details, Client, Ajouter Details, Modifier, and Supprimer.

ID	Date d'opération	Total du Credit	Date d'échéance	Etat	Avance	Details	Client	Ajouter Details	Modifier	Supprimer
1	Feb. 25, 2023	236.0	Feb. 28, 2013	Payé	0.0	:	datascientist2	+		
2	Jan. 22, 2023	560.0	Feb. 28, 2023	Payé	50.0	:	thedata scientist	+		

Below the table are three logos:

- FSJEST TANGER
- Master DataScience
- FSJEST TANGER

Figure 19 : consultation des crédits

Interface client :

Pour que le client peut s'authentifier, il visite l'interface de login, et il insert l'ID et le mot de passe fournis par son épicerier.

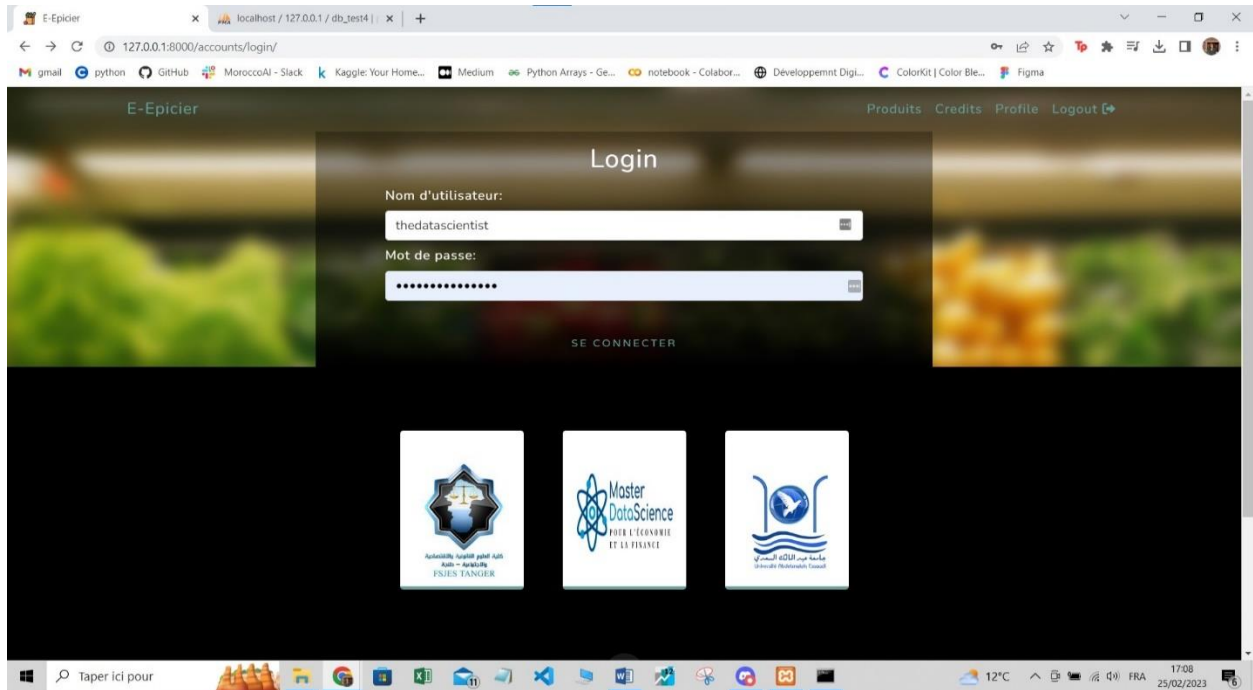


Figure 20: authentification du client

Ce dernier peut consulter les produits disponibles et non disponibles chez son épicerie, comme l'interface suivant montre :

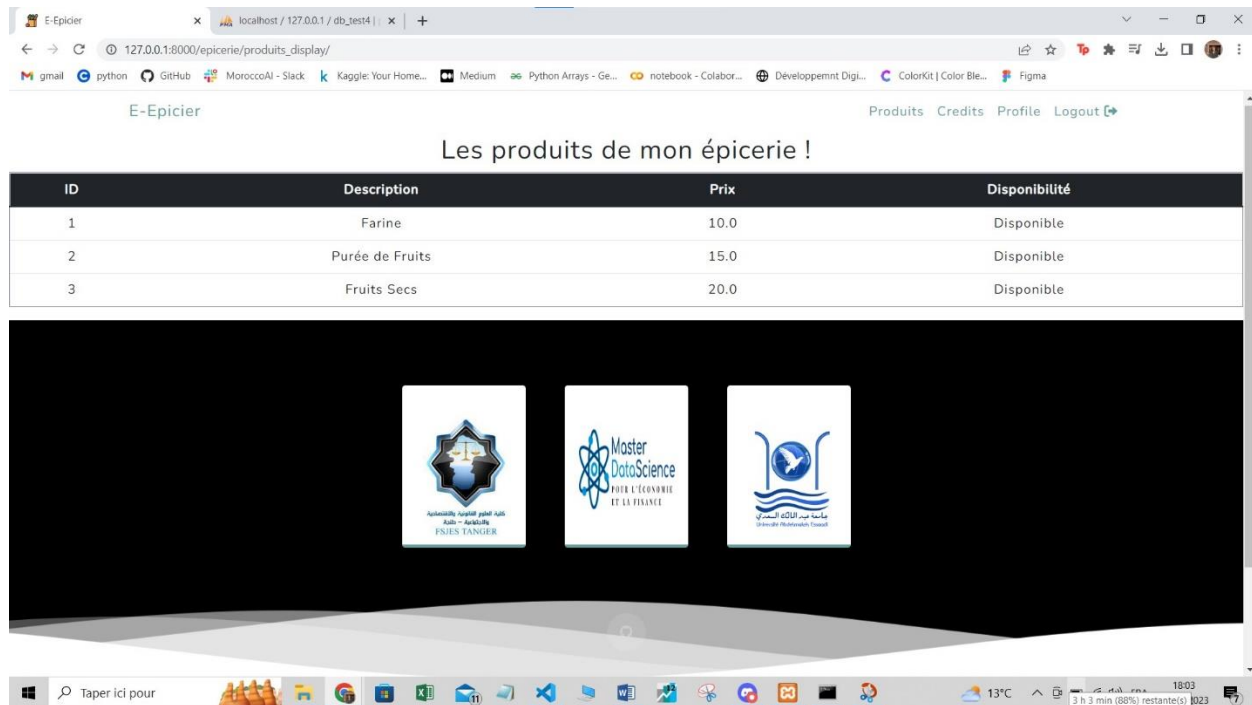
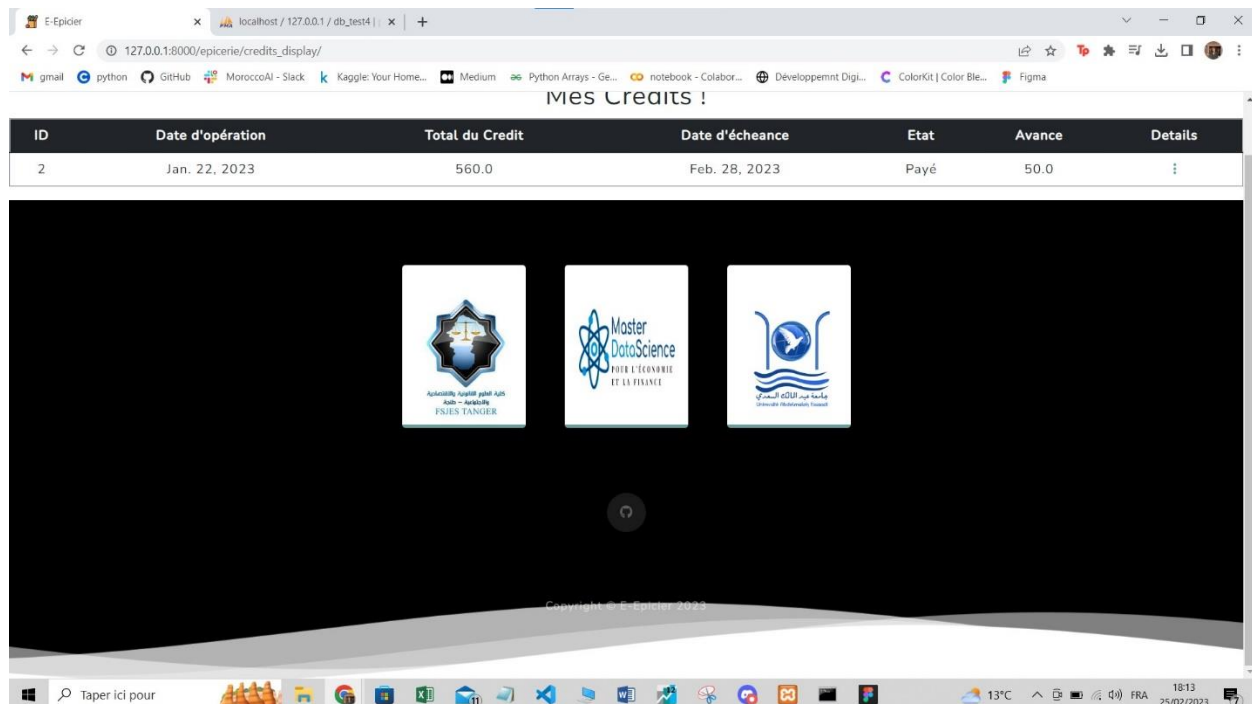


Figure 21: consultation des produits par le client

Ainsi, le client peut voir ses crédits à tout moment en visitant l'interface des crédits, il peut également connaître la date d'échéance et l'état de crédit s'il est payé ou pas encore :



Conclusion :

Au cours de ce chapitre dédié à l'implémentation, nous sommes passés de la théorie à la pratique, elle nous a permis d'atteindre nos objectifs décrits dans les étapes précédentes.

Nous avons présenté les outils qui nous ont aidé à concevoir notre application, le langage de programmation et le framework utilisé les interfaces pour bien illustrer le travail qui a été fait.

Conclusion générale

Au cours de ce travail, nous avons présenté les différentes étapes ayant conduit à la mise en œuvre d'une application web dédiée à la gestion des crédits dans un épicerie

Nous avons commencé par recenser les difficultés que rencontre l'épicier afin d'apporter la solution adéquate et spécifier ainsi les besoins.

Le langage de modélisation UML et le processus unifié UP ont constitué le support de l'analyse des besoins et la conception de notre application web via les différents diagrammes UML couvrant les aspects fonctionnels et statiques de tout le développement.

Pour enfin réaliser l'application, nous avons utilisé ' MYSQL ' pour implémenter la base de données et le langage Python via le framework Django qui est basé sur ORM pour les interfaces, et le Bootstrap pour le design.

Ce projet a fait l'objet d'une expérience intéressante, très bénéfique pour nous. En effet, il nous a permis d'enrichir nos connaissances théoriques et compétences dans le domaine de la conception et de la programmation. Ajoutant à ceci, la mise en application des connaissances acquises tout au long de nos études.

En plus, c'était une bonne occasion pour réaliser un travail concret avec des objectifs clairs et bien définis. Ce projet nous a également permis de nous familiariser avec l'environnement de travail et de la vie professionnelle.

Cependant des perspectives restent envisageables, telles que l'enrichissement de l'application par la fonctionnalité permettant d'établir des statistiques ainsi que l'amélioration de la qualité des renseignements via une application Android et iOS qui viendra compléter l'application web, la possibilité de payer à distance.

Documentations :

- <https://www.youtube.com/watch?v=eDwQxJ9zvco&list=PLh-rUZWaw76Grm5uRdntEhupKs-8pC4My&index=1>
- <https://www.w3schools.com/django/>
- <https://dev.to/sm0ke/how-to-use-mysql-with-django-for-beginners-2ni0>