

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE MATHÉMATIQUES ET DE SCIENCES PHYSIQUES

---

TRAVAUX PRATIQUES BIG DATA

DÉVELOPPEMENT D'UNE PLATEFORME DE TROC D'OBJETS

MASTER I - INFORMATIQUE



ENSEIGNANTS:

BONDIOMBOUY CARLYNA  
VALDURIEZ PATRICK

ÉTUDIANTS:

GNOGUEM NESTOR  
HOUNGUE ISAAC  
SAIBOU AZIZ

MARS 2019

# TABLE DES MATIÈRES

I	INTRODUCTION GÉNÉRALE . . . . .	3
I.1	CONTEXTE . . . . .	3
I.2	PROBLÉMATIQUE . . . . .	3
I.3	OBJECTIFS . . . . .	3
I.4	PLAN DU RAPPORT . . . . .	3
II	DESCRIPTION DU PROJET . . . . .	4
III	ANALYSE ET CONCEPTION . . . . .	5
III.1	IDENTIFICATION DES ACTEURS . . . . .	5
III.2	LES BESIONS FONCTIONNELS DU SYSTÈME . . . . .	5
III.3	MODÈLE DYNAMIQUE . . . . .	6
IV	RÉALISATION . . . . .	9
IV.1	DATASet vs BESOIN EN DONNÉES . . . . .	9
IV.2	ENVIRONNEMENT DE DÉVELOPPEMENT LOGICIEL . . . . .	9
IV.3	QUELQUES INTERFACES DE NOTRE APPLICATION . . . . .	11
IV.4	DÉMONSTRATION . . . . .	12
V	CONCLUSION . . . . .	13

# I INTRODUCTION GÉNÉRALE

---

## I.1 CONTEXTE

Le procédé qui consiste en l'abandon d'une chose à quelqu'un dans le but d'avoir quelque chose en contrepartie est un échange dit **troc**. En plus de créer des relations ou tout simplement d'établir un contact humain solidaire, le **troc** permet à certains d'obtenir des objets ou de réaliser des projets qu'ils n'auraient pas pu acquérir sans cet échange de biens ou de savoir-faire. En outre, les objets que d'autres désirent jeter sont ainsi récupérés par un tiers, permettant la réduction de déchets et de donner à ceux qui en ont la nécessité. Les objets s'offrent une nouvelle vie au lieu d'être jetés inutilement et le pire, c'est que tout le monde est gagnant sans avoir déboursé un seul euro. Un bon moyen de répartir le sourire aux lèvres. Par contre la difficulté de connaître de trouver des personnes avec qui troquer constitue la première limite de ce système d'échange. La mise en place d'une plateforme de **troc** s'affranchira de la limite citée ci haut et permettra un échange lucide renforçant les liens sociaux et réduisant le nombre d'objets qui prématurément, se retrouvent dans les décharges.

## I.2 PROBLÉMATIQUE

Les système de **troc** existant posent de grands problèmes de confiance et ne sont ainsi confrontés qu'à la méfiance de la part des utilisateurs. Il faudra ainsi résoudre ce problème de confiance en implémentant dans notre système les principes de l'interaction homme-machine tels la robustesse, le guidage... La réussite de ce projet permettra aux populations de réduire la conservation des objets non utilisés ainsi que l'évacuation dans les décharges des objets qui auraient pu être utiles à d'autres personnes.

## I.3 OBJECTIFS

Notre projet permettra à toute personne de trouver des gens avec qui troquer librement des objets. Nous devons dans ce projet rendre compte de la compréhension du **DataSet**, **BesoinEnData**. Toutefois, il n'est pas évident de reconnaître la nécessité ou non de l'utilisation des technologies du **BigData** dans les projets informatiques. On doit ainsi rendre compte de la maîtrise des outils soit des concepts suivants et justifier nos choix des solutions (outils) que nous avons utilisés: **HDFS**, **Spark**, **Neo4J**, **MongoDB**, **DataSet**.

## I.4 PLAN DU RAPPORT

Partant de l'analyse et à la réalisation, on exposera toutes les différentes étapes nécessaires à la construction de notre programme. Il suivra donc une conclusion résumant tout notre travail.

## II DESCRIPTION DU PROJET

---

Le projet consiste en le developpement d'une plateforme pour les personnes intéressées par l'échange d'objets (**troc**). Notre système analysera les conditions de d'approbation ou de validation du **troc** entre deux ou plusieurs personnes. Les deux parties impliquées dans l'échange (les utilisateurs voulant échanger) doivent s'abonner à la plateforme avant de pouvoir utiliser notre service. Tout abonné peut ajouter des objets dans le but de l'échanger avec des objets qu'auront proposé d'autres abonnés. Un abonné serait donc en mesure de suggérer des objets contre ceux ou celui d'autres utilisateurs, qui les plaisent. A travers notre plateforme les abonnés relèverons leurs besoins sans affaiblir leur position de négociation. Cet avantage est important si l'abonné se trouve face à un monopole discriminant pour un bien très désiré.

Nous pourrions tirer profit de ce projet en mettant en place des abonnements payants, des publicités ou des frais supplémentaires pour ceux qui veulent que leur objet soit bien placé sur la plateforme parmi les tops.

### III ANALYSE ET CONCEPTION

---

L'analyse des besoins est une phase qui consiste à comprendre et à déterminer les différentes fonctionnalités et besoins du système.

#### III.1 IDENTIFICATION DES ACTEURS

L'identification des acteurs d'une application est une phase primordial de l'analyse des besoins, c'est ainsi qu'on a pu identifier les acteurs suivants:

**CLIENT:** Toute personne physique capable de poster ou de consulter un objet.

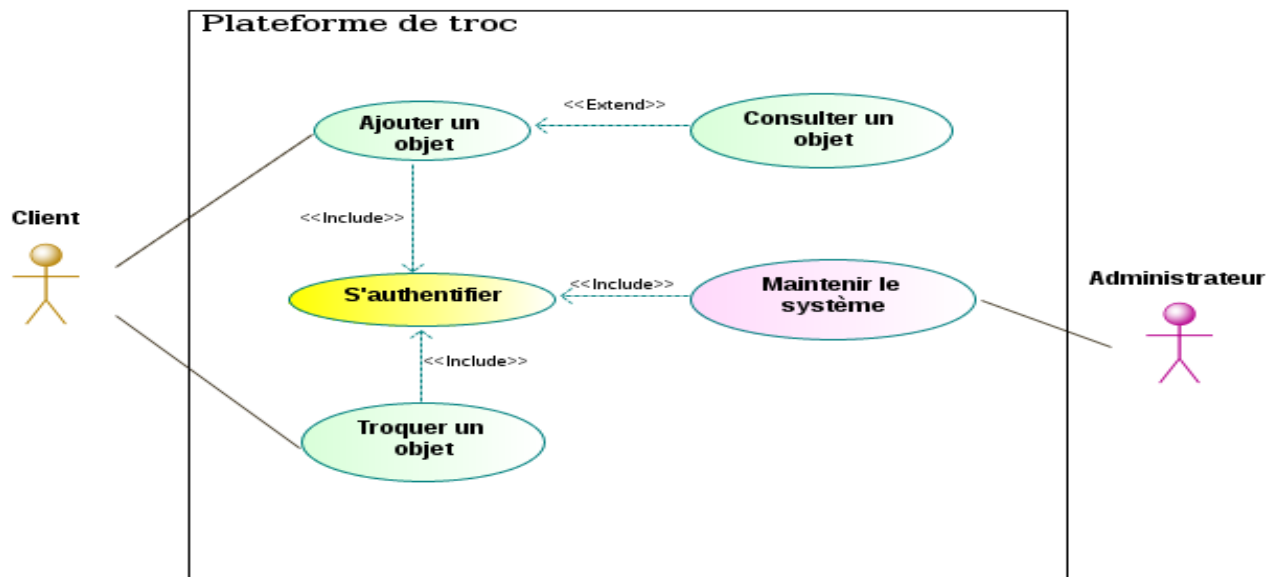
**ADMINISTRATEUR:** Toute personne susceptible de maintenir le système du point de vue gestion des compte des utilisateurs.

#### III.2 LES BESIIONS FONCTIONNELLS DU SYSTÈME

Les besoins fonctionnels constituent les fonctionnalités que le système devra assurer, parmi les quelles, nous décrivons principalement les fonctions suivantes:

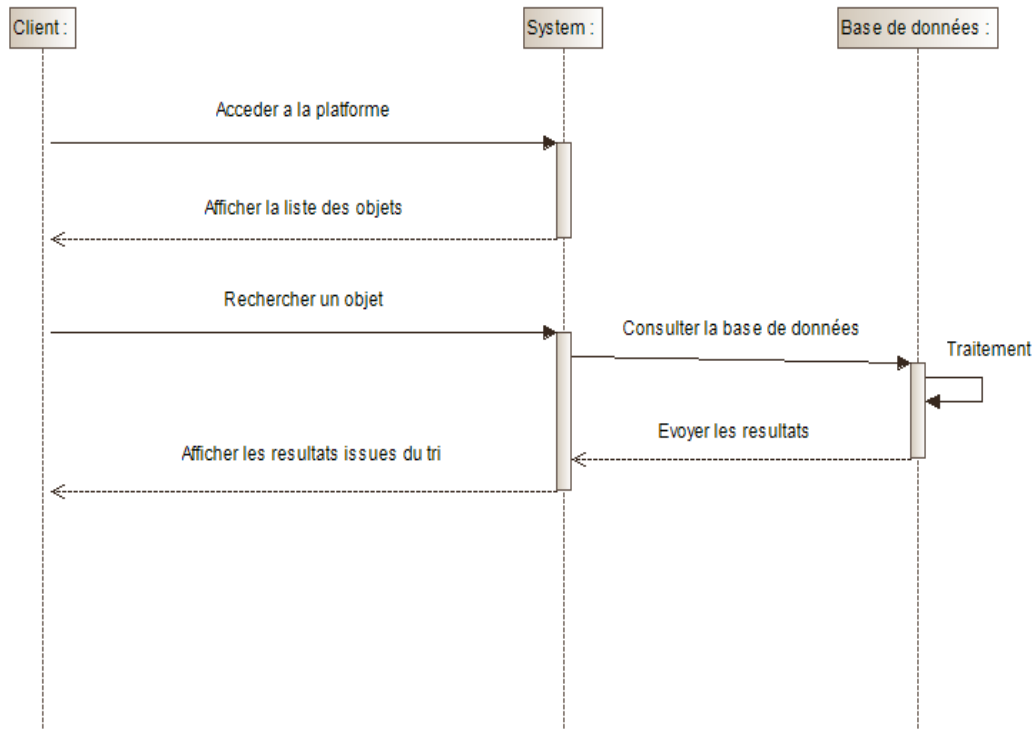
1. Consulter des objets
2. Troquer un objet
3. Ajouter un objet

Le diagramme des cas d'utilisation suivant rend compte des fonctionnalités de notre application

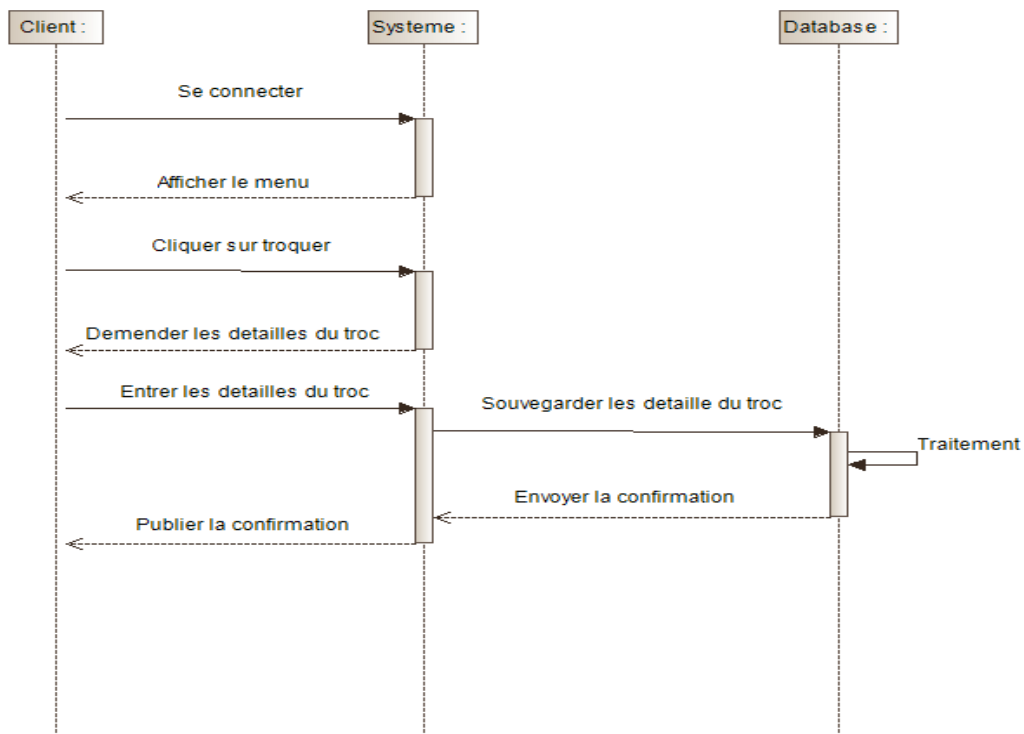


### III.3 MODÈLE DYNAMIQUE

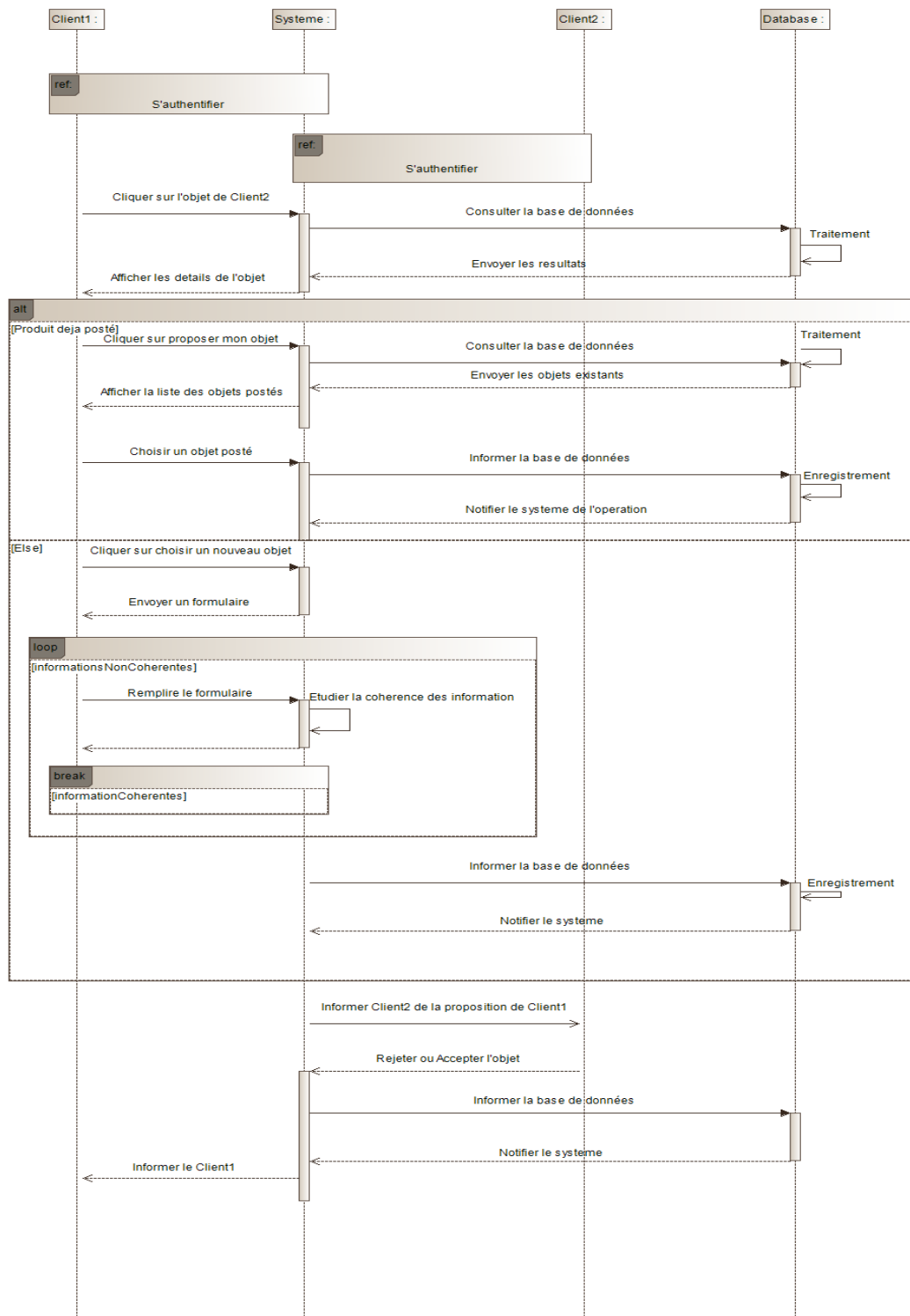
Après la description des différents cas d'utilisation, nous allons élaborer le modèle dynamique dans lequel nous allons décrire les scénarios de quelques cas d'utilisations les plus importants sous forme de diagrammes de séquence.



La figure précédente est celui du diagramme de séquence est du cas "Rechercher un objet"

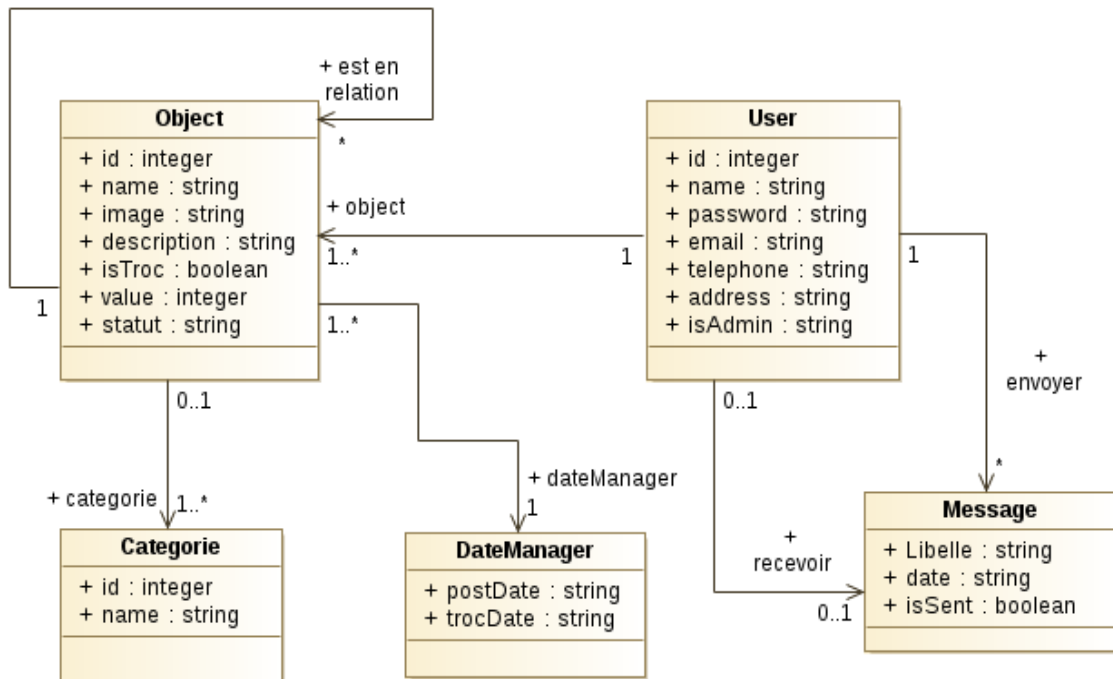


La figure précédente est celui du diagramme de séquence est du cas "Ajouter un objet"



La figure précédente est celui du diagramme de séquence est du cas "Troquer un objet"

Le diagramme de classe dictant le comportement de notre programme est le suivant:





## IV RÉALISATION

---

### IV.1 DATASET VS BESOIN EN DONNÉES

Vu que le caractère des objets à échanger n'est pas prédéfini, on doit s'attendre à une grande variété presque imprevisible de données de la part des clients. Pensant la plateforme à une échelle mondiale, il y aura tout de même la vitesse de génération des données qui deviendra un facteur déterminant dans le choix des solutions le long de la phase de développement de notre plateforme.

Non seulement notre système implémente les trois premiers *V* de la théorie du **BigData** (*volume, velocity veracity*), mais aussi, le concept de valorisation des données en **BigData** (utilisation des données pour en extraire de la valeur) à des fins publicitaires par exemple, nous motive à opter pour une solution NoSQL en ce qui concerne le stockage et la gestion de nos données.

#### CHOIX DU SGBD NoSQL

MongoDB est un sgbid orienté document. Il repose sur le principe du clé/valeur, mais avec une extension sur les champs qui composent ce document. Il propose ainsi un langage d'interrogation riche permettant de faire des manipulations complexes sur chaque attribut du document (et sous-documents) comme dans une base de données traditionnelles, tout en passant à l'échelle dans un contexte distribué.

Considérons le sgbid **Spark**, cette solution est moins appropriée pour la lecture de données spécifiques comme pour les clés/valeurs, ce qui ne nous motive pas à l'utiliser.

De la même façon, **Neo4j** est une solution utilisée dans les systèmes où la gestion de chemins, de propagations est une priorité.

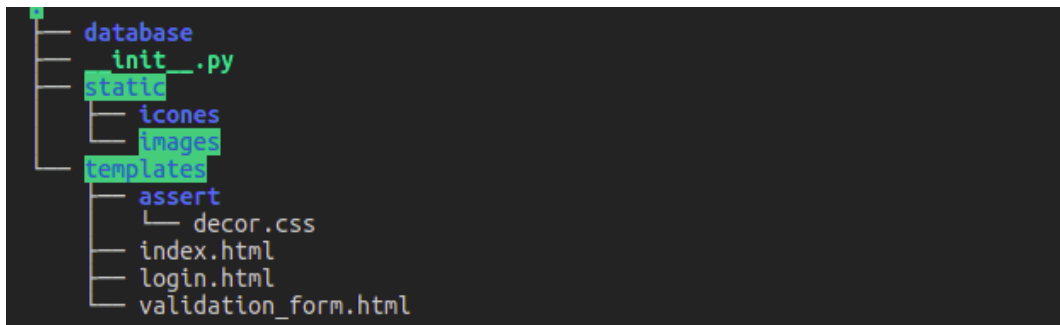
On a ainsi choisi MongoDB.

#### PRÉSENTATION DU DATASET

Pour mieux appréhender le comportement de notre système il est très utile que les données de test (**DataSet**) lors de la phase de développement soient considérables et cohérentes avant d'entrer en production. Nous avons donc utilisé le **DataSet** inspiré de notre diagramme de classe et contenu dans le fichier `dataset.json` annexé à notre rapport.

### IV.2 ENVIRONNEMENT DE DÉVELOPPEMENT LOGICIEL

En vue d'écrire un programme cohérent très tôt compréhensible par nous-même et éventuellement par d'autres développeurs en vue d'une maintenance très aisée, nous avons adopté la structuration suivante pour nos fichiers.



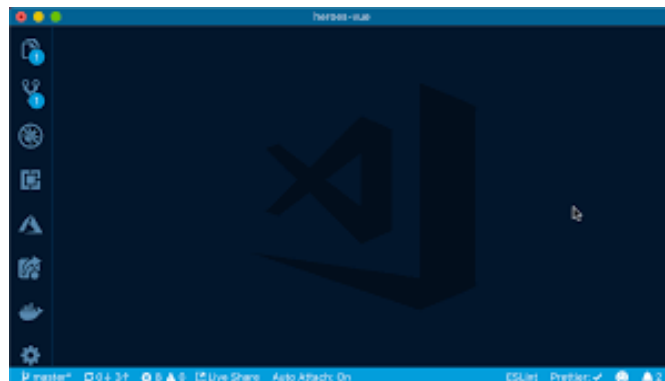
1. Le dossier `static` contient les données static de notre application
2. Le dossier `templates` contient les templates de notre application
3. Le dossier `assert` contient tous les fichiers de styles (`.css`) de notre application.
4. Le dossier `images` sous le dossier `static` contient toutes les images des objets de notre application.
5. Le dossier `icones` sous le dossier `static`

LES OUTILS LOGICIELS QUE NOUS AVONS UTILISÉ SONT LES SUIVANTS

MODELIO 3.7: Un logiciel de modélisation open-source



VISUAL-STUDIO-CODE 1.28: Un éditeur de texte avancé



FLASK 1.0: Un *mini-framework* open-source de développement web en Python



JINJA2: Un moteur de template



MONGO-DB: Un système de gestion de base de données orienté document open-source développé en C++

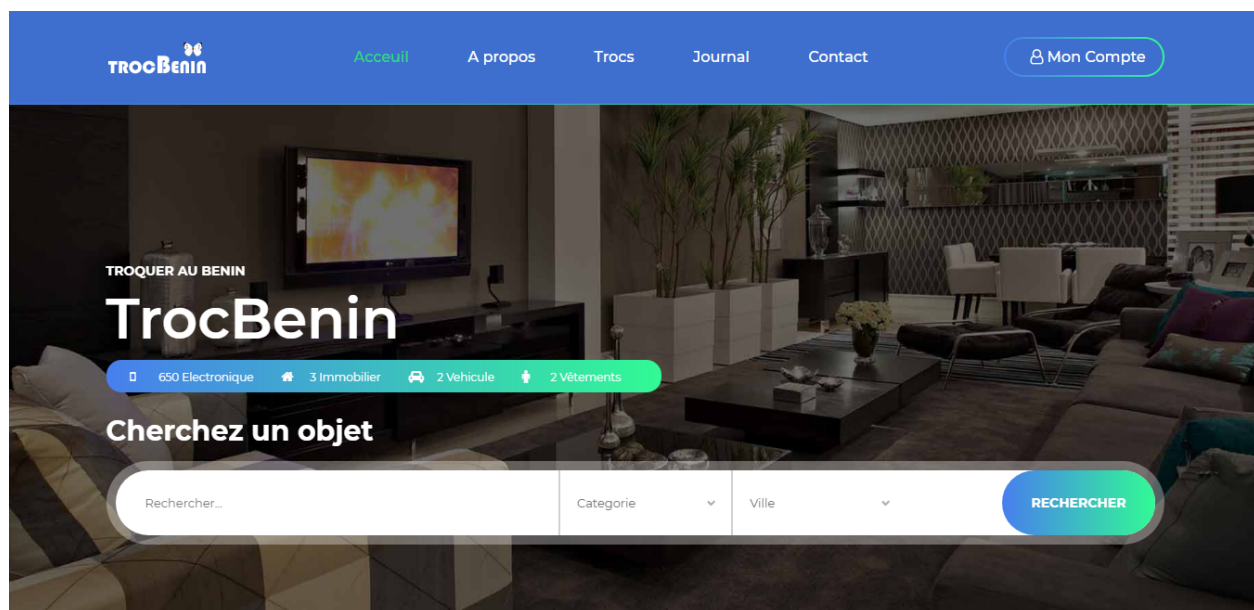


**MOCKAROO:** Une plateforme de génération de données. Nous l'avons utilisé pour pour générer notre DataSet



### IV.3 QUELQUES INTERFACES DE NOTRE APPLICATION

Voici quelques interfaces de notre application:



L'interface précédente est la page d'accueil de l'application. Le client a possibilité de rechercher un objet.

L'interface précédente est la page permettant aux utilisateurs d'ajouter un objet pour un troc

## IV.4 DÉMONSTRATION

## V CONCLUSION

---

Dans la première partie de ce travail, nous avons introduit une plateforme permettant à ses utilisateurs d'échanger des objets. Cette plateforme sera utilisée au Bénin. Dans la deuxième partie, nous présentons avec plus de précision le contexte du projet. Après avoir examiné le contexte, nous avons modélisé le projet à l'aide de diagrammes UML dans la troisième section. Nous avons présenté quatre cas d'utilisation. le client utilise le système pour ajouter un objet, échanger un objet et rechercher un objet offert en échange. L'administrateur mentien le systeme. Nous avons présenté la séquence des interactions impliquées dans ces quatre cas d'utilisation ainsi qu'un diagramme de classes montrant les différentes entités sous forme de classes impliquées dans notre système. Dans la quatrième section, nous avons présenté notre choix de technologies de mise en œuvre et notre data set. Nous avons choisi MongoDB sur Spark et Neo4J parce qu'il propose un langage de requête riche permettant d'effectuer des manipulations complexes sur chaque attribut du document. Nous avons ensuite présenté quelques captures d'écran de la hiérarchie des fichiers de notre projet et quelques captures d'interface.