

UNIVERSITE DE YAOUNDE 1

FACULTE DES SCIENCES

BP 812 Yaoundé



UNIVERSITY OF YAOUNDE 1

FACULTY OF SCIENCES

P.O Box 812 Yaounde

## **INF3 6: PROJET**

# **GESTION DES SERRURES CONNECTEES**

## **Cahier des charges Fonctionnel**

### **MEMBRE DU GROUPE**

- **BANDE TEUMOU MANUELA VIGNOL 18N2784**
- **DJEUMENI DJOMBISSIE LEVINNE CLEMENCE 19M2077**
- **KEUNMENE TEGANG Joseph Magloire 16y349**
- **METILI NGUELE Donatien 17A2387**
- **ZOGO ABOUMA ZOZIME ACHAIRE 18N2824**

**Superviseur:** Dr. ABESSOLO

**Encadreur :** Dr. MESSI NGUELE

**Superviseur adjoint :** M. Esteban PAMOE

## Table des matières

<b>I. Introduction</b>	3
<b>I. PRESENTATION DE L'ENVIRONNEMENT DE DEVELOPPENMENT</b>	4
1. Le système d'exploitation	4
2. IDE : Environnement de Développement Intégré	4
3. IDE : Langages de programmation	4
4. Outils et technologies à utiliser	4
<b>II. AVANTAGES DU FRAMEWORK CHOISI</b>	4
1. Django : Framework Python	4
2. Flutter: Framework Dart	5
3. Node JS: Librairie JavaScript	5
<b>III. Choix de la méthodologie de conception</b>	6
Présentation d'UML	6
<b>IV. Diagrammes UML</b>	6
1. Diagrammes de cas d'utilisation	6
2. Diagramme de classe	7
3. Diagrammes de séquence	8
a. Diagramme de séquence fonctionnalité : ajouter un accès	8
b. Diagramme de séquence fonctionnalité :supprimer un accès	9
c. Diagramme de séquence fonctionnalité :modifier un accès	10
d. Diagramme de séquence fonctionnalité : lister les codes de déverrouillage	11
e. Diagramme de séquence fonctionnalité : afficher la liste des serrures	11
f. Diagramme de séquence fonctionnalité: afficher la liste des personnes ayant accès à une serrure	12
g. Diagramme séquence fonctionnalité : s'authentifier coté web & mobile Module gestion des serrures	14
h. Diagramme séquence fonctionnalité : Ajout d'une serrure	15
4. Diagrammes d'état transition	15
<b>V. Modèle de base de données</b>	16
1. Modelé de conception des données	16
2. Modelé Logique des données	16
<b>VI. Conclusion</b>	17

# I. INTRODUCTION

La modélisation conceptuelle et organisationnelle constitue une étape importante dans la convergence des notations utilisées dans le domaine de l'analyse de conception objet puisqu'elle représente une synthèse pour notre système. Le système à développer est celui d'un système permettant de gérer les serrures connectées. Pour ce faire, il est primordial de faire une conception du système pour permettre une bonne visualisation et compréhension détaillée du système afin de faciliter son développement

Dans ce chapitre nous commencerons par le modèle conceptuel détaillé de notre application, ensuite nous exposons le modèle logique des données. Enfin nous clôturons ce chapitre par une conclusion.

# I. PRESENTATION DE L'ENVIRONNEMENT DE DEVELOPPEMENT

## 1. Le système d'exploitation

Notre projet sera réalisé dans le Système Linux via une de ses distributions : **Ubuntu**. Il offre un environnement de travail totalement gratuit, sécurisé et sans virus.

## 2. IDE : Environnement de Développement Intégré

**VS Code** est IDE gratuit, léger, qui prends en charge plusieurs langages et qui dispose de nombreuses extensions de plug-in pour faciliter le développement

## 3. IDE : Langages de programmation

- Pour le développement de l'application Web : **Python**

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet.

- Pour le développement de l'application Mobile :

- Front end : **Dart**

Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est développé par Google et est utilisé pour créer des applications mobiles, de bureau, de serveur et web.

- Back-end : **JavaScript**

## 4. Outils et technologies à utiliser

Pour établir notre serveur de base de données locale, **XAMPP** sera utilisée. XAMPP est un progiciel qui permet d'établir un serveur local pouvant fonctionner sur un ordinateur de bureau ou portable

# II. AVANTAGES DU FRAMEWORK CHOISI

## 1. Django : Framework Python

- **La documentation** : un Framework mal documenté fait perdre plus de temps qu'il n'en fait gagner, Django offre une documentation très riche.

- **La communauté** : lorsqu'on rencontre une difficulté il est bon de recevoir une aide rapide pour ne pas rester bloqué longtemps sur un problème de détail, la communauté de Django est réactive.
- **La durée d'apprentissage** : un temps d'apprentissage trop long fait perdre l'intérêt d'un framework même si on factorise notre connaissance en multipliant les cas d'application, Django est rapide à prendre en mains tout en ne faisant aucune concession sur les fonctionnalités.
- **La richesse des outils** : certains framework sont intéressants mais nécessitent de nombreuses greffes pour devenir vraiment opérationnels. Laravel est à la base suffisamment complète pour ne nécessiter que des extensions pour des besoins particuliers.
- **La simplicité** : utiliser un framework et se retrouver à écrire un code plus complexe que celui qu'on aurait utilisé avec simplement du PHP n'est pas une opération judicieuse, le code qu'on écrit avec Django est toujours simple, lisible, concis.
- **Les possibilités d'extension** : on doit pouvoir facilement étendre les fonctionnalités d'un framework sans détruire son organisation, Django permet de le faire avec élégance en suivant toujours les mêmes principes avec ses bundles.

## 2. Flutter: Framework Dart

Flutter est un projet open source développé par Google. Il s'appuie sur le langage de programmation DART, lancé en 2011.

- Flutter est un Framework capable de faire du développement multiplateformes (plateformes Ios et Android).
- Flutter propose un code commun qui fonctionne aussi bien sur iOS que sur Android.
- Les corrections de bugs sont rapides et régulières.

## 3. Node JS: Librairie JavaScript

Node JS est une librairie de JavaScript qui présente de nombreux atouts.

- **Rapidité**  
Les tâches courantes des applications web, comme la lecture ou l'écriture dans la base de données, la lecture et l'écriture dans les connexions réseau et la lecture ou l'écriture dans le système de fichiers, peuvent être exécutées rapidement à l'aide de Node.js.
- **Communauté active**  
Node.js encourage le partage avec la présence du gestionnaire de Node Package ou NPM. Il comprend le dépôt de 50 000 modules, qui aide les développeurs à créer des solutions efficaces.

- **Évolutivité**

Node vous permet d'étendre les demandes simultanées, ce que les autres langages ne peuvent le faire. Certains ont atteint des niveaux d'extensibilité de plus d'un million de requêtes simultanées. Bien sûr, tout dépend du travail que vous faites derrière chaque requête et du nombre de ressources dont vous disposez, même si Node est toujours aussi performant pour faire évoluer les choses.

### **III. Choix de la méthodologie de conception**

#### **Présentation d'UML**

UML (Unified Modeling Language) est un langage formel et normalisé en termes de modélisation objet. Son indépendance par rapport aux langages de programmation, aux domaines de l'application et aux processus, son caractère polyvalent et sa souplesse ont fait de lui un langage universel. En plus UML est essentiellement un support de communication, qui facilite la représentation et la compréhension de solution objet. Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation des solutions. L'aspect de sa notation, limite l'ambiguïté et les incompréhensions.

### **IV. Diagrammes UML**

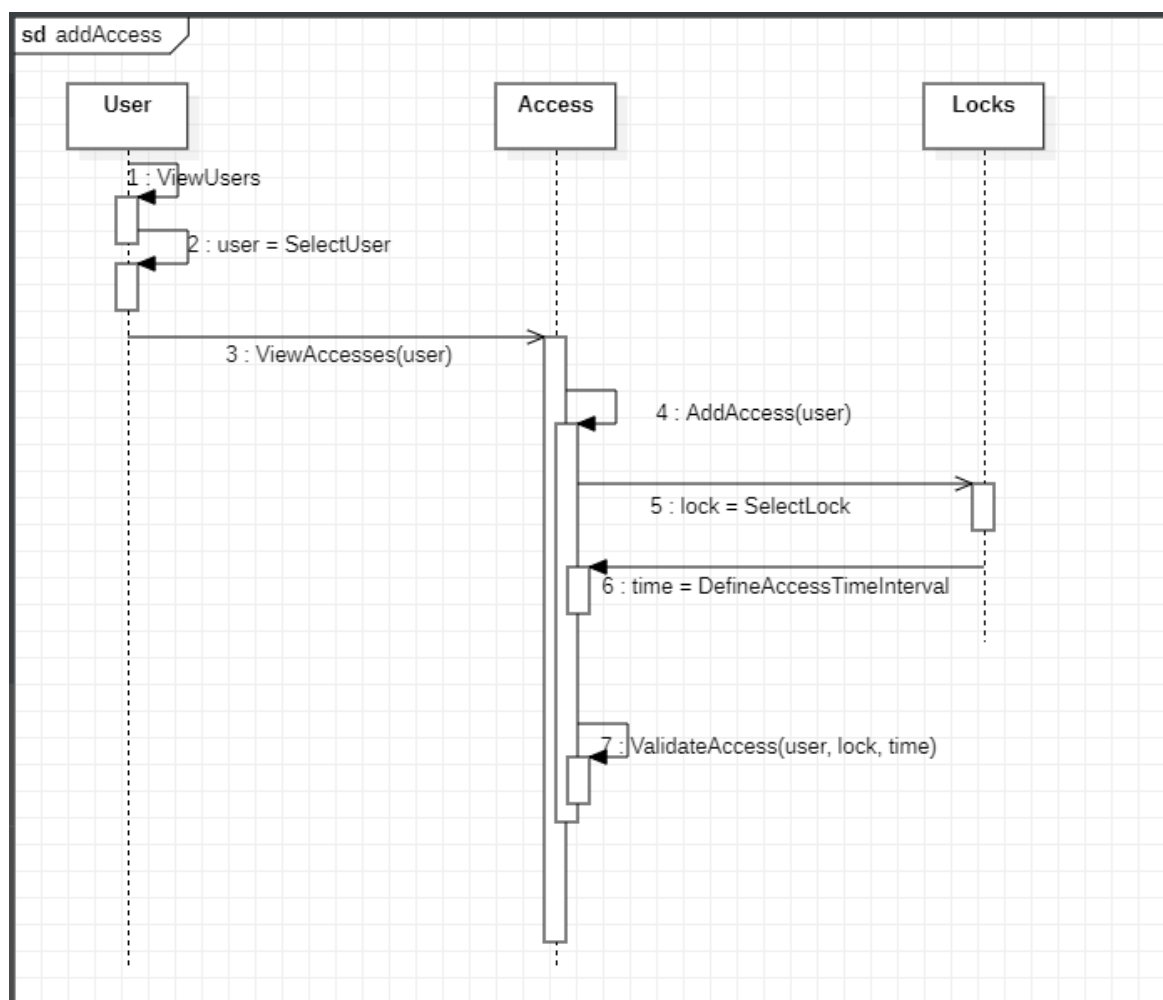
#### **1. Diagrammes de cas d'utilisation**

Ce diagramme présente les différents acteurs du système ainsi que leurs cas d'utilisations respectifs



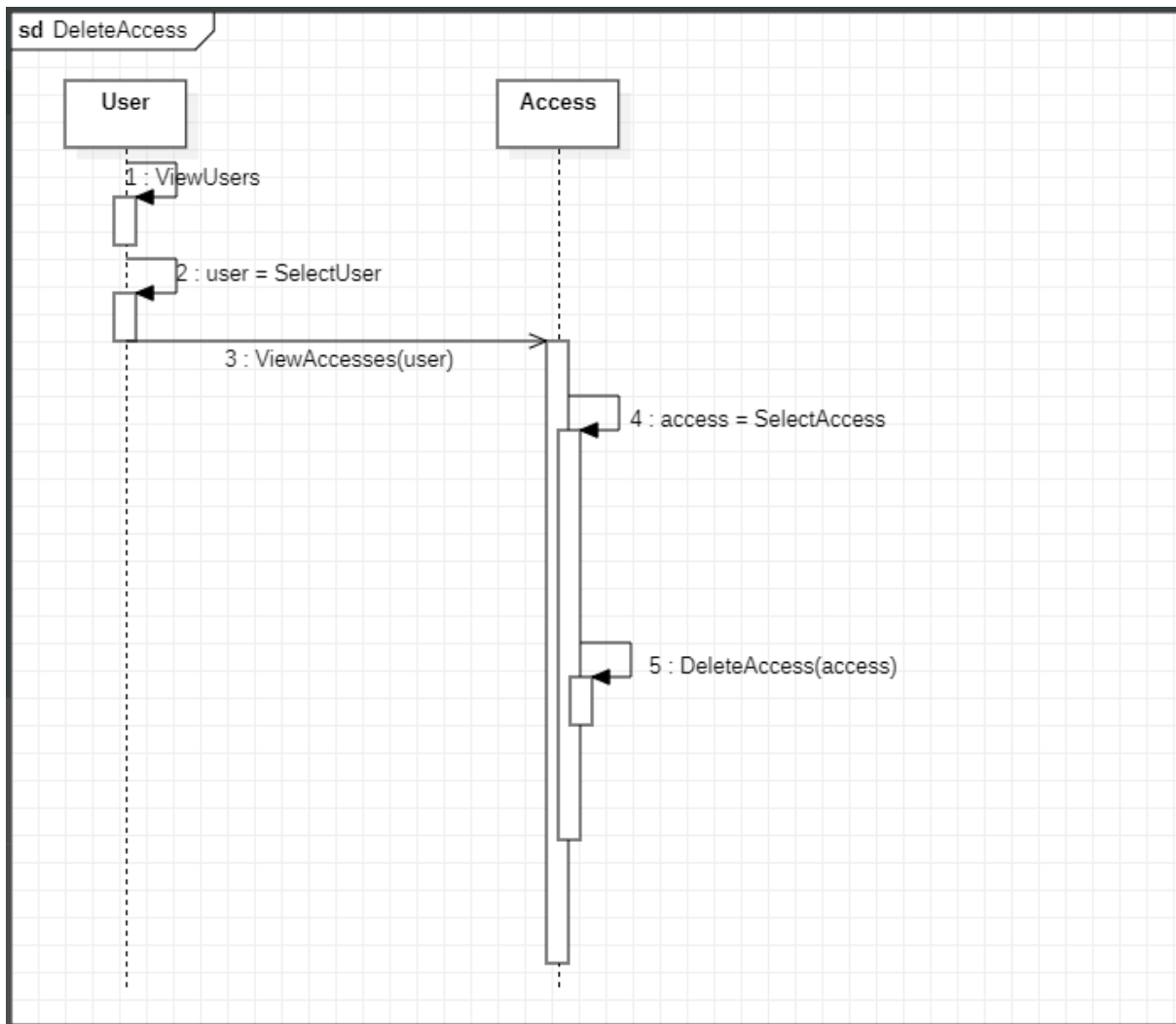
### 3. Diagrammes de séquence

#### a. Diagramme de séquence fonctionnalité : ajouter un accès

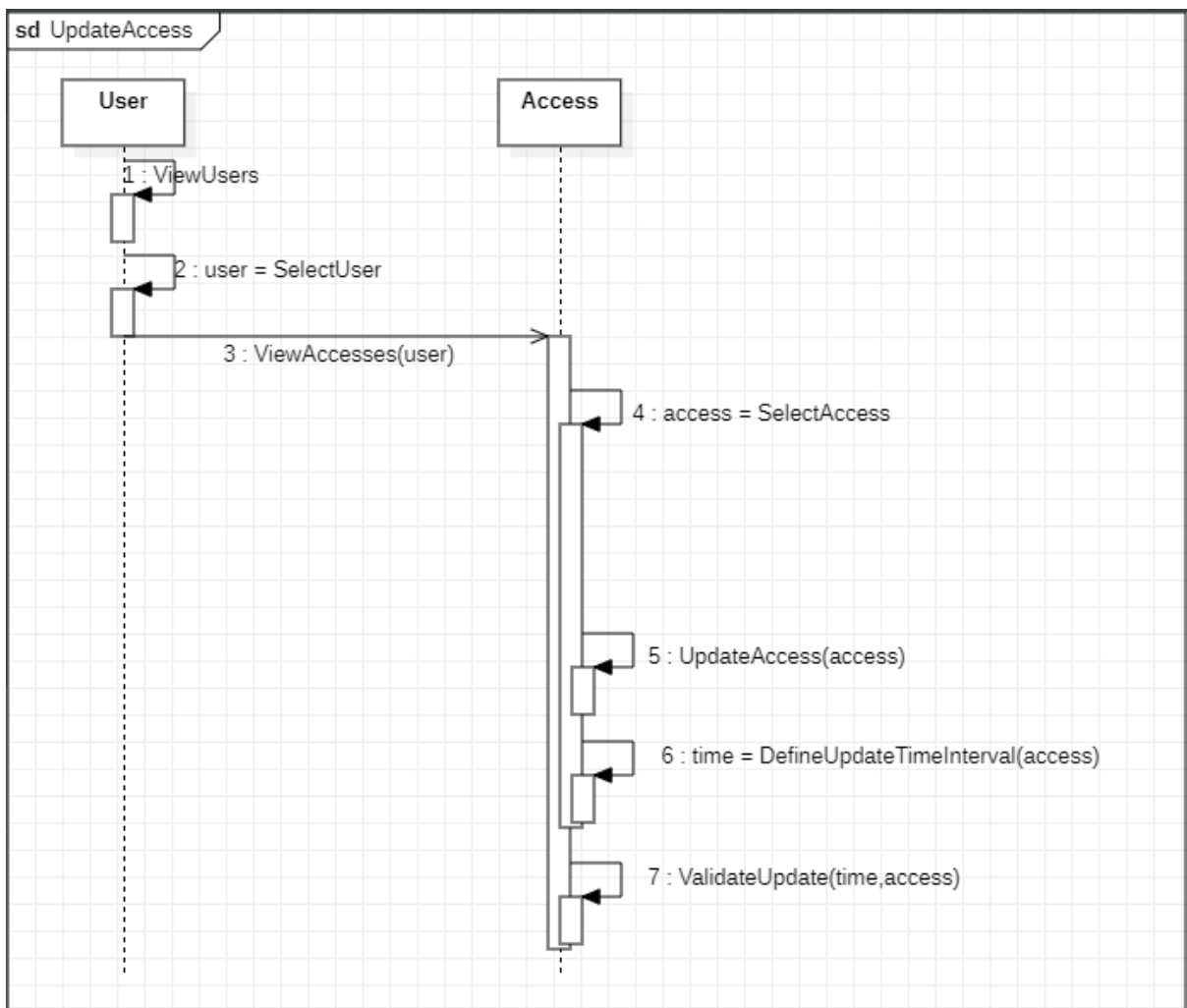




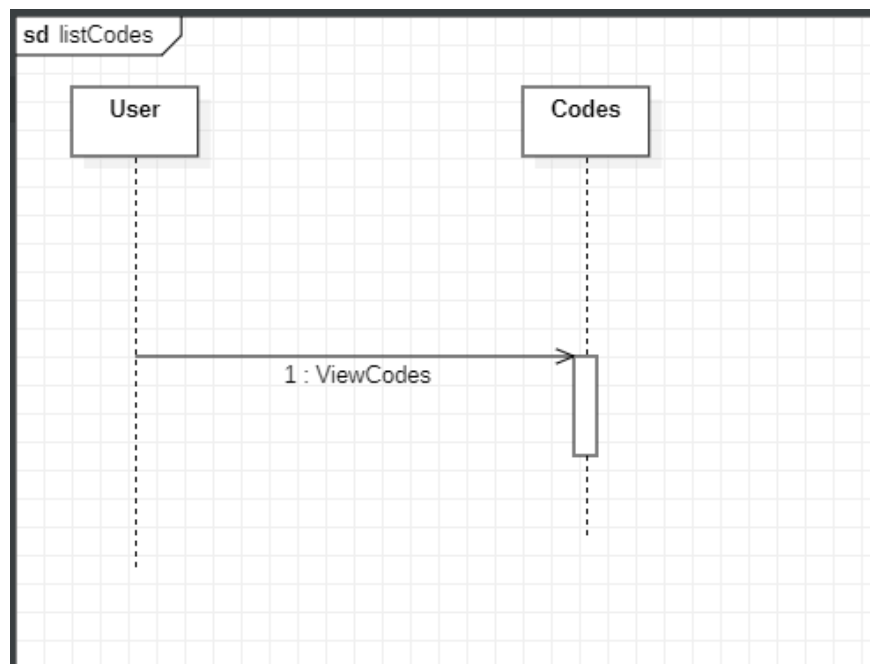
b. Diagramme de séquence fonctionnalité :supprimer un accès



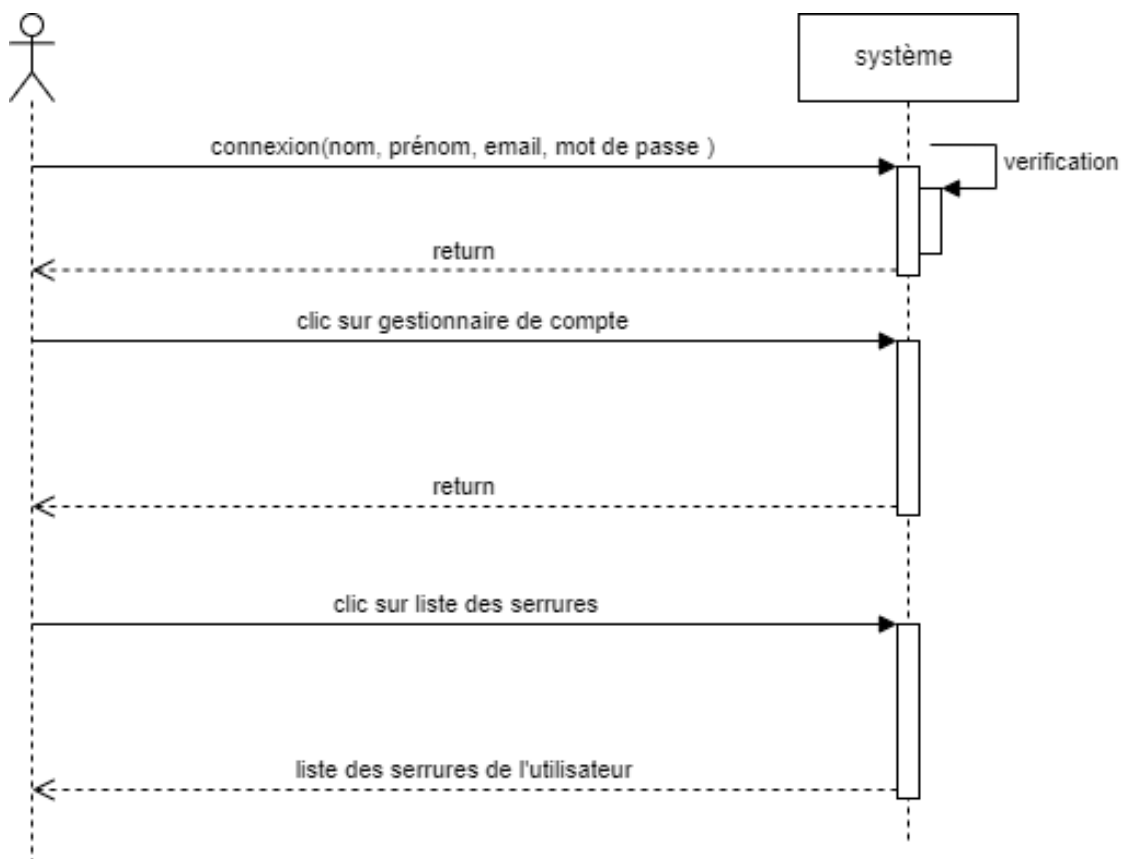
c. Diagramme de séquence fonctionnalité : modifier un accès



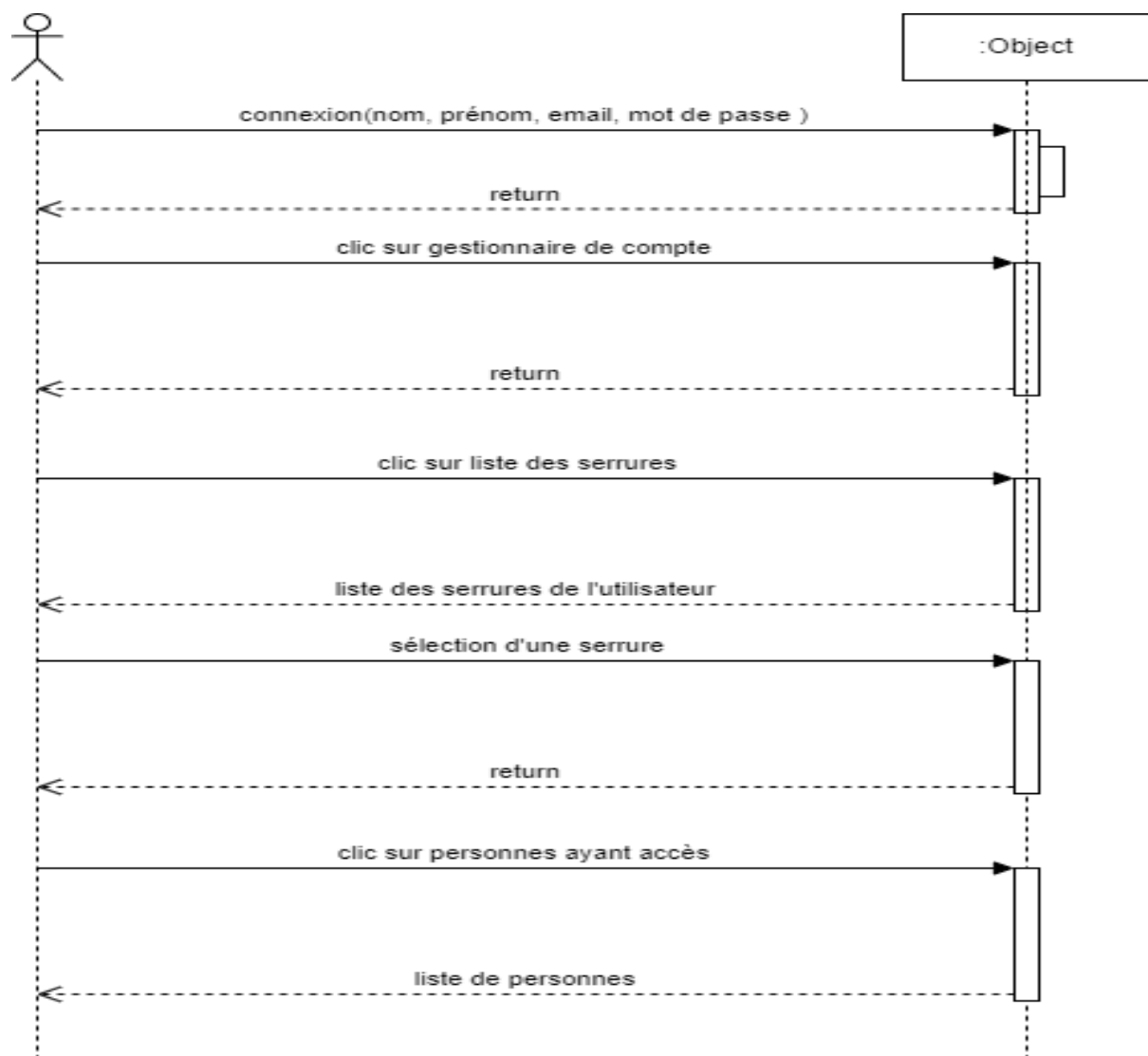
d. Diagramme de séquence fonctionnalité : lister les codes de déverrouillage



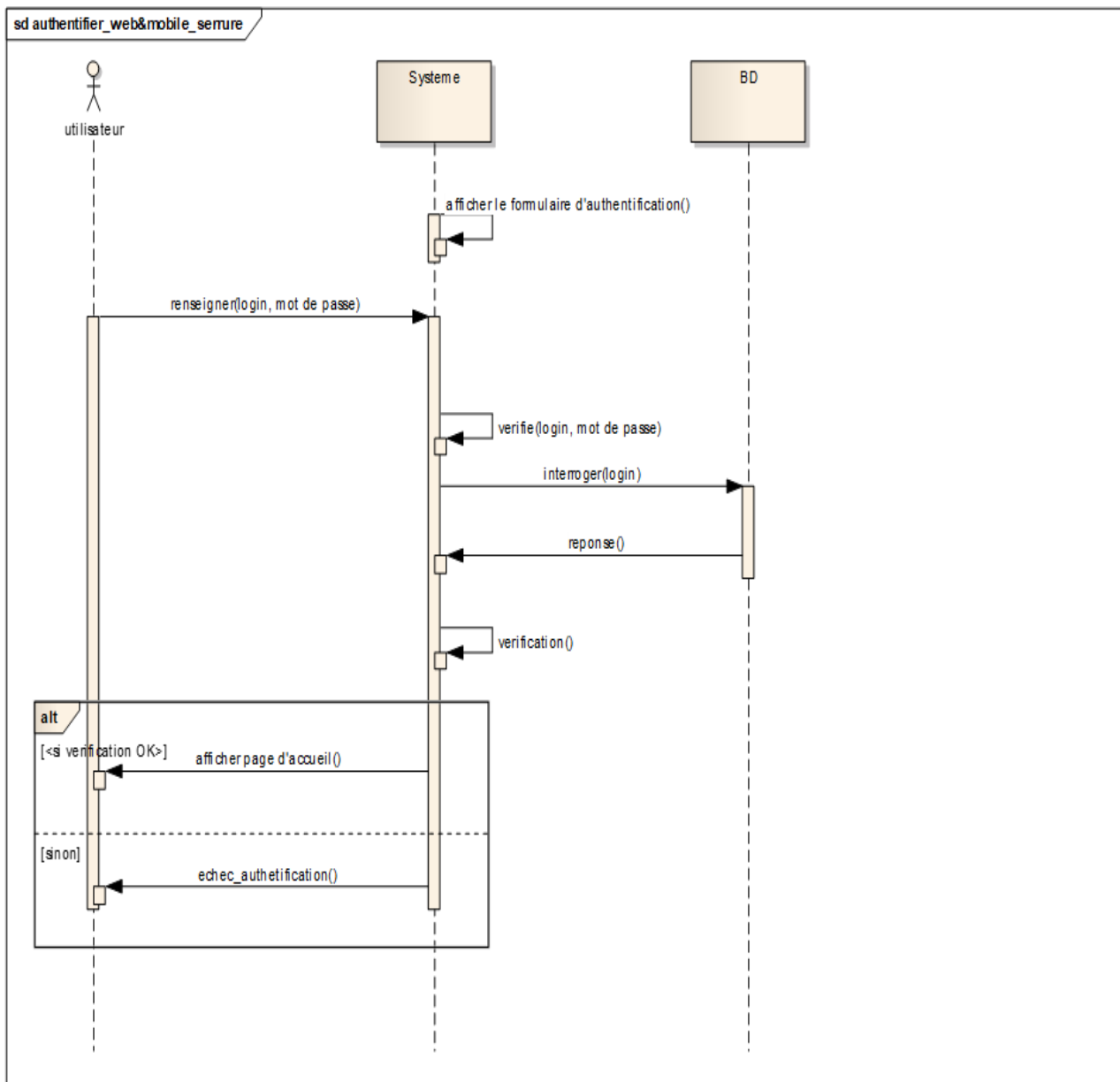
e. Diagramme de séquence fonctionnalité : afficher la liste des serrures



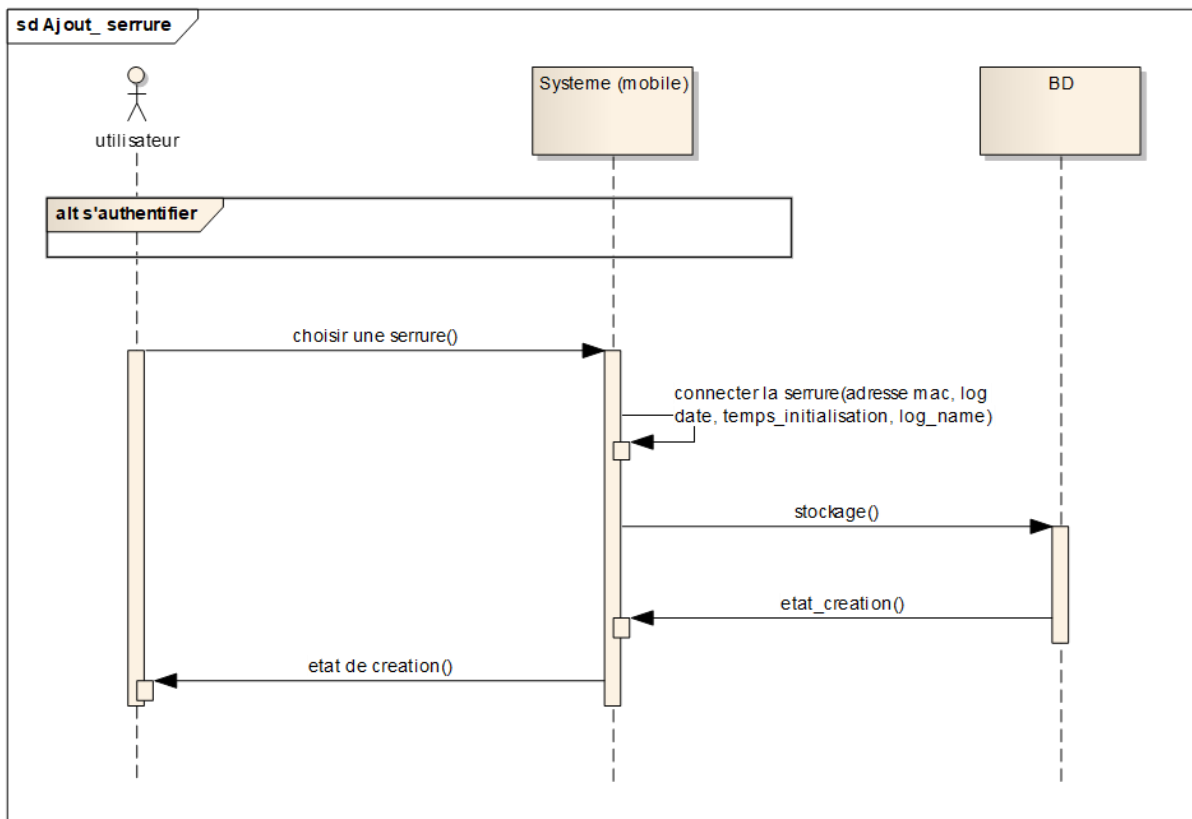
- f. Diagramme de séquence fonctionnalité: afficher la liste des personnes ayant accès à une serrure



- g. Diagramme séquence fonctionnalité : s'authentifier coté web & mobile Module gestion des serrures

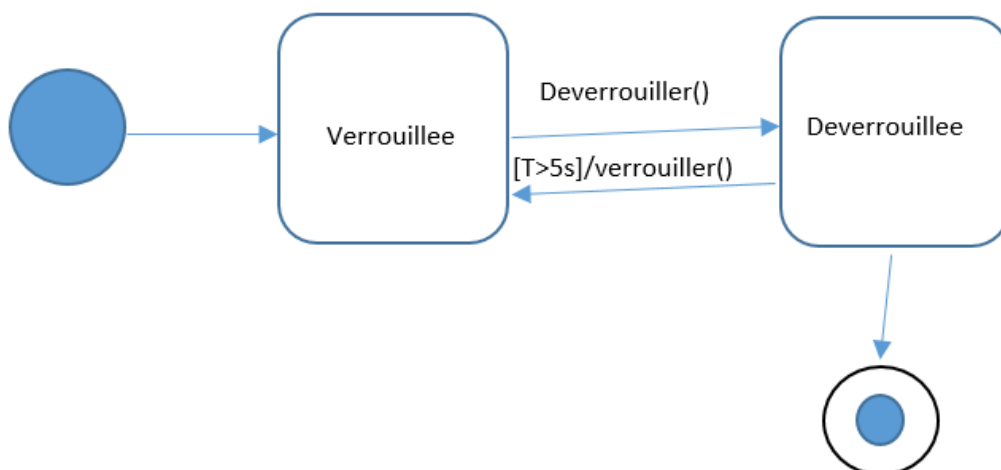


h. Diagramme séquence fonctionnalité : Ajout d'une serrure



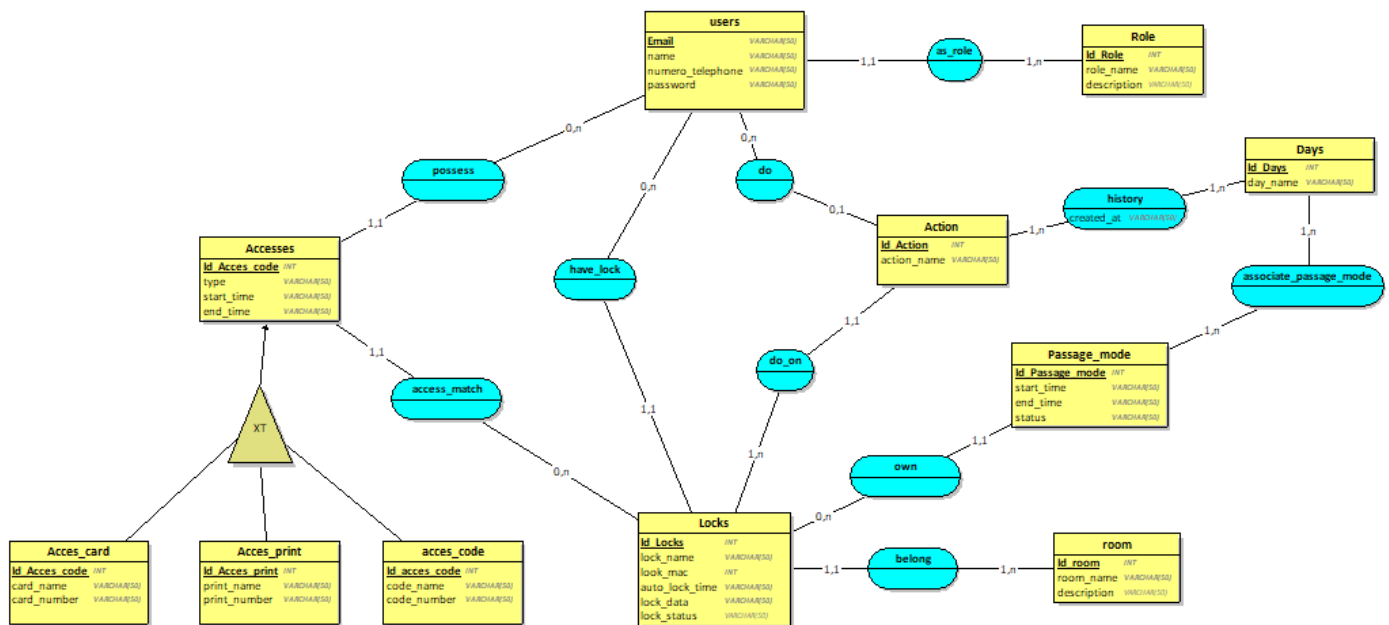
4. Diagrammes d'état transition

Déverrouiller une serrure

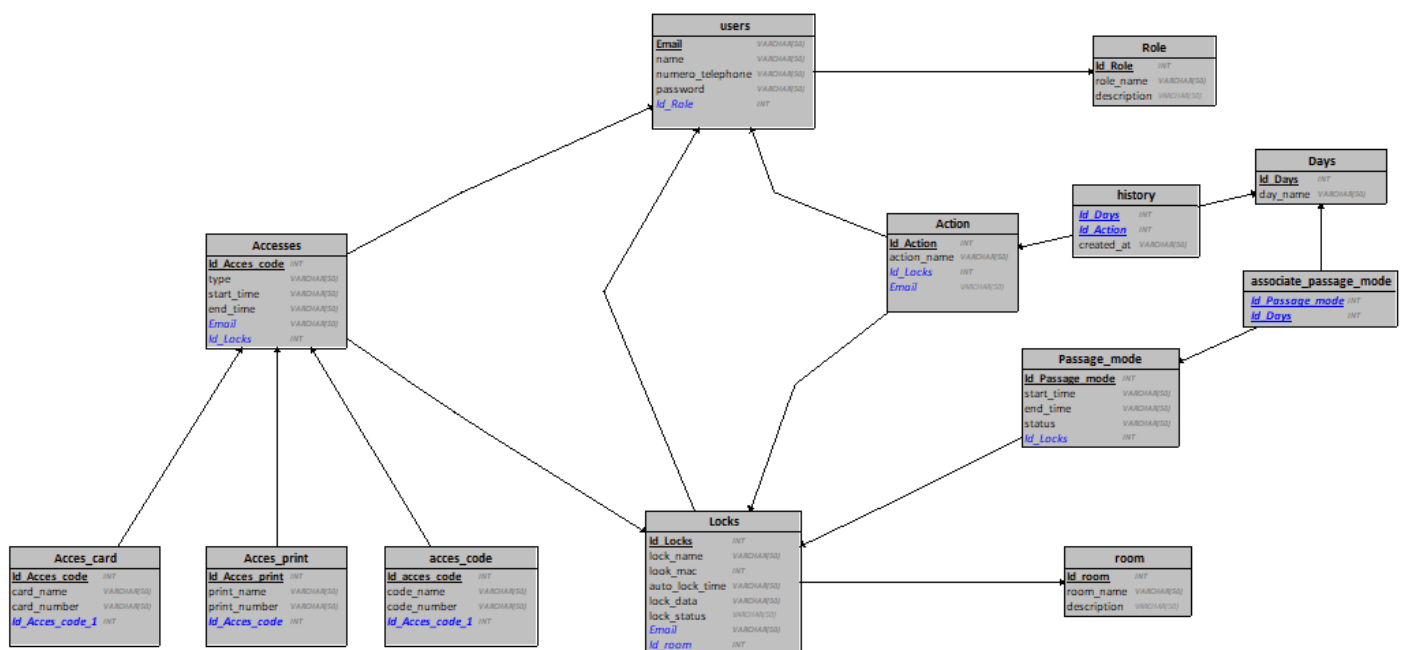


## V. Modèle de base de données

## 1. Modelé de conception des données



## 2. Modelé Logique des données





## VI. Conclusion

Dans cette partie, nous avons réalisé la modélisation organisationnelle et logique de notre application. Cette modélisation nous a permis de dégager le modèle logique des données qui sera exploité lors de l'implémentation. Ce modèle sera transformé en modèle physique de données qui fera l'objet du chapitre suivant (cahier d'implémentation ou réalisation).