

OC Inventory

Solution d'inventaire de parc informatique

Dossier de conception fonctionnelle

Version 1.0

Auteur

David Bouzerar

Développeur

IT Consulting & Development



IT Consulting & Development

**IT Consulting &
Development**
www.it-dev.ovh

10, rue d'Amiens 80000 Amiens – 03 23 45 67 89 – contact@it-dev.ovh

S.A.R.L. au capital de 1 000,00 € enregistrée au RCS d'Amiens – SIREN 999 999 999 – Code APE : 6202A

TABLE DES MATIERES

1 - Versions	3
2 - Introduction	4
2.1 - Objet du document.....	4
2.2 - Références.....	4
2.3 - Besoin du client	4
2.3.1 - Contexte	4
2.3.2 - Diagramme de contexte	5
2.3.3 - Enjeux et Objectifs	6
3 - Description générale de la solution	8
3.1 - Les principes de fonctionnement.....	8
3.2 - Les acteurs	8
3.3 - Les cas d'utilisation	9
3.3.1 - Cas d'utilisation général :.....	10
3.3.1.1 - Description du diagramme	11
3.3.2 - Cas d'utilisation « Authentification » :	12
3.3.2.1 - Description du diagramme	12
3.3.3 - Cas d'utilisation « Gestion de l'inventaire ».....	13
3.3.3.1 - Description du diagramme	14
3.3.4 - Cas d'utilisation « Consultation de l'inventaire ».....	15
3.3.4.1 - Description du diagramme	15
4 - Le domaine fonctionnel	17
4.1 - Packages.....	17
4.1.1.1 - Description du diagramme	17
5 - Application « OC Inventory »	19
6 - Glossaire	21

1 - VERSIONS

Auteur	Date	Description	Version
David Bouzerar	09/10/2022	Création du document	0.1
David Bouzerar	12/10/2022	Introduction	0.2
David Bouzerar	15/10/2022	Description générale de la solution	0.3
David Bouzerar	17/10/2022	Usecases	0.4
David Bouzerar	18/10/2022	Domaine fonctionnel	0.5
David Bouzerar	22/10/2022	Application	0.6
David Bouzerar	23/10/2022	Glossaire	0.7
David Bouzerar	06/04/2022	Mise à jour du document	0.8
David Bouzerar	09/04/2022	Finalisation du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception fonctionnelle de l'application d'inventaire Utilisateurs / Périphériques. Cette solution est destinée à améliorer l'inventaire et le suivi du ou des matériel(s) assigné(s) à un utilisateur au sein d'une université.

L'objectif de ce document est de présenter la solution fonctionnelle proposée pour répondre aux besoins exprimés.

2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants :

1. **DCT-02** : Dossier de conception technique de l'application
2. **DEXP-03** : Dossier d'exploitation de l'application

2.3 - Besoin du client

2.3.1 - Contexte

L'université, qui exprime le besoin de disposer de cette application, se situe dans les Hauts de France et dispose de plusieurs antennes réparties dans les trois départements de la région picarde. Pour information, environ 33 000 étudiants y sont inscrits et 2500 employés y sont affectés dont 1300 enseignants-chercheurs et 1200 employés administratifs et techniques.

A la demande de la **DSI**, il est nécessaire de mettre en place une solution d'inventaire Utilisateur/Périphérique concernant le matériel informatique. Celle-ci doit permettre notamment :

- D'augmenter l'efficacité du suivi du ou des matériels assignés aux utilisateurs
- De disposer d'informations concernant les différents périphériques
- Pouvoir identifier un utilisateur (**nom, prénom, affectation...**)
- De suivre le stock des périphériques disponibles

- De suivre l'état des différents périphériques du parc informatique :
- De proposer une interface aux utilisateurs authentifiés (**administrateurs, responsables, techniciens, administratifs**)
- D'alerter au sujet des anomalies éventuellement détectées
- De disposer d'un « **Dashboard** » contenant quelques informations (statistiques) sur l'état du parc informatique

2.3.2 - Diagramme de contexte

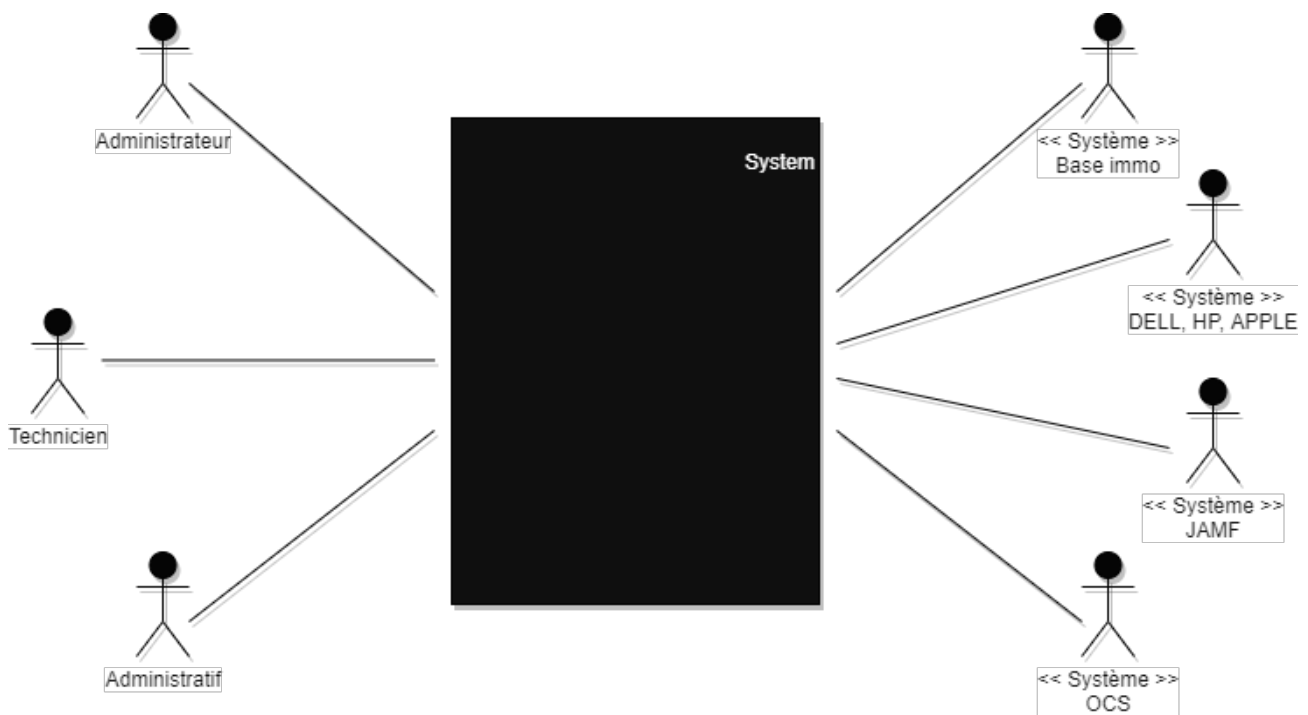


Diagramme de contexte

2.3.3 - Enjeux et Objectifs

L'application est, dans ce diagramme, représentée par le rectangle noir (**System**).

Les acteurs principaux (à gauche du rectangle) représentent :

- **L'administrateur :**

Il s'agit du compte ayant tous les droits sur l'application, il peut :

- Créer, modifier, supprimer des utilisateurs authentifiés de l'application
- Créer, modifier, supprimer des utilisateurs
- Créer, modifier, supprimer des périphériques
- Importer les informations des bases « externes »
- Attribuer des droits aux utilisateurs authentifiés de l'application

- **Le technicien :**

- Créer, modifier, supprimer des utilisateurs
- Créer, modifier, supprimer des périphériques
- Importer les informations des bases « externes »

- **L'administratif :**

- Peut consulter les utilisateurs
- Peut consulter les périphériques

- **Le système « base immo » :**

- Le **système « base immo »**, en tant qu'acteur secondaire, sera sollicité lors de l'import d'informations liées à l'immobilisation des périphériques

- **Le système « Dell, HP, APPLE » :**

- Ce système, en tant qu'acteur secondaire, sera sollicité lors de l'import d'informations liées aux garanties des périphériques par le biais d'API

- **Le système « OCS » :**
 - Le **système « OCS »**, en tant qu'acteur secondaire, permettra de recueillir des informations liées aux caractéristiques des périphériques

3 - DESCRIPTION GENERALE DE LA SOLUTION

3.1 - Les principes de fonctionnement

La solution qui sera mise en place consiste à développer une **application WEB** selon les besoins exprimés précédemment. Cette application disposera d'une interface depuis laquelle, les utilisateurs authentifiés, pourront consulter les informations issues de la base de données.

D'autre part, l'**application WEB** contiendra également une partie « **Administration** » destinée à être utilisée par les administrateurs et techniciens. Ils pourront importer, modifier ou supprimer des informations selon les droits attribués.

3.2 - Les acteurs

- **Visiteur / Utilisateur :**

- I. **En tant que visiteur :**

Le **visiteur** n'a pas la possibilité d'interagir avec l'application, si ce n'est la possibilité de s'authentifier en saisissant ses identifiants (il devient alors **utilisateur**).

- II. **En tant qu'utilisateur :**

Une fois authentifié, l'**utilisateur** a la possibilité de se rendre sur le **Dashboard** de l'application et consulter les informations liées à son compte. Pour le reste de ses possibilités, cela dépend du type de droit qui lui sont attribués.

- **Administrateur :**

L'administrateur est le premier compte créé lors de la mise en place de l'authentification. Celui dispose de tous les droits sur l'application. Il a notamment la possibilité de créer, modifier ou supprimer d'autres utilisateurs authentifiés destinés à utiliser l'application. Il peut également définir les droits qui seront octroyés à ces utilisateurs. Il peut également agir sur les utilisateurs et périphériques présents dans la base de données, et aussi, importer des données issues des bases externes pour fournir celle de l'application lorsque c'est nécessaire.

- **Technicien :**

Le technicien peut agir sur les utilisateurs et périphériques présents dans la base de données. Il peut également importer des données issues des bases externes pour fournir celle de l'application si c'est nécessaire.

- **Administratif :**

L'administratif peut lui consulter et faire des recherches sur les utilisateurs et périphériques.

3.3 - Les cas d'utilisation

3.3.1 - Cas d'utilisation général :

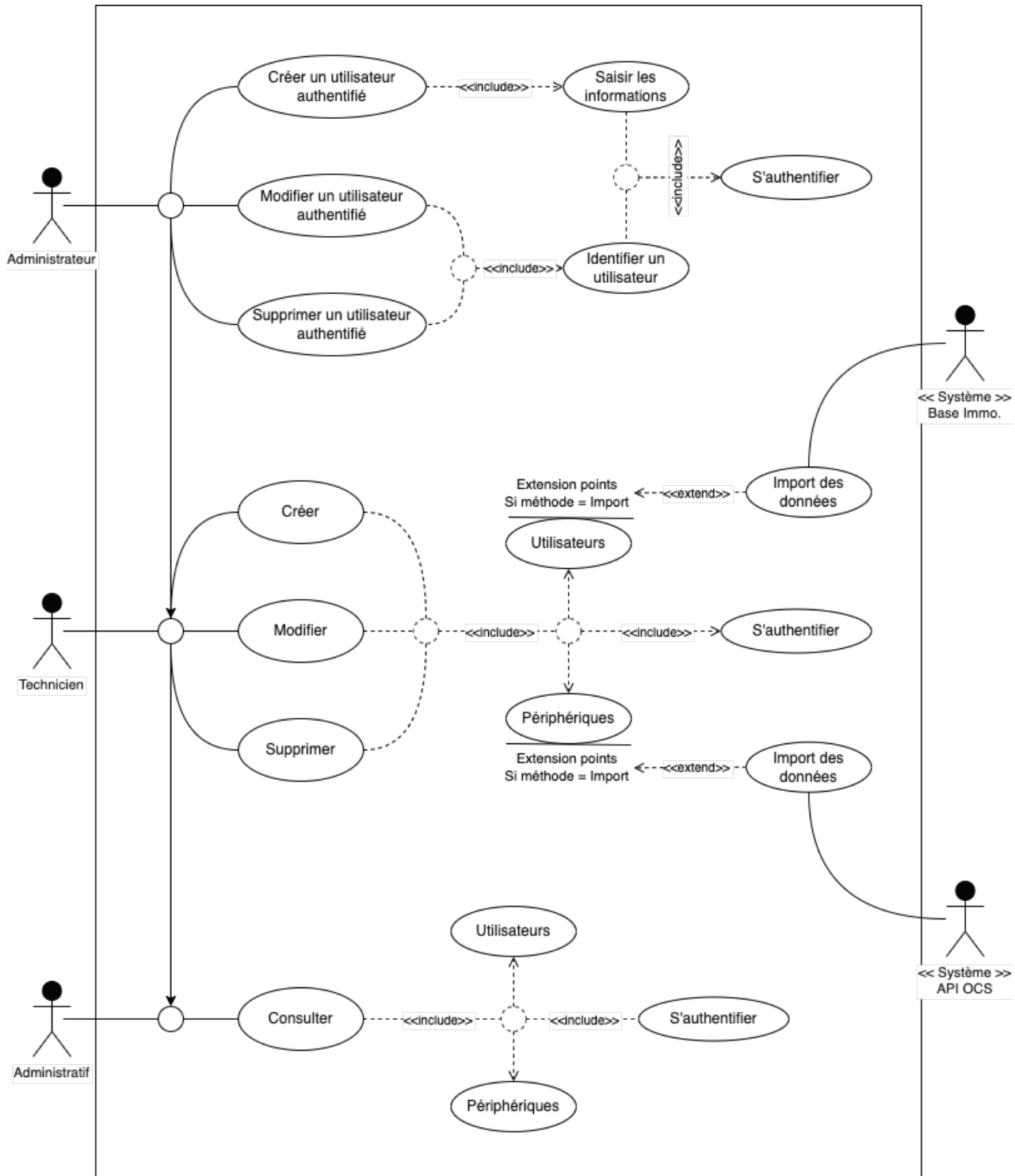


Diagramme de usecase général

3.3.1.1 - Description du diagramme

Nous pouvons voir dans ce diagramme, plusieurs cas d'utilisations ainsi que l'interaction avec les différents acteurs :

- **Administrateur :**

Dans ce scénario, on met en évidence le fait que l'administrateur soit le seul utilisateur à pouvoir créer, modifier et supprimer d'autres utilisateurs authentifiés. Dans le cas de la création, il devra dans un premier temps saisir les informations qui identifieront le nouvel utilisateur. Dans les deux autres, il devra donc préalablement identifier les comptes avant de les modifier/supprimer. Bien entendu, pour accéder à cela, il devra préalablement s'authentifier. Il est important de noter que l'administrateur pourra également faire les actions citées dans les deux cas illustrant les deux autres acteurs (technicien, administratif).

- **Technicien :**

Ici, on met en évidence le fait que le technicien peut créer, modifier et supprimer de nouveaux utilisateurs et périphériques dans la base principale de l'application. Encore une fois, ces actions ne seront possibles que s'il s'est authentifié. Dans un scénario alternatif, on peut voir également que les deux acteurs secondaires (Api **OCS** et base **Immo.**) peuvent être sollicités pour importer les données concernant les utilisateurs et périphériques.

- **Administratif :**

Dans ce dernier cas, nous illustrons la possibilité, pour l'administratif, de consulter les informations contenues dans la base de données principale, que ce soit les utilisateurs ou les périphériques. De la même manière que précédemment, il doit pour cela, s'authentifier.

3.3.2 - Cas d'utilisation « Authentification » :

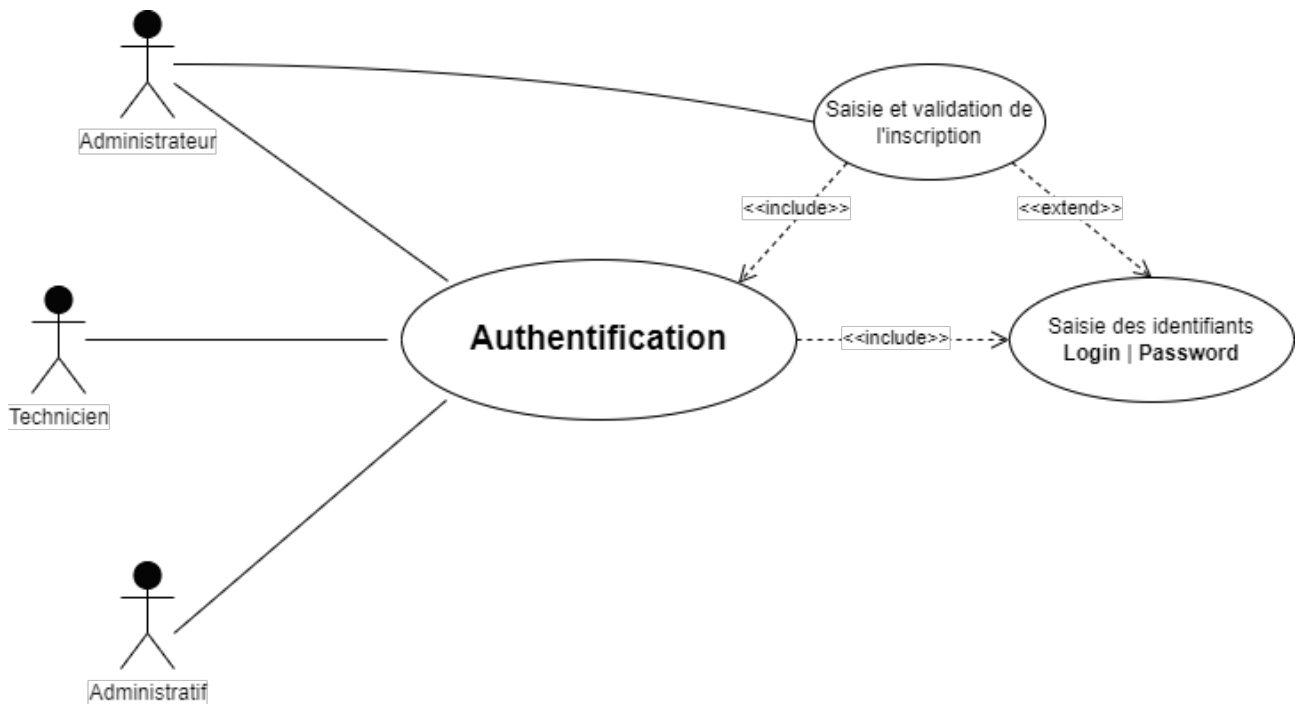


Diagramme de usecase « Authentification »

3.3.2.1 - Description du diagramme

Le diagramme ci-dessous met en avant la façon dont les acteurs principaux doivent s'authentifier sur l'application.

- **Administrateur :**

L'administrateur doit s'authentifier en utilisant ses identifiants. Il est à noter qu'il se charge également de saisir et valider l'inscription des autres acteurs.

▪ **Technicien et Administratif :**

Concernant ces acteurs, ils doivent préalablement s'authentifier (grâce aux identifiants fournis par l'administrateur) pour pouvoir utiliser l'application.

3.3.3 - Cas d'utilisation « Gestion de l'inventaire »

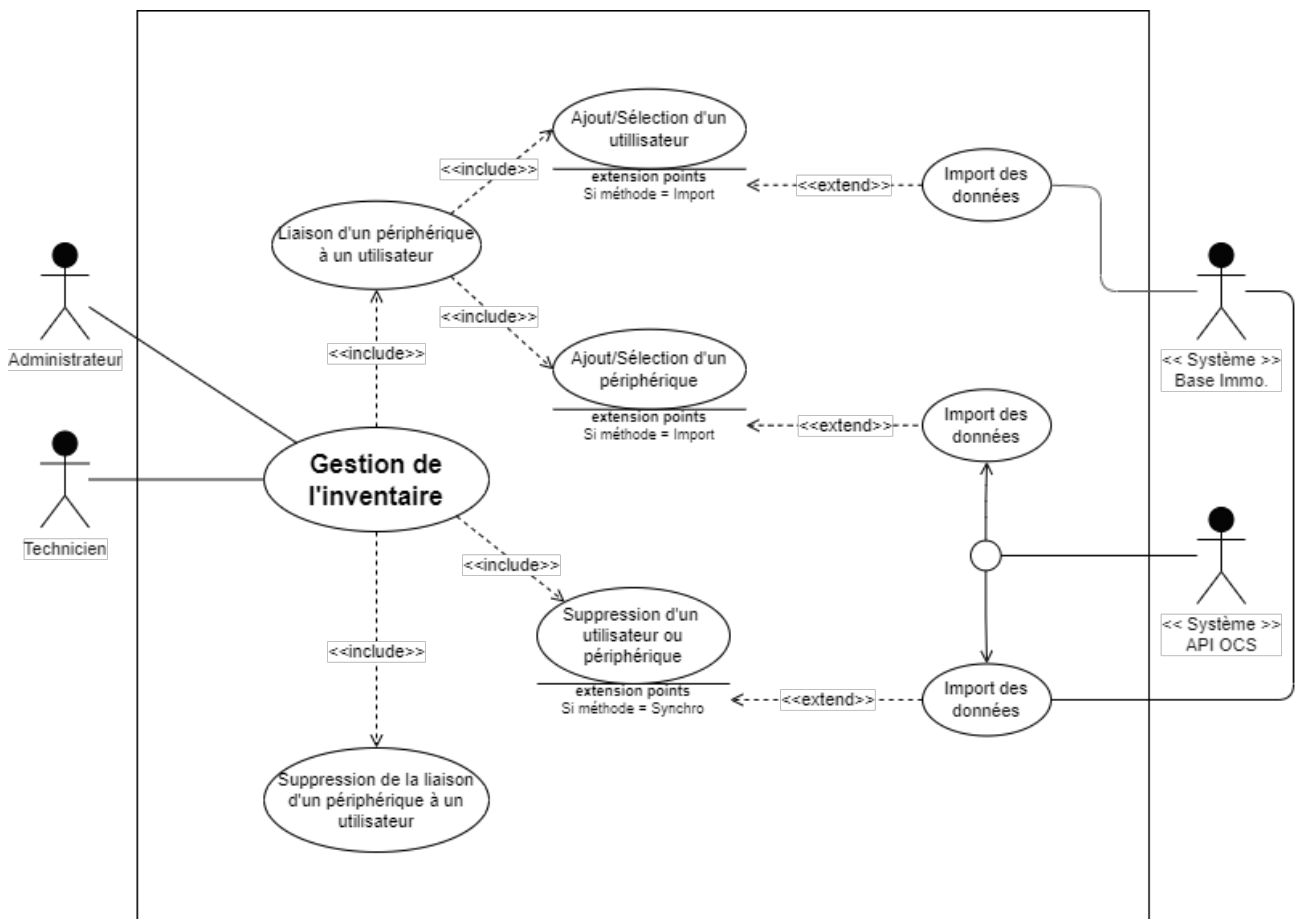


Diagramme de usecase « Gestion de l'inventaire »

3.3.3.1 - Description du diagramme

Le diagramme ci-dessus, permet de visualiser le cas d'utilisation de la gestion de l'inventaire, partie gérée uniquement par l'administrateur et le technicien.

Voici les différentes étapes décrites :

- L'acteur principal attribue un périphérique à un utilisateur :
 - Un utilisateur doit être ajouté ou sélectionné (s'il est déjà existant)
 - Un périphérique doit être ajouté ou sélectionné (s'il est déjà existant)
 - L'ajout d'utilisateurs et périphériques peut se faire manuellement par l'acteur principal, ou encore par l'import des données contenues dans les bases de données **Immo** et **OCS**.
- Il supprime la liaison d'un périphérique à un utilisateur (pour l'attribuer à un autre ou le sortir du parc géré par exemple)
- Il supprime un utilisateur ou un périphérique :
 - L'acteur principal peut le faire manuellement
 - Cela peut également se faire par le biais d'une synchronisation avec les informations présentes dans les base de données **Immo** et **OCS**.

3.3.4 - Cas d'utilisation « Consultation de l'inventaire »

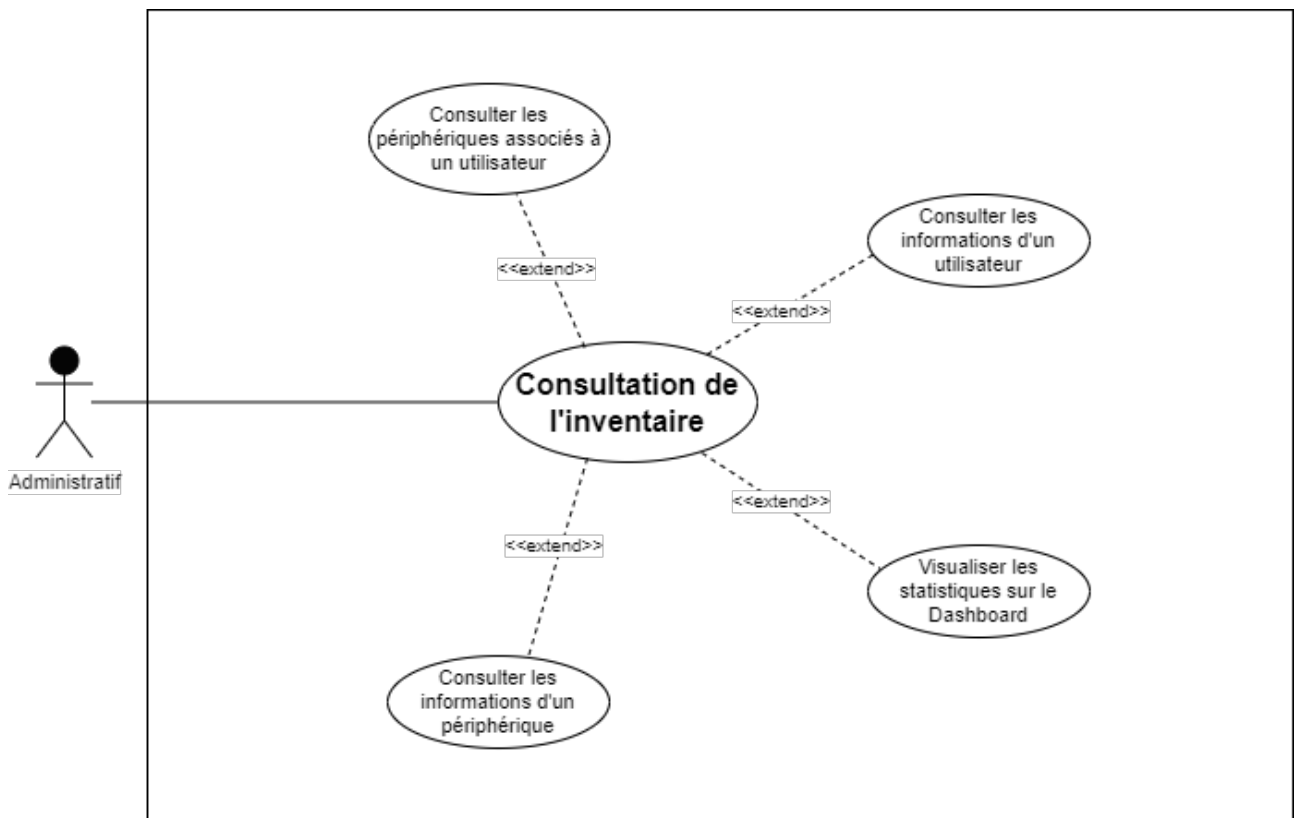


Diagramme de usecase « Consultation de l'inventaire»

3.3.4.1 - Description du diagramme

Le diagramme ci-dessus, nous permet d'observer le cas d'utilisation de la consultation de l'inventaire.

Rappel : Nous partons du principe que des informations ont été importées ou saisies dans la base de données de l'application.

L'administratif consulte l'application, il voit :

- Différentes statistiques liées aux informations présentes dans la base de données de l'application sur son « **Dashboard** »
- Les périphériques associés à des utilisateurs
- Les informations concernant un utilisateur sélectionné
- Les informations concernant un périphérique sélectionné

4 - LE DOMAINE FONCTIONNEL

4.1 - Packages

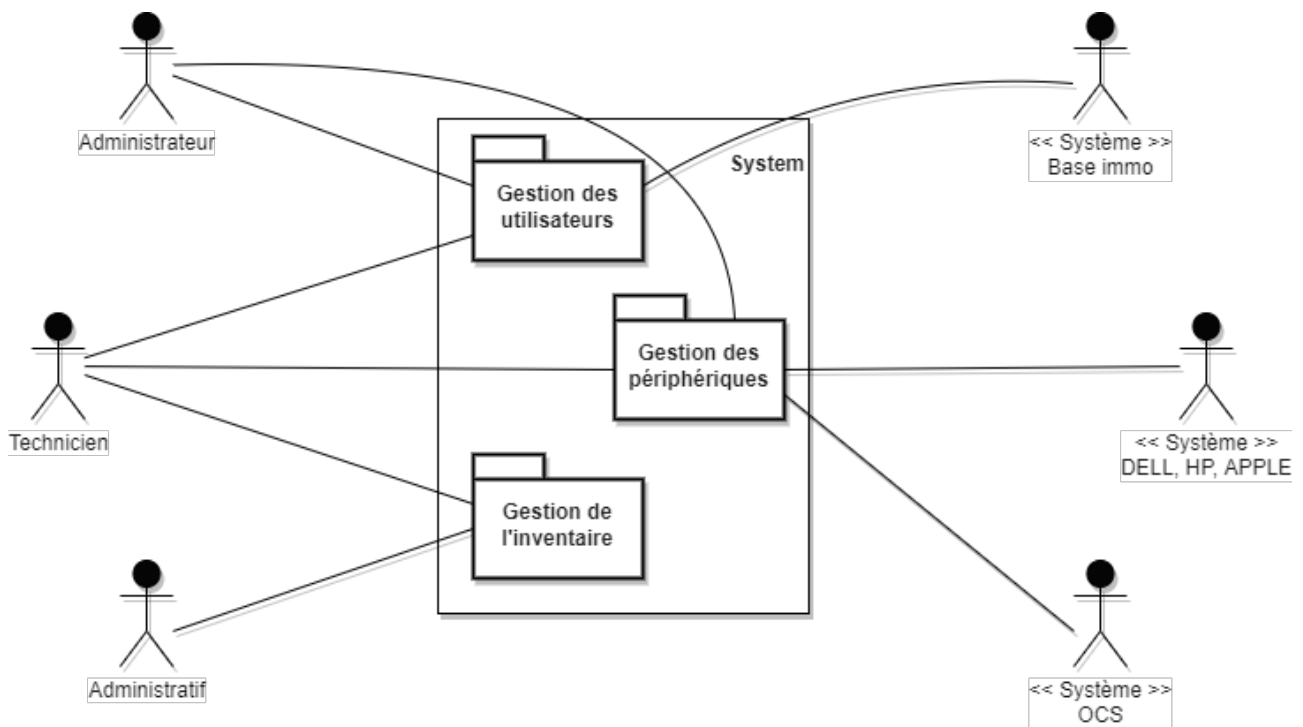


Diagramme de packages

4.1.1.1 - Description du diagramme

Nous avons fait le choix de découper le « **système** » en trois **packages** différents :

- **Gestion des utilisateurs :**

L'administrateur et le technicien seront amenés à interagir avec les différentes fonctionnalités de ce package. L'administrateur lors de la gestion des comptes utilisateurs authentifiés, puis l'administrateur et le technicien lors de la gestion des utilisateurs qui seront liés, ensuite, à des périphériques. A noter que la base « **Immo** » interagira également avec ce package lors de l'import d'utilisateurs.

- **Gestion des périphériques :**

L'administrateur et le technicien interagiront également tous les deux avec ce package. Cela se fera lors de la gestion des périphériques éventuellement liés à des utilisateurs. L'API « **OCS** » ainsi que les API **Dell**, **HP** et **Apple** seront également sollicitées par certaines fonctionnalités de ce package. Pour la première lorsqu'il faudra renseigner les configurations des différents périphériques, et les API constructeurs lorsqu'il faudra notamment renseigner la garantie ou d'autres informations de ce type.

- **Gestion de l'inventaire :**

Ce **package** concernera principalement le technicien et l'administratif (même si techniquement, l'administrateur pourrait également l'être, mais ce n'est pas là sa fonction principale). Les acteurs pourront, ici, consulter les différentes informations contenues dans l'inventaire mais également gérer les associations utilisateurs et périphériques si ses droits lui permettent.

5 - APPLICATION « OC INVENTORY »

Le développement de la solution se fera « **From scratch** », le délai étant suffisant pour développer les différentes fonctionnalités et effectuer tous les tests avant la livraison de l'application.

Voici les principaux éléments de la solution **OC Inventory** :

Interface Web	HTML5 – JavaScript – CSS / Bootstrap
Authentification	Python « Django » – PostgreSQL
Gestion de l'inventaire	Python « Django » – PostgreSQL
Gestion des utilisateurs / périphériques	Python « Django » – PostgreSQL
Import des données utilisateurs	Base de données Immo
Import des données périphériques	API OCS – (Dell, HP, Apple)

Le développement se fera par l'intermédiaire du Framework **Django** basé sur le langage de programmation **Python**. Ce Framework a été utilisé pour concevoir des sites très célèbres, nous pourrions citer notamment :

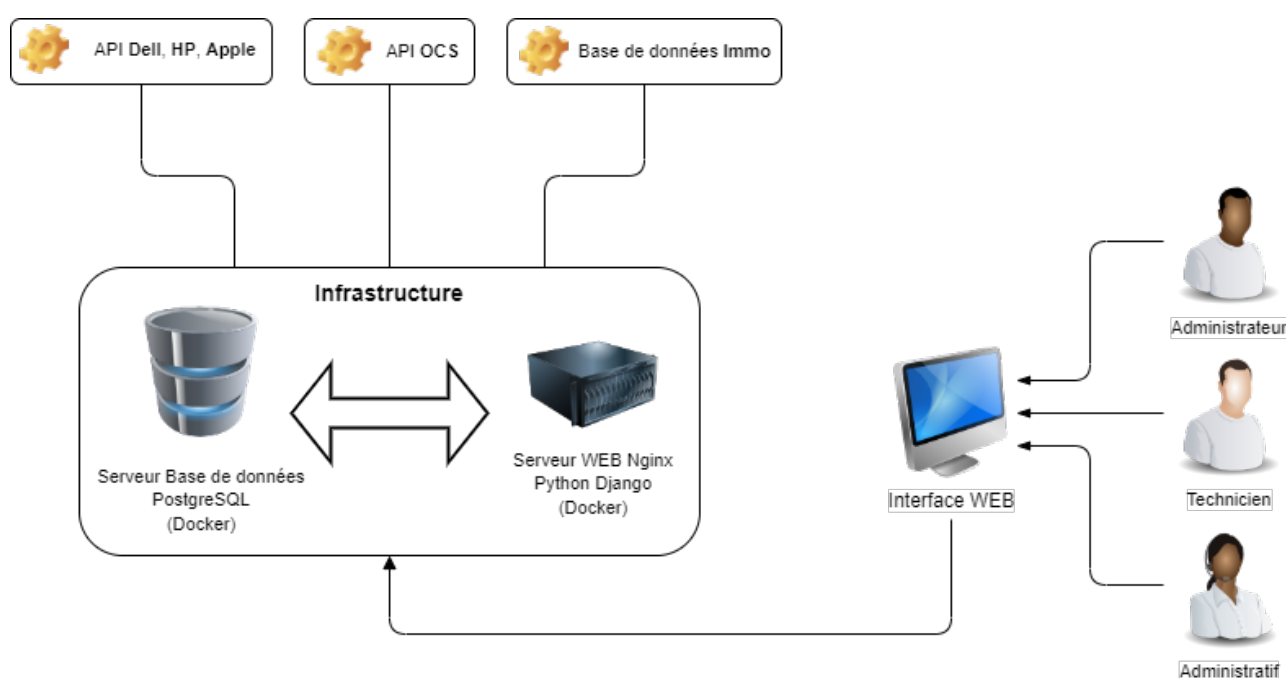
- [Pinterest](#)
- [Instagram](#)
- [The Washington Post](#)
- [National Geographic](#)
- Et bien d'autres...

HTML5, **Javascript** et **CSS** seront également utilisés, notamment pour la partie « **application web** ». Un Template type **Bootstrap** sera utilisé pour la partie graphique du site, le choix de celui-ci restant à définir.

Concernant le **SGBDR**, nous utiliserons **PostgreSQL**. Celui-ci présente l'avantage d'être libre mais également d'être reconnu pour sa stabilité.

La partie import des données se fera par l'intermédiaire d'autres bases de données et API. La base de données « **Immo** » sera sollicitée pour intégrer de nouveaux utilisateurs alors que l'API OCS permettra de renseigner les différents périphériques qui seront associés. Les API constructeurs seront utilisées, par la suite, pour avoir plus de détails concernant la « validité » de la garantie des périphériques ou encore l'état de la maintenance si une machine devait être retournée.

Le diagramme ci-dessous permet d'illustrer la partie matérielle de la solution :



Vue « **Serveurs** » de la solution **OC Inventory**

L'interface WEB étant accessible depuis n'importe quel périphérique (**PC/Mac**, tablette **iOS/Android**, Smartphone **iOS/Android**), il est nécessaire que, celle-ci, soit « **Responsive** ». C'est-à-dire qu'elle pourra s'adapter à n'importe quel périphérique et quel que soit la résolution.

L'application sera hébergée sur un serveur (**Linux**) Web **NGINX** où seront, notamment, installés **Python** et **Django**. Un autre serveur (**Linux**) où **PostgreSQL** sera installé hébergera la base de données. Le serveur Web pourra communiquer avec la base de données lorsque cela sera requis par l'application.

6 - GLOSSAIRE

A

application WEB

Application manipulable en ligne grâce à un navigateur internet 8

B

Bootstrap

Framework CSS utilisé pour l'aspect graphique..... 19

C

CSS

Cascading Style Sheets..... 19

D

Dashboard

Tableau de bord 5, 8, 16

Django

Framework web basé sur le langage Python 19

DSI

Directrice des Systèmes d'information 4

F

Framework

Ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application..... 19

From scratch

Développement de l'application en partant de "zéro" 19

H

HTML5

HyperText Markup Language 5 19

J

JavaScript

Language de programmation JavaScript..... 19

P**packages**

Groupement d'éléments en un ensemble cohérent	17
PostgreSQL	
Système de gestion de base de données relationnelle	19, 20
Python	
Langage de programmation Python	19, 20

R**Responsive**

Dit d'une application pouvant s'adapter à différentes résolutions d'écran	20
---	----

S**SGBDR**

Système de gestion de base	19
----------------------------------	----