 <p>ÉCOLE SUPÉRIEURE PRIVÉE DES SCIENCES APPLIQUÉES ET DE MANAGEMENT</p>	<p>REPUBLIQUE TUNISIENNE **** MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE <b>Département : Ingénierie Informatique</b></p>	
---	--	--

**RAPPORT**  
**Du**  
**Projet Personnel Encadré**

**Conception, la mise en place et test d'une  
application d'Internet of MobileThings**

**Élaboré par :**

M. Anas NAJJAR

M. Habib AROUA

Mme imen TRABELSI

**Encadré par :**

Dr. Fatma SOMAA

**Année Universitaire : 2018 / 2019**

# ***REMERCIEMENTS***

Premièrement nous remercions Dieu source de toute connaissance...

Nous tenons à présenter nos vifs remerciements et nos profondes gratitude à notre encadreur madame Fatma SOMAA pour sa pédagogie, sa compétence, sa modestie et son aide précieuse tout au long de ce projet.

# Sommaire

## Table des matières

Chapitre N°1 : Cadre du projet .....	1
Introduction.....	1
I. Définition de IoT .....	1
II. Étude et critique de l'existant .....	3
III. La solution proposée .....	3
Conclusion .....	4
Chapitre N°2 : Choix méthodologique et technologique .....	5
Introduction.....	5
I. Choix Méthodologique .....	5
I.1 Scrum.....	6
II. Planification .....	7
II.1 Définition des rôles.....	7
II.2 Modélisation avec UML.....	8
II.4 Les Diagrammes de UML .....	8
II.5 Choix technologique .....	9
Conclusion : .....	10
Chapitre 3 : Sprint 0.....	11
Analyse et Spécification des besoins.....	11
Introduction.....	11
I.Étude des besoins.....	11
I.1. Présentation des acteurs .....	11
I.3. Besoin non fonctionnel.....	12
II. Le diagramme de cas d'utilisation global .....	12
III. Le backlog de produit : .....	13
Conclusion .....	14
I. Introduction.....	15
II.Diagramme de classe : .....	15
III.Modèle relationnel : .....	15
IV.Le diagramme de déploiement .....	16
V.Les diagrammes cas d'utilisation détaillés de sprint 1.....	17
V.1. Les descriptions textuelles : .....	18
VI. Diagramme des séquences .....	19
IX. Les maquettes .....	22

Conclusion .....	24
Chapitre 5 : Sprint 2.....	25
Introduction.....	25
I.Les diagrammes cas d'utilisation détaillés de sprint 2 .....	25
II.Les descriptions textuelles : .....	28
III.Diagramme de séquence de sprint 2.....	28
IV.Les maquettes : .....	30
Conclusion .....	31

# Liste des figures

Figure 1 : L'architecture de L'IoT.....	1
Figure 2: L'architecture du Scrum .....	7
Figure 3: Diagramme de cas d'utilisation globale.....	13
Figure 4: Diagramme de classe .....	15
Figure 5: Le modèle relationnel .....	16
Figure 6 : Le diagramme de déploiement.....	16
Figure 7 : Diagramme cas d'utilisation gérer les utilisateurs.....	17
Figure 8 : Diagramme cas d'utilisation consulter les objets .....	17
Figure 9 : Diagramme cas d'utilisation consulter une carte réelle.....	18
Figure 10 : Diagramme de séquence s'authentifier.....	20
Figure 11 : Diagramme de séquence Ajouter un utilisateur .....	21
Figure 12 : Page d'accueil.....	22
Figure 13 : page d'authentification .....	22
Figure 14 : Page gestion des utilisateurs .....	23
Figure 15 : Page consulter tous les objets .....	23
Figure 16 : Page consulter une carte réelle.....	24
Figure 17 : Consulter les information de chaque objet.....	24
Figure 18 : Diagramme cas d'utilisation gérer le profil .....	26
Figure 19 : Diagramme cas d'utilisation consulter des statistiques .....	26
Figure 20 : Diagramme cas d'utilisation gérer les notifications .....	27
Figure 21 : Diagramme de séquence modifier mot de passe.....	29
Figure 22 : Page consulter le profil .....	30
Figure 23 : Interface pour modifier le mot de passe.....	30
Figure 24 : Interface pour modifier les informations .....	31
Figure 25 : Page de consulter les statistiques .....	31
Figure 26 : Page de gestion des notifications .....	31

# Introduction générale

La progression technologique a permis d'interconnecter plusieurs objets hétérogènes afin de répondre à un besoin bien spécifiques des utilisateurs.

Ses objets doivent être surveillés en permanence localement ou à distance.

Cependant, vu l'hétérogénéité des objets intelligents, il n'existe pas une application en open source et compatible avec les critères à surveiller par chaque utilisateur. En particulier, le cas des objets mobiles. En effet, la mise en œuvre de cette application suscite plusieurs problématiques. La communication avec des objets mobiles et distants. L'acquisition des informations et la synchronisation en temps réel et la prise de mesures nécessaires permettant d'éviter toute probabilité de blocage de l'application (crash)

Par conséquent, On nous a proposé de développer une application web en IoMT (Internet of Mobile Objects) qui offre des fonctionnalités supplémentaires par rapport à l'IoT (Internet of Things) et adaptables à la mobilité des objets. Ce projet, présente également, une opportunité inestimable pour l'enrichissement de nos connaissances en développement web (en Front End et en Back End) et met en valeur certaines pratiques utiles qu'on a apprise au cours de cette année.

Le travail sera résumé dans ce présent rapport composé de 5 Chapitres, le premier chapitre présentera le cadre de ce projet, le deuxième chapitre présentera les choix méthodologiques et technologiques dans ce projet et les raisons pour lesquelles nous avons fait ses choix, dans le troisième chapitre nous analyserons les besoins et détaillerons les spécifications de ce projet, et enfin dans le quatrième et cinquième chapitre nous détaillerons les sprints 1 et 2.

# Chapitre N°1 : Cadre du projet

## Introduction

Avant d'entrer dans les détails de notre projet, nous devons l'encadrer dans son contexte. Dans ce chapitre, nous commençons par l'étude de l'existant et la découverte de ses limites qui nous permettent de dégager notre solution à mettre en œuvre.

## I. Définition de IoT

Les environnements intelligents tels que les bâtiments intelligents, santé intelligente, réseau intelligent, et le transport intelligent font partie de notre quotidien à cause des avancements dans les technologies de capteurs, leur communication et leur inter connectivité aux infrastructures cyber-physiques avancées. Les capteurs interconnectés, placés dans des infrastructures cyber-physiques fixes, sont collectivement nommés Internet of Things (IoT). L'IoT représente des choses statiques interconnectés telles que les compteurs intelligents dans un réseau électrique intelligent, des capteurs intelligents dans des systèmes d'eau avancés, radio d'identification (RFID) et les capteurs de mouvement dans les bâtiments intelligents, ou les caméras de circulation dans aux intersections routières. En addition de l'IoT statique, la mobilité des objets prend sa part à travers les téléphones mobiles et les véhicules qui sont équipés de plus en plus capteurs avancés.

Les appareils mobiles avec leurs capteurs sont alors capables de communiquer les uns avec les autres, avec des infrastructures cyber-physiques et représentent l'Internet des objets mobiles.

L'architecture d'un système IoT est composée de plusieurs niveaux qui communiquent entre eux pour relier le monde tangible des objets au monde virtuel des réseaux et du cloud. Tous les projets n'adoptent pas une architecture formellement identique, néanmoins il est possible de schématiser le parcours de la donnée.

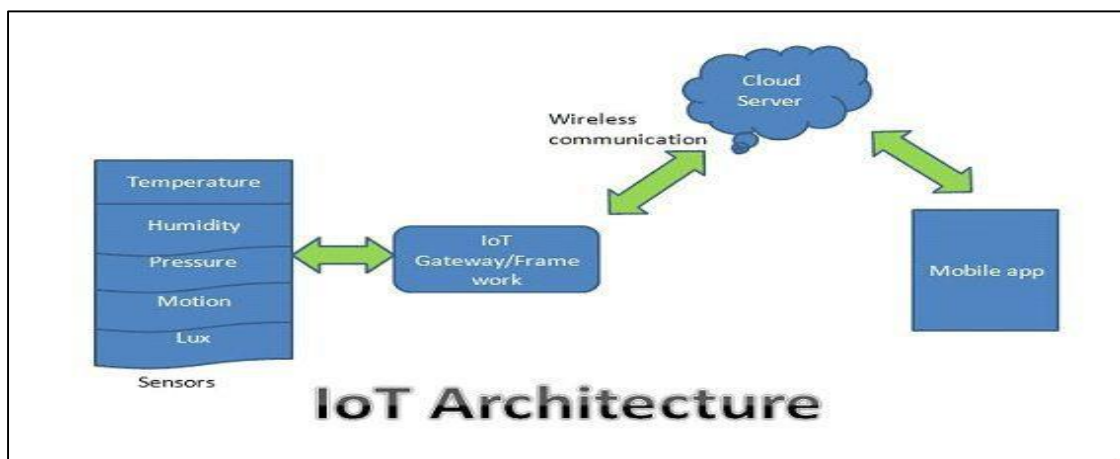


Figure 1 : L'architecture de L'IoT

Ses nœuds capteur peuvent être fixes donc nous parlons de IoT dans ce cas ou mobiles et nous parlons alors de IoMT

La différence entre l'IoT et l'IoMT est que lorsqu'on considère la mobilité des choses, des changements majeurs se produisent en termes de :

La différence entre l'IoT et l'IoMT est que lorsqu'on considère la mobilité des choses, des changements majeurs se produisent en termes de :

- Contexte, par exemple, où se trouve le périphérique mobile, dans quel emplacement il se trouve.
- Accès à Internet et connectivité, par exemple, si l'appareil mobile est connecté, et lorsqu'il est connecté à quel réseau sans fil ou câblé, à quel niveau de bande passante, et avec quelle sécurité.
- La disponibilité d'énergie, par exemple, ou l'appareil mobile peut se recharger ? combien d'énergie l'application mobile a-t-elle besoin ?
- La sécurité et la confidentialité, par exemple, quel type d'infrastructures de sécurité l'appareil mobile peut rencontrer lors de son déplacement entre les différents endroits ? et quelles informations privées les fournisseurs de services ont-ils à propos de l'utilisateur utilisant un dispositif mobile ?
- Par conséquent, lorsqu'on considère l'IoMT, la mobilité devient un objet de première classe et on doit regarder l'IoMT séparément de l'IoT. Il est important de noter que la mobilité des appareils tels que les téléphones mobiles et les véhicules a été étudié pendant de nombreuses années 1-5 en particulier la conception des appareils et leurs relations avec la mobilité et l'utilisation par les utilisateurs dans des environnements mobiles. Mais ce qui change maintenant est l'augmentation du nombre de capteurs par appareil mobile, l'augmentation de la densité des appareils mobiles dans les environnements des utilisateurs, et surtout l'inter-connectivité augmentée et la dépendance augmentée des utilisateurs aux appareils mobiles, rendant les appareils mobiles et leurs inter connectivité une partie de la routine quotidienne des utilisateurs et des environnements intelligents.



## **II. Étude et critique de l'existant**

Les exemples de l'IoT sont plusieurs, parmi lesquelles on peut citer :

- Amazon Echo - Smart Home : L'Amazon Echo fonctionne à travers l'assistante par voix Alexa de telle façon que les utilisateurs peuvent parler afin de performer certaines fonctions. Les utilisateurs peuvent dire à Alexa de jouer la musique, donner un rapport de météo, avoir les scores des sports etc. [ref]

L'Amazon Echo assure son rôle parfaitement mais il a une limite à noter et c'est la nécessité de donner les commandes vocales à courte distance par rapport au récepteur vocal et pas n'importe où dans la maison.

- Fitbit One - Objets portables : Le Fitbit One suit vos pas, les étages montés, les calories brûlées et la qualité du sommeil. L'appareil se synchronise également sans fil avec les ordinateurs et les smartphones afin de transmettre vos données de condition physique sous forme de graphiques compréhensibles pour surveiller vos progrès. [ref]

Malgré sa performance le Fitbit One est pourtant très cher pour quelques-uns et ça ne pourra pas fonctionner si ce n'est pas portable par l'utilisateur.

- AT&T - Voitures connectées : AT&T a ajouté 1.3 millions de voitures à son réseau en 2016, ce qui porte à 9,5 millions le nombre total de voitures connectées. Les conducteurs n'ont pas besoin de s'abonner ou de payer un abonnement mensuel pour les données afin que AT&T les compte comme abonnés.

Dans ce projet les voitures mobiles sont connectées et échangent les données avec le centre de la société à distance et en temps réel et ça donne une valeur ajoutée aux conducteurs. Néanmoins, ça n'offre pas plus de fonctionnalités à part de cela c'est à dire pas au point de contrôler les voitures connectées à distance, c'est juste un échange d'informations tout court.

## **III. La solution proposée**

Suite aux limites et contraintes dégagées des solutions existantes, nous devons créer une application qui permet de surveiller et contrôler à travers des commandes des nœuds mobiles en temps réel.

En effet, cette application nous aide à collecter les informations transmises par les objets via des capteurs intégrés qui seraient reçus par l'application en temps réel.

Il permet de connecter, analyser et intégrer en toute sécurité des données IoT en temps réel, à grande échelle, entre des terminaux connectés et vos applications d'entreprise. En plus, il gère tous les états liés à l'identité du point de terminaison, aux métadonnées et au cycle de vie de vos terminaux.

## **Conclusion**

Dans ce chapitre nous avons présenté le cadre général de notre projet en déterminant la problématique et en proposant une solution envisagée pour faire face à la situation courante.

Dans le chapitre suivant on va présenter la méthodologie et les technologies utilisés.

## **Chapitre N°2 : Choix méthodologique et technologique**

### **Introduction**

Nous avons permis de prendre connaissance et d'appliquer un ensemble de nouvelles méthodologies de travail auxquelles nous n'avions jamais été confrontés.

Avant d'aborder l'analyse et la conception de notre application, nous allons présenter notre choix des méthodologies de travail et des technologies nécessaires pour mener à terme de réalisation de notre application.

### **I. Choix Méthodologique**

Un projet informatique, quelle que soit sa taille et la portée de ses objectifs, nécessite la mise en place d'un planning organisationnel tout au long de son cycle de vie. C'est ainsi qu'est apparue la notion de méthode.

Une méthode, dans le contexte informatique, peut être définie comme une démarche fournissant une méthodologie et des notations standards qui aident à concevoir des logiciels de qualité.

Malgré la diversité des méthodes d'analyse et de conception, il est possible de les classer en trois catégories :

- ✓ Les méthodes cartésiennes ou fonctionnelles : SADT
- ✓ Les méthodes systémiques : MERISE, AXIAL.
- ✓ Les méthodes objet : OMT,
- ✓ Les méthodes agiles : SCRUM, etc.

Pour atteindre les objectifs, les méthodes agiles partagent pour la plupart un ensemble de technique,

L'une des particularités des méthodes agiles est de considérer le groupe projet comme une équipe plus qu'une somme de personnes. Les notions de rôles et de hiérarchie sont réduites à leur strict minimum et, c'est l'esprit de groupe qui est favorisé. Ce groupe doit partager un but commun : celui de réussir le projet

.

## **I.1 Scrum**

La méthode Scrum est une méthode agile, créée en 2002, dont le nom est un terme emprunté au rugby qui signifie « la mêlée ». Elle s'appuie sur le découpage des projets en itérations encore nommées « Sprints ». Un Sprint peut avoir une durée qui varie généralement entre deux semaines et un mois.

Scrum signifie mêlée au rugby. Scrum utilise les valeurs et l'esprit du rugby et les adapte aux projets de développement. Comme le pack lors d'un ballon porté au rugby, l'équipe chargée du développement travaille de façon collective, soudée vers un objectif précis. Comme un demi de mêlée, le Scrum Master aiguillonne les membres de l'équipe, les repositionne dans la bonne direction et donne le tempo pour assurer la réussite du projet.

Scrum est issu des travaux de deux des signataires du Manifeste Agile, Ken Schwaber et Jeff Sutherland, au début des années 1990. Il appartient à la famille des méthodologies itératives et incrémentales et repose sur les principes et les valeurs agiles.

Le plus souvent, les experts de Scrum, même ses fondateurs, le décrivent comme un cadre ou un patron de processus orienté gestion de projet et qui peut incorporer différentes méthodes ou I. Langage et méthodologie de conception

La méthodologie est une démarche organisée rationnellement pour aboutir à un résultat. Parmi les différentes méthodologies existantes, nous pouvons citer le modèle en cascade utilisée souvent dans les simples projets dont les besoins sont clairs et bien définis dès le début, le modèle en Y utiliser pour le développement des applications mobiles, ainsi que le processus unifié et les méthodologies agiles (Scrum & extrême programming) caractérisées par leurs souplesses et utilisées dans des grands projets.

Pour bien conduire notre projet et nous assurer du bon déroulement des différentes phases, nous avons opté Scrum comme une méthodologie de conception et de développement.

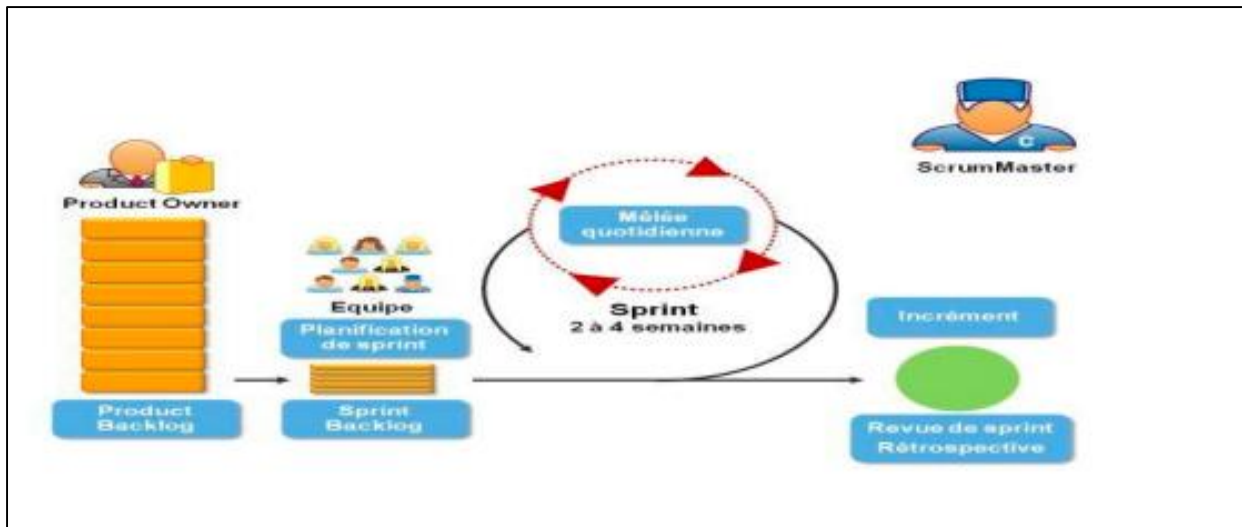


Figure 2: L'architecture du Scrum

Le choix de Scrum comme une méthodologie de pilotage pour notre projet s'est basé sur les atouts de ce dernier. Il se résumé comme suit :

- Plus de souplesse et de réactivité.
- La grande capacité d'adaptation au changement grâce à des itérations courtes.
- Et la chose plus importante, c'est que Scrum rassemble les deux cotés théorique et pratique et se rapproche beaucoup de la réalité.

## II. Planification

SCRUM est un processus de gestion du projet vise à organiser les exigences d'affaires, établir le coût et le calendrier précis du projet (y compris une liste des livrables et leurs dates de livraison), planifier l'organisation du travail et obtenir l'autorisation des gestionnaires. Dans SCRUM la planification se fait par niveau, chaque niveau correspondant à un Sprint. Une réunion des collaborateurs s'effectue généralement sur 8h et en deux temps.

Dans SCRUM l'équipe choisit, à partir du Backlog de produit, les éléments qu'elle s'engage à finir. Une fois le Backlog de Sprint est créé, les tâches sont identifiées et estimées (1-16 heures). Finalement on peut alors lancer le Sprint.

### II.1 Définition des rôles

Le tableau suivant présente les membres de SCRUM et leur rôle.

Le directeur du projet (SCRUM Master)	AROUAHabib
Le membre de l'équipe (SCRUM Team)	NAJJAR Anas, TRABELSIImen

## **II.2 Modélisation avec UML**

Vu l'importance cruciale de la modélisation dans le cycle de vie de n'importe quelle application, il fallait utiliser une méthode de modélisation qui s'adapte le mieux à nos besoins et à nos exigences qui sont entre autres : L'ouverture, la réutilisabilité, la modularité et l'extensibilité. Pour répondre à ces exigences, nous avons choisis de modéliser avec le langage de modélisation UML qui s'adapte parfaitement à la modélisation des applications à base d'objets et qui offre grâce à ses différents diagrammes une grande souplesse permettant la modélisation de différents aspects de l'application. Notre choix de ce langage se justifie aussi par le fait qu'UML est devenu un standard de modélisation adopté pour toutes les applications à aspect orienté objet.

## **II.3 Définition de UML**

UML(Unified Modeling Language ) : Se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier, concevoir des solutions et communiquer des points de vue. UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage, c'est pour ça qu'UML est présenté parfois comme une méthode alors qu'il ne l'est absolument pas. UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la définition des besoins jusqu'au codage.

## **II.4 Les Diagrammes de UML**

Ci-dessous une présentation rapide des différents diagrammes UML qui vont être utilisés tout au long du projet : Le diagramme des cas d'utilisation : Il représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est normalement utilisé lors des étapes de capture des besoins fonctionnels et techniques ; Le diagramme de classes : Sûrement l'un des diagrammes les plus importants dans un développement orienté objet. Sur la branche fonctionnelle, ce diagramme est prévu pour développer

La structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet ; Le diagramme de séquences : Il représente les échanges de messages entre objets, dans le cadre d'un fonctionnement particulier du système ; Le diagramme de composants : Il permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données, etc.).

## II.5 Choix technologique

L'environnement de développement est un terme qui désigne l'ensemble d'outils et de langage utilisé pour l'implémentation d'une solution informatique. Nous commençons par l'environnement matériel.

Les logiciels utilisés pour l'implémentation de notre solution sont les suivants :

- **WebStorm :**

Cette nouvelle version aide apporter de nombreuses fonctionnalités supplémentaires et des améliorations majeures, parmi lesquelles la refactorisation du composant Extract React, de nouvelles intégrations, de meilleurs workflows de test et l'expérience de débogage de Node.js, en plus c'est un IDE pour les langages Web (HTML, CSS et JavaScript), développé par l'entreprise JetBrains et basé sur la plateforme IntelliJIDEA.

L'éditeur offre diverses fonctionnalités :

- Éditeur JavaScript
  - Auto-complétion intelligente
  - Inspection du code et correction rapide
  - Navigation rapide du code et recherche d'usage d'un code
  - Refactorisation du code
- Débogueur et Tests

Débogueur JavaScript

Test unitaire JavaScript

Prise en charge de Node.js

- **Power designer :**

Cette solution nous aide à gérer des architectures d'informations complexes et à visualiser l'impact potentiel des nouvelles technologies avant leur mise en œuvre.

Power Designer est le nouvel outil de développement issu du portail Power BI et vous verrez que par bien des aspects, il va vous permettre de vous faciliter le développement de tableau de bord

C'est un logiciel de conception créé par la société SAP, qui permet de modéliser les traitements informatiques et leurs bases de données associées. [5]

- **Postman :**

Cette solution permet aux équipes de toutes tailles de collaborer de manière transparente en temps réel sur des espaces de travail et des collections partagés. Les espaces de travail de l'équipe Postman permettent aux équipes de rester organisées et de conserver une source de vérité unique tout au long du cycle de développement de l'API

Est un outil pour tester les API RESTful , C'est une application moderne est construite sur des APIs. C'est sur cette phrase d'accroche que nous découvrons le site de Postman, un outil qui permet de construire et de tester rapidement des requêtes HTTP.

- **SGBD MySql**

MySQL est un système de gestion de bases de données relationnelles.

## **Conclusion :**

Dans ce chapitre nous avons présenté présenter la méthodologie et les technologies utilisés.En déterminant la cause de notre choix Dans le chapitre suivant on va présenter la spécification de besoin.



## **Chapitre 3 : Sprint 0**

### **Analyse et Spécification des besoins**

#### **Introduction**

La spécification des besoins constitue une étape essentielle dans le cycle de développement de chaque système informatique, puisqu'elle permet de déterminer les fonctionnalités de l'application.

#### **I.Étude des besoins**

L'objectif de la phase de spécification des besoins consiste à définir avec détail l'ensemble des fonctionnalités offertes par le système ainsi que les acteurs. Les besoins dégagés ont été répartis en deux groupes fonctionnels et non fonctionnels.

##### **I.1. Présentation des acteurs**

Ils sont définis comme étant les utilisateurs directs de l'application. Dans le cadre de notre projet, nous distinguons deux acteurs principaux :

- **Admin** : il possède tous les privilèges d'accès à l'application. C'est celui qui va administrer notre plateforme.
- **Utilisateur** : il possède tous les privilèges d'accès à l'application sauf la gestion des utilisateurs (comptes utilisateurs). Il représente une personne qui a certains objets à localiser et surveiller.

##### **I.2. Besoin fonctionnel**

Notre application doit être capable de répondre aux besoins suivants

- S'authentifier
- Afficher une carte réelle
- Afficher des statistiques
- Gérer les notifications
- Gérer le profil
- Gérer les utilisateurs
- Consulter les objets

### I.3. Besoin non fonctionnel

- **Sécurité d'accès**
  - L'**authentification** pour un système informatique est un processus permettant au *système* de s'assurer de la *légitimité* de la demande d'accès faite par une entité (être humain ou un autre système...) afin d'autoriser l'accès de cette entité à des ressources du système qui concerne la Sécurité d'accès
- **Rapidité**
  - L'application doit répondre dans des délais acceptables aux requêtes des utilisateurs.
- **Ergonomique**
  - Le futur logiciel doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales, simples, ergonomiques et adaptées à l'utilisateur.
- **Aptitude à la maintenance et la réutilisation**
  - Le système doit être conforme à une architecture standard et claire permettant sa maintenance et sa réutilisation.

## II. Le diagramme de cas d'utilisation global

L'étude approfondie de la spécification des besoins permet de dégager plusieurs cas d'utilisation. Un cas d'utilisation décrit une utilisation du système par un acteur particulier. Ce qui revient à présenter les besoins fonctionnels de façon formelle.

Les cas d'utilisation permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Ils centrent l'expression des exigences du système sur ses utilisateurs.

Le diagramme de cas d'utilisation est un schéma qui montre les cas d'utilisations reliés par des associations à leurs acteurs.

Dans notre application nous avons réparti les cas d'utilisation selon les acteurs cités précédemment.

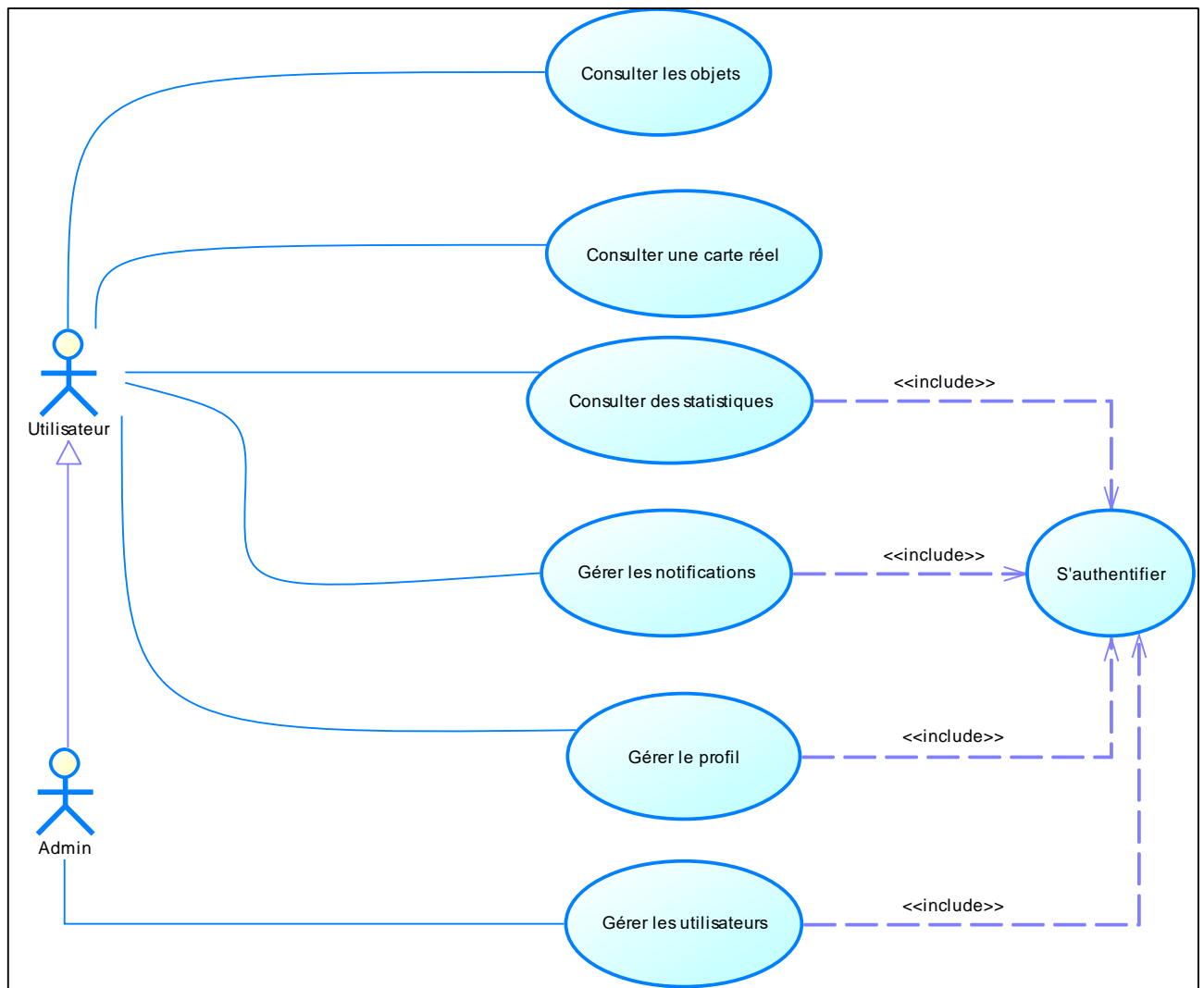


Figure 3: Diagramme de cas d'utilisation globale

### III. Le backlog de produit :

Le backlog du produit est l'artefact le plus important de Scrum, c'est l'ensemble de caractéristiques fonctionnelles ou techniques qui constituent le produit souhaité. Les caractéristiques fonctionnelles sont appelées des histoires utilisateur (user story) et les caractéristiques techniques sont appelées des histoires techniques (technical story).

Le Tableau 1 résume le backlog produit de notre application. Il est à noter que nous n'avons pas cité les histoires techniques comme la préparation de la maquette graphique, les travaux de conception et les jeux de tests, etc. Dans ce tableau chaque histoire utilisateur est caractérisée par un rang déduit à partir de ses risques et sa priorité expliqués dans la section II de ce même chapitre. Pour le traitement de nos histoires utilisateur nous choisissons de commencer avec les cas d'utilisation les plus prioritaires et ayant le risque le moins élevé.

ID	User Story	Priorité
1	En tant qu'administrateur je peux authentifier	1
2	En tant qu'administrateur je peux ajouter un utilisateur	2
3	En tant qu'administrateur je peux consulter la liste des utilisateurs	3
4	En tant qu'administrateur je peux supprimer un utilisateur	4
5	En tant qu'administrateur ou utilisateur je peux consulter la liste des objets	3
6	En tant qu'administrateur ou utilisateur je peux consulter la carte	3
7	En tant qu'administrateur ou utilisateur je peux modifier mes informations	3
8	En tant qu'administrateur ou utilisateur je peux modifier mon mot de passe	3
9	En tant qu'administrateur ou utilisateur je peux consulter le nombre des objets connecté en temps réel	4
10	En tant qu'administrateur ou utilisateur je peux consulter le taux des énergies	4
11	En tant qu'administrateur ou utilisateur je peux télécharger les chart graphiques	5
12	En tant qu'administrateur ou utilisateur je peux consulter les notifications	5
13	En tant qu'administrateur ou utilisateur je peux supprimer une notification	5

## Conclusion

Dans ce chapitre nous avons présenté la spécification de besoin. En déterminant les cas d'utilisation global et Le backlog de produit. **Dans** le chapitre suivant on va présenter le Sprint (1)

# Chapitre 4 : Sprint 1

## I. Introduction

La réunion de planification des Sprints est l'événement le plus important dans SCRUM. L'un des produits de cette réunion est le choix de la durée des Sprints et qui diffère selon la complexité du projet et la taille de l'équipe. Le but principal de Sprint (1) est de mettre en œuvre l'histoire utilisateur qui est présentée par le tableau de Backlog suivant :

ID	User Story	Priorité
1	En tant qu'administrateur je peux authentifier	1
2	En tant qu'administrateur je peux ajouter un utilisateur	2
3	En tant qu'administrateur je peux consulter la liste des utilisateurs	3
4	En tant qu'administrateur je peux supprimer un utilisateur	4
5	En tant qu'administrateur ou utilisateur je peux consulter la liste des objets	3
6	En tant qu'administrateur ou utilisateur je peux consulter la carte	3

## II. Diagramme de classe :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci.

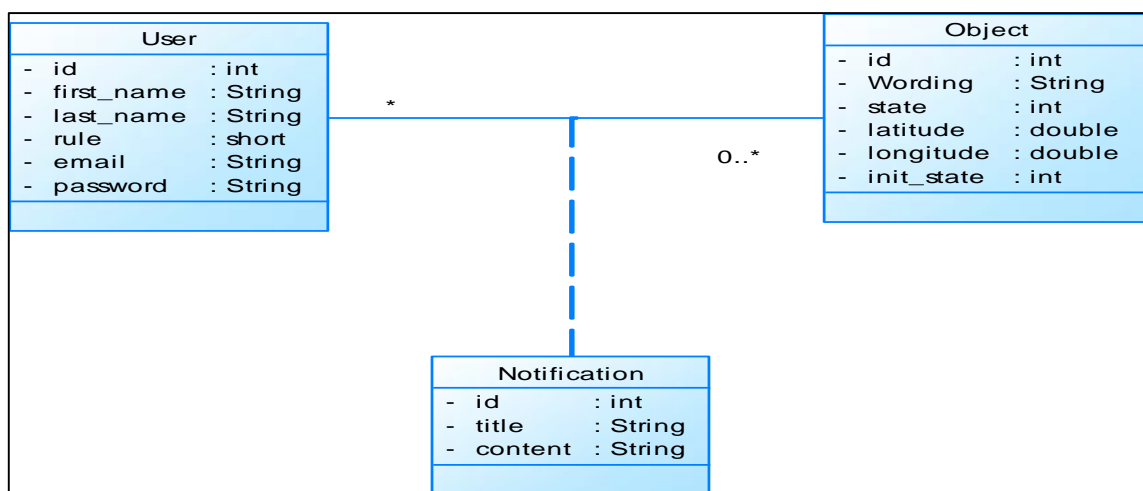


Figure 4: Diagramme de classe

## III. Modèle relationnel :

Le modèle relationnel est une manière de modéliser les relations existantes entre plusieurs informations, et de les ordonner entre elles. Cette modélisation qui repose sur des principes mathématiques mis en avant par E.F. Codd est souvent retranscrite physiquement (« implémentée ») dans une base de données.

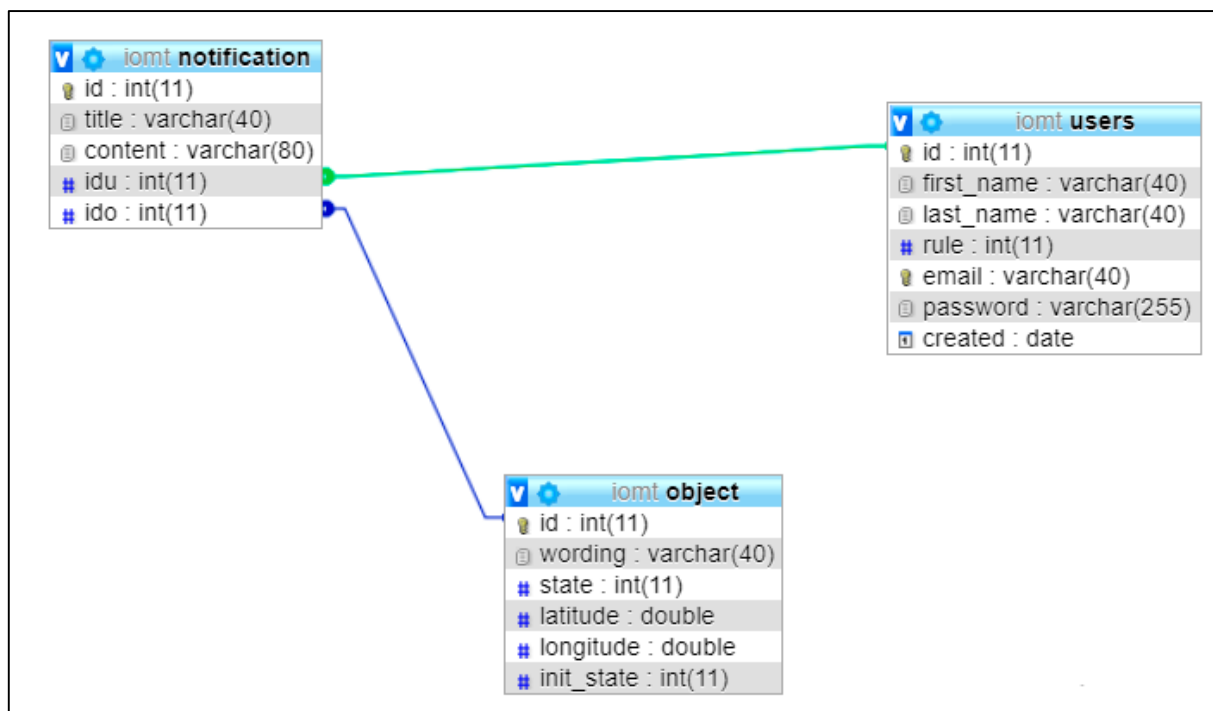


Figure 5: Le modèle relationnel

#### IV. Le diagramme de déploiement

Un diagramme de déploiement, présenté dans la figure 6, est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux.

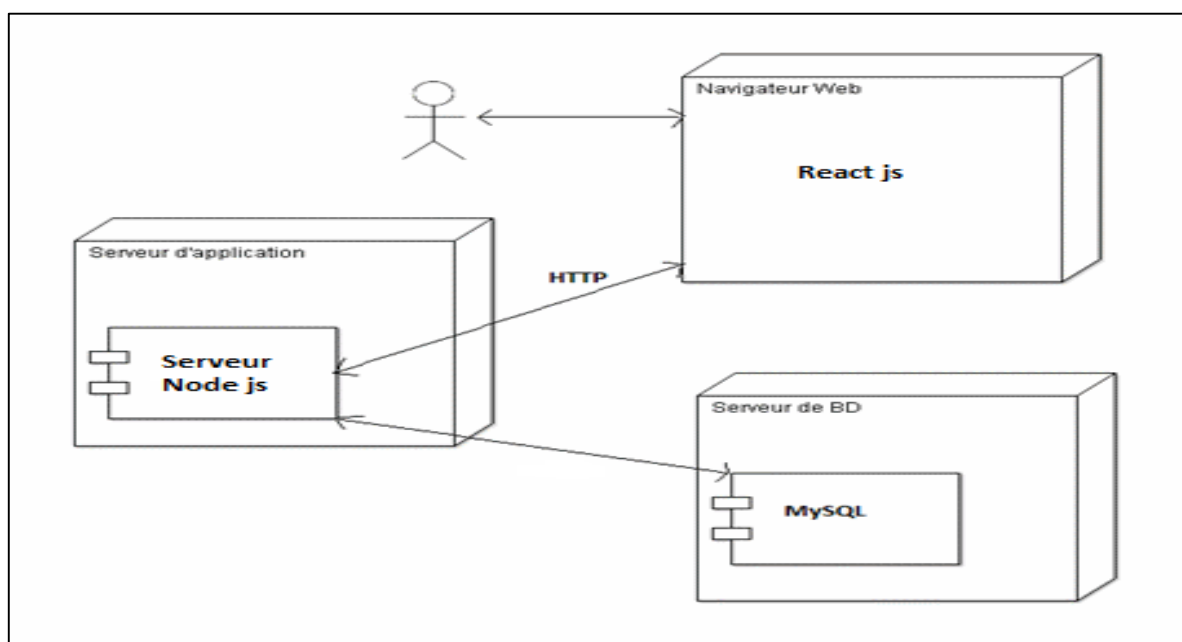


Figure 6 : Le diagramme de déploiement

## V. Les diagrammes cas d'utilisation détaillés de sprint 1

La figure N° 7 indique que l'administrateur, après être identifié, peut effectuer des opérations de gestion des utilisateurs tel que : Ajouter ou supprimer un utilisateur

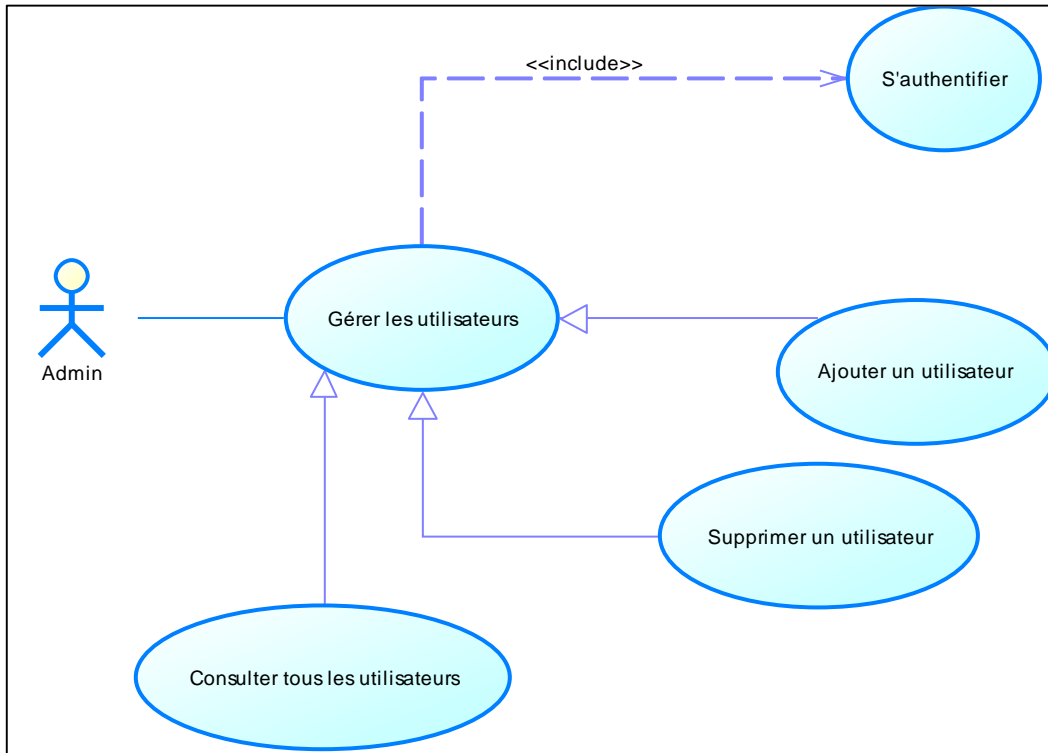


Figure 7 : Diagramme cas d'utilisation gérer les utilisateurs

La figure N°8 indique que l'administrateur ou l'utilisateur peut consulter les objets

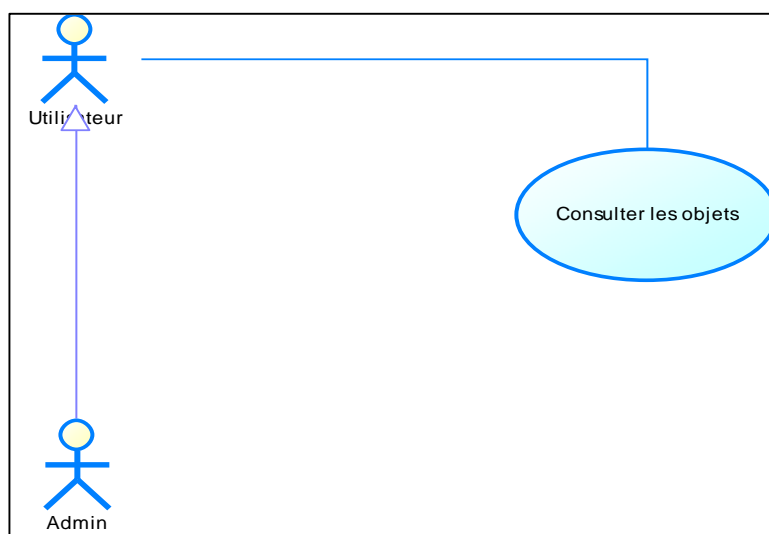


Figure 8 : Diagramme cas d'utilisation consulter les objets

La figure N° 9 indique que l'administrateur ou l'utilisateur peut consulter l'emplacement d'un ou plusieurs objets, consulter les informations de chaque objet et consulter le déplacement des objets sur la carte.

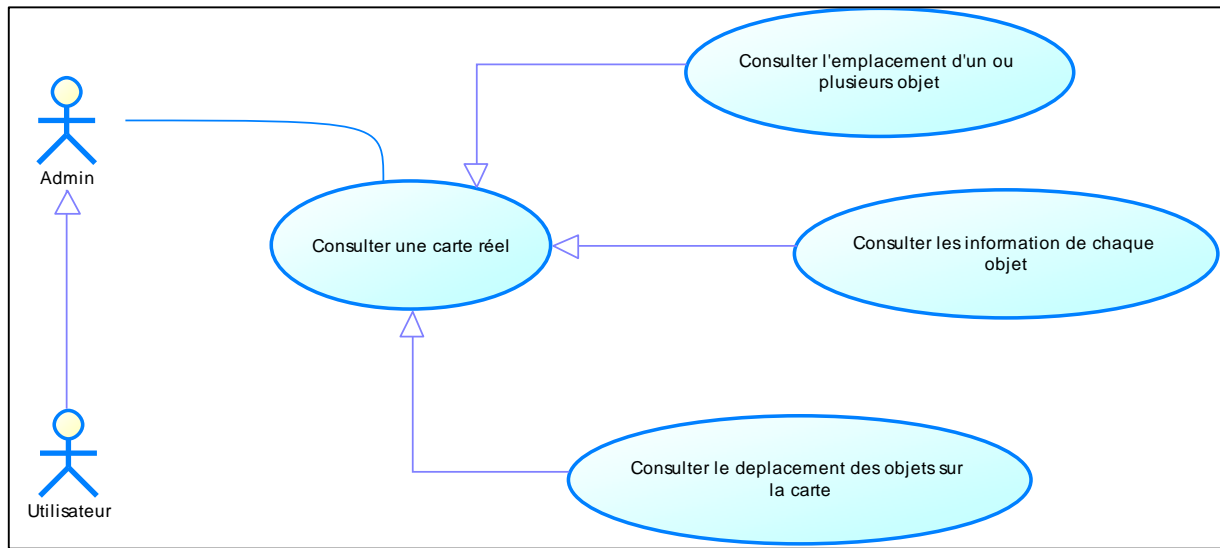


Figure 9 : Diagramme cas d'utilisation consulter une carte réelle

### V.1. Les descriptions textuelles :

La description détaillée du cas d'utilisation « s'authentifier » est présentée dans ci-dessous

**Titre :** s'authentifier

**Acteur :** Admin, Utilisateur

**Pré condition :** S'authentifier pour accéder au système.

**Enchaînement nominal :**

1. L'utilisateur demande le système d'afficher l'interface
2. Le système affiche l'interface
3. L'utilisateur saisit le login et mot de passe
4. L'utilisateur demande la validation des champs
5. Le système valide les champs
6. Le système retourne un message

**Enchaînement alternative :**

A1 : un utilisateur n'existe pas ou mot de passe non correcte ou champ vide

L'enchaînement A1 démarre au point 3 du scénario nominal. Le scénario nominal reprend au point 3.

**Post-condition(s) :** Admin ou utilisateur authentifié



La description détaillée du cas d'utilisation « Ajouter un utilisateur » est présentée dans ci-dessous

**Titre :** Ajouter un utilisateur

**Acteur :** Admin

**Pré condition :** Système opérationnel

**Enchaînement nominal :**

1. L'admin demande le système d'afficher l'interface
2. Le système affiche l'interface
3. L'admin saisit les champs
4. L'admin demande la validation des champs
5. Le système valide les champs
6. Le système retourne un message

**Enchaînement alternative :**

A1 : un utilisateur est existé

L'enchaînement A1 démarre au point 3 du scénario nominal. Le scénario nominal reprend au point 3.

A2 : champs vide

L'enchaînement A1 démarre au point 3 du scénario nominal. Le scénario nominal reprend au point 3.

**Post-condition(s) :**

Un utilisateur est ajouté

## **VI. Diagramme des séquences**

Un scénario décrit une exécution particulière d'un cas d'utilisation du début à la fin. Il représente un ensemble ordonné de messages échangés par les objets (instance de classe d'analyse ou instance d'acteur).

## S'authentifier

Pour accéder au système, l'utilisateur doit donner son login et son mot de passe sinon, si une information est erronée, l'accès lui sera refusé.

Afin de pouvoir accéder aux services de notre application, l'utilisateur doit tout d'abord s'authentifier en saisissant son login et son mot de passe. Si ces derniers sont corrects, le système retrouve l'utilisateur correspondant et lui ouvre son propre session. Sinon, le système lui demande de ressaisir son login et son mot de passe.

Ce scénario est décrit par le diagramme de séquence « Authentification » présenté dans la figure ci-dessous.

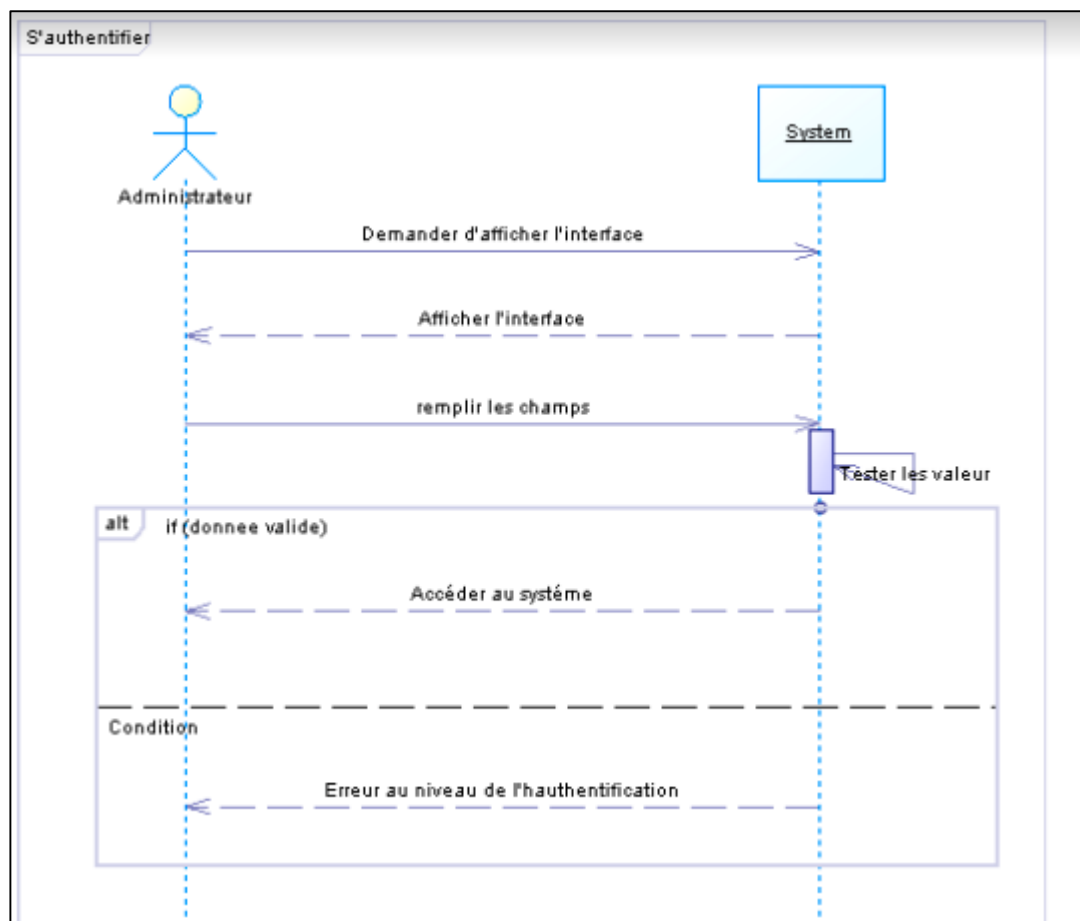


Figure 10 : Diagramme de séquence s'authentifier

## Ajouter utilisateur

Afin d'avoir le droit d'ajouter un utilisateur, l'admin doit s'authentifier tout d'abord pour pouvoir se connecter aux différentes interfaces. Une fois il est authentifié, il peut ajouter un utilisateur, donc il doit remplir tous les champs.

Voici le diagramme ci-dessous qui permet d'expliquer cette opération.

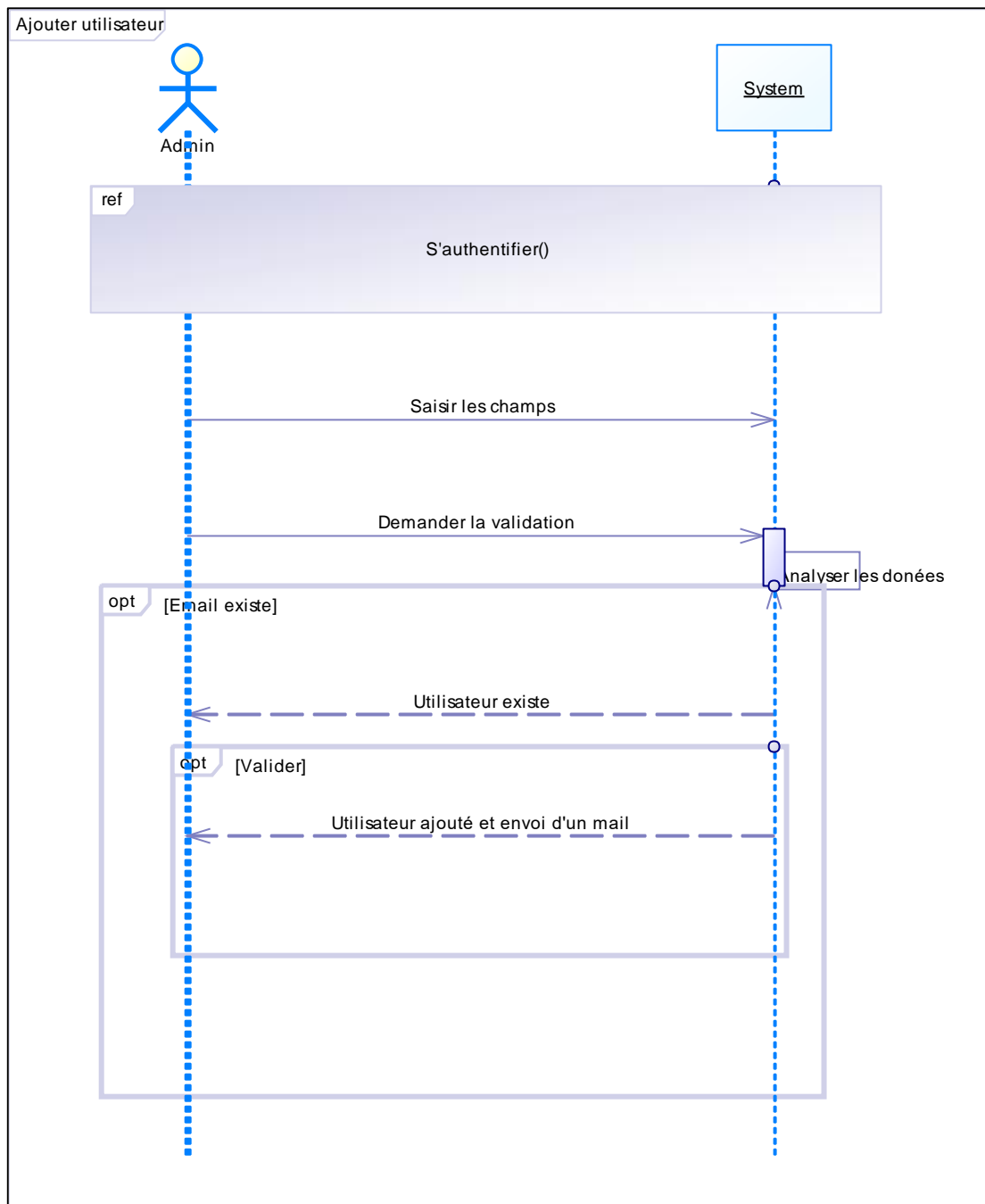
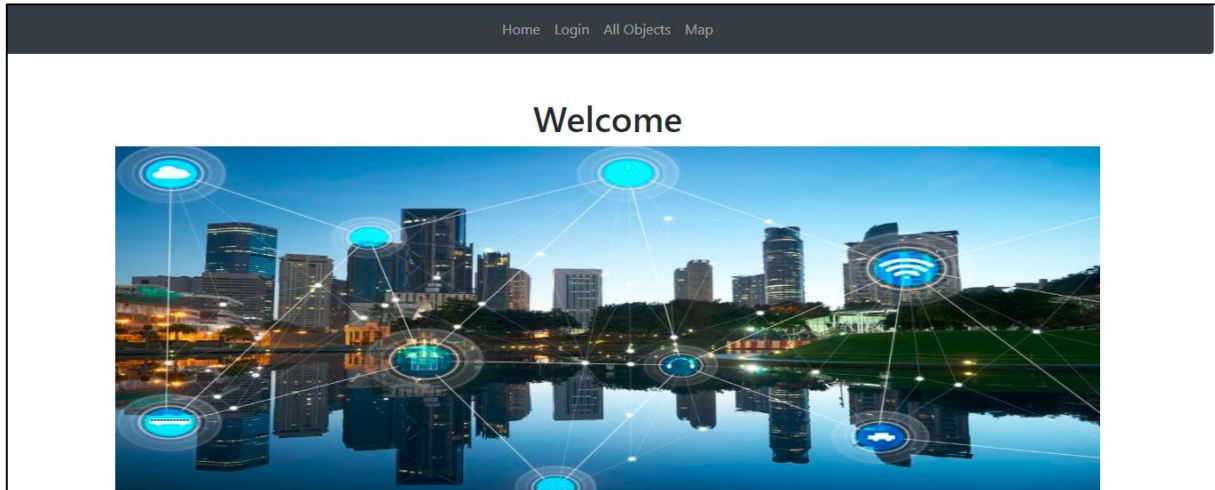


Figure 11 : Diagramme de séquence Ajouter un utilisateur

## IX. Les maquettes

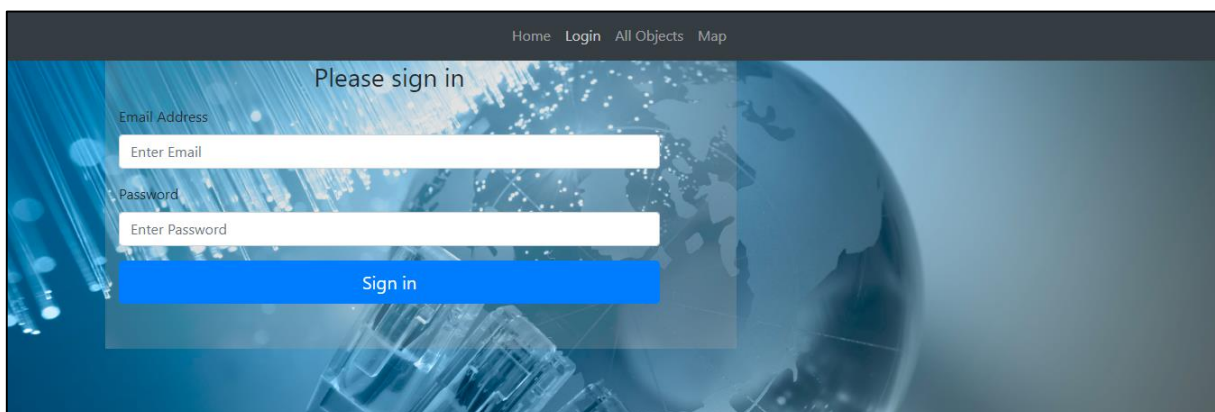
Voici les maquettes (Sprint 1) de notre application présentées dans les figures 12,13,14,15,16 et 17.

La figure 12 présente la page d'accueil de notre site



*Figure 12 : Page d'accueil*

Pour accéder à l'application, l'administrateur ou l'utilisateur doit tout d'abord s'authentifier, pour cela il introduit son email et son mot de passe comme l'indique la figure 13.



*Figure 13 : page d'authentification*

Parmi les fonctionnalités de notre application, l'administrateur peut gérer les utilisateurs.

Pour gérer les utilisateurs il peut ajouter ou supprimer un utilisateur et consulter la liste des utilisateurs. (Figure 14).

The screenshot shows a web application interface for user management. At the top, there is a navigation bar with links: Home, My Profile, User Management, Statistics, Notifications (2), Order, Map, and Logout. The main content area is divided into two sections. On the left, titled 'Add new user', there is a form with five input fields: 'First name' (placeholder: 'Enter your first name'), 'Last name' (placeholder: 'Enter your last name'), 'Email Address' (placeholder: 'Enter Email'), 'Password' (placeholder: 'Enter Password'), and 'Confirm your Password' (placeholder: 'Enter Password'). Below these fields are two buttons: 'Add new user' and 'Init all values'. On the right, titled 'List of users', there is a table with columns: 'Id', 'First Name', 'Last Name', 'Email', and 'Delete'. The table contains six rows of user data, each with a 'Delete' button.

Id	First Name	Last Name	Email	Delete
3	Anas	Najjar	anas.najjar@sesame.com.tn	Delete
6	Imen	Trabelsi	imen.trabelsi@sesame.com.tn	Delete
14	Fatma	Somaa	fatma.somaa@sesame.com.tn	Delete
16	Manel	Trabelsi	manel.trabelsi@sesame.com.tn	Delete
17	Mohamed Amine	Gharbi	amine.com@sesame.com	Delete
22	Laila	Yermani	lailayermani1930@gmail.com	Delete

Figure 14 : Page gestion des utilisateurs

La liste des objets est visible pour tout le monde comme l'indique la figure 15.

The screenshot shows a web application interface for viewing all objects. At the top, there is a navigation bar with links: Home, Login, All Objects, and Map. The main content area is titled 'All Objects' and contains a table with five columns: 'Id', 'Wording', 'State', 'Latitude', and 'Longitude'. The table contains five rows of object data.

Id	Wording	State	Latitude	Longitude
1	Server of Sesame	95	36.869287	10.196011
2	Machine of iot	50	36.8975331	10.1928377
3	Switch	66	36.8975331	10.1926389
5	Testor	30	36.8975331	10.1926489

Figure 15 : Page consulter tous les objets

La figure 16 indique l'emplacement de chaque objet dans la carte si on veut savoir les caractéristiques de chaque objet alors on clique à sa position.

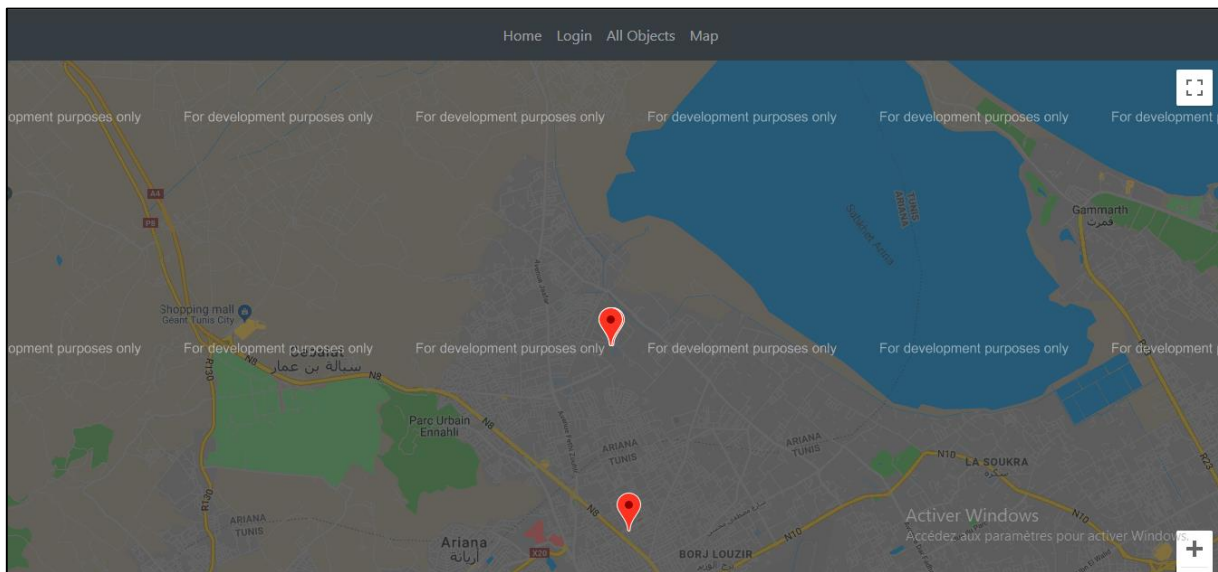


Figure 16 : Page consulter une carte réelle

Après le clique le système affiche les caractéristiques d'un objet comme indique la figure 17.

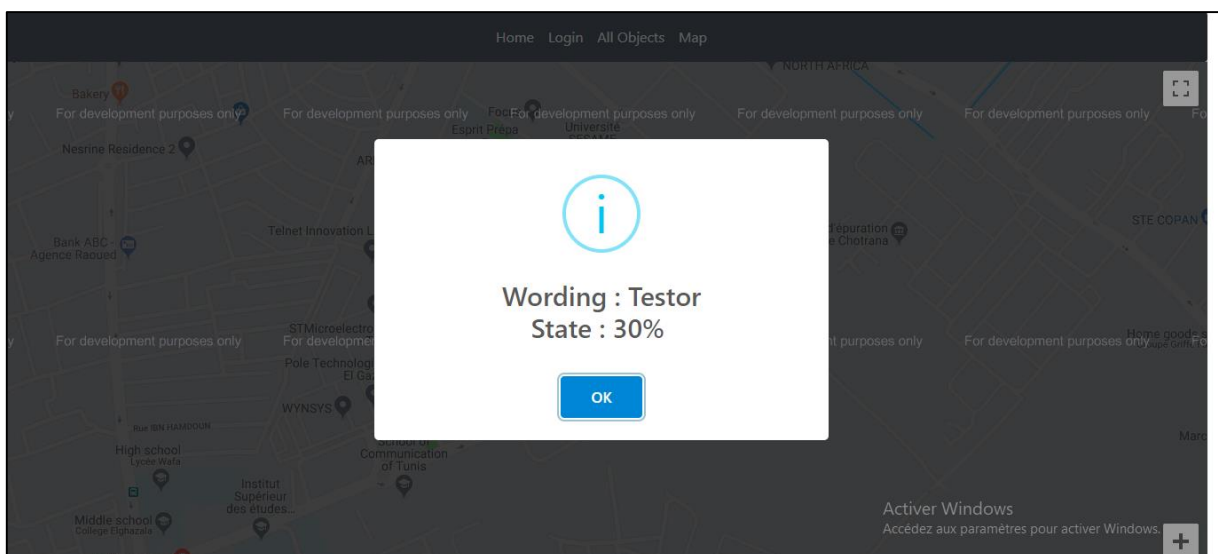


Figure 17 : Consulter les information de chaque objet

## Conclusion

Le résultat de sprint 1 est un produit potentiellement livrable. À la fin de ce chapitre, nous avons réussi à produire un incrément ayant suffisamment de valeur pour le client et pourra être utilisé dans un environnement de production.

Dans le chapitre qui suit, notre effort sera consacré pour produire un nouveau Sprint

## Chapitre 5 : Sprint 2

### Introduction

Au début de chaque sprint, on définit un objectif et on lui associe une liste de fonctionnalités qui constitueront le backlog de sprint. Au cours d'un sprint, l'objectif et la composition de l'équipe ne peuvent être modifiés. Ci-dessous la définition de l'objectif et des développeurs de ce sprint.

ID	User Story	Priorité
7	En tant qu'administrateur ou utilisateur je peux modifier mes informations	3
8	En tant qu'administrateur ou utilisateur je peux modifier mon mot de passe	3
9	En tant qu'administrateur ou utilisateur je peux consulter le nombre des objets connecté en temps réel	4
10	En tant qu'administrateur ou utilisateur je peux consulter le taux des énergies	4
11	En tant qu'administrateur ou utilisateur je peux télécharger les chart graphiques	5
12	En tant qu'administrateur ou utilisateur je peux consulter les notifications	5
13	En tant qu'administrateur ou utilisateur je peux supprimer une notification	5

### I. Les diagrammes cas d'utilisation détaillés de sprint 2

La figure N° 18 indique que l'administrateur ou l'utilisateur peut gérer son profil, tout d'abord il doit authentifier au système.

Pour gérer son profil il peut modifier ses informations ou bien son mot de passe.

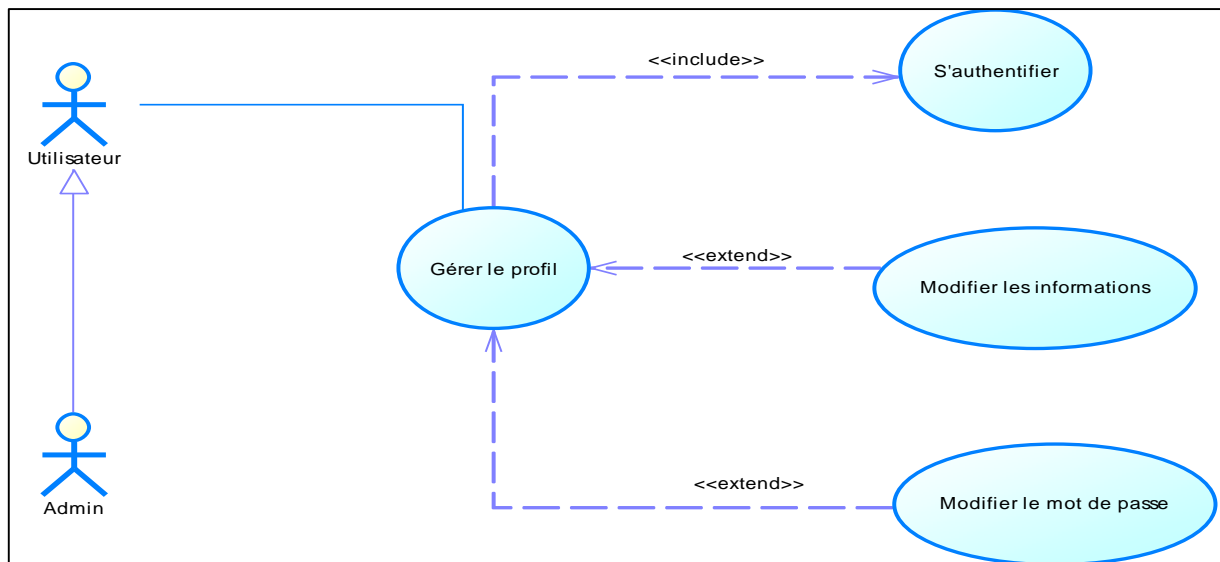


Figure 18 : Diagramme cas d'utilisation gérer le profil

La figure N° 19 indique que l'administrateur ou l'utilisateur peut consulter les statistiques, tout d'abord il doit authentifier au système.

Donc il peut consulter le nombre d'objet connecté au temps réel et consulter le taux d'énergie de chaque objet et enfin il peut télécharger les chart graphique soit en format .png soit en format .svg.

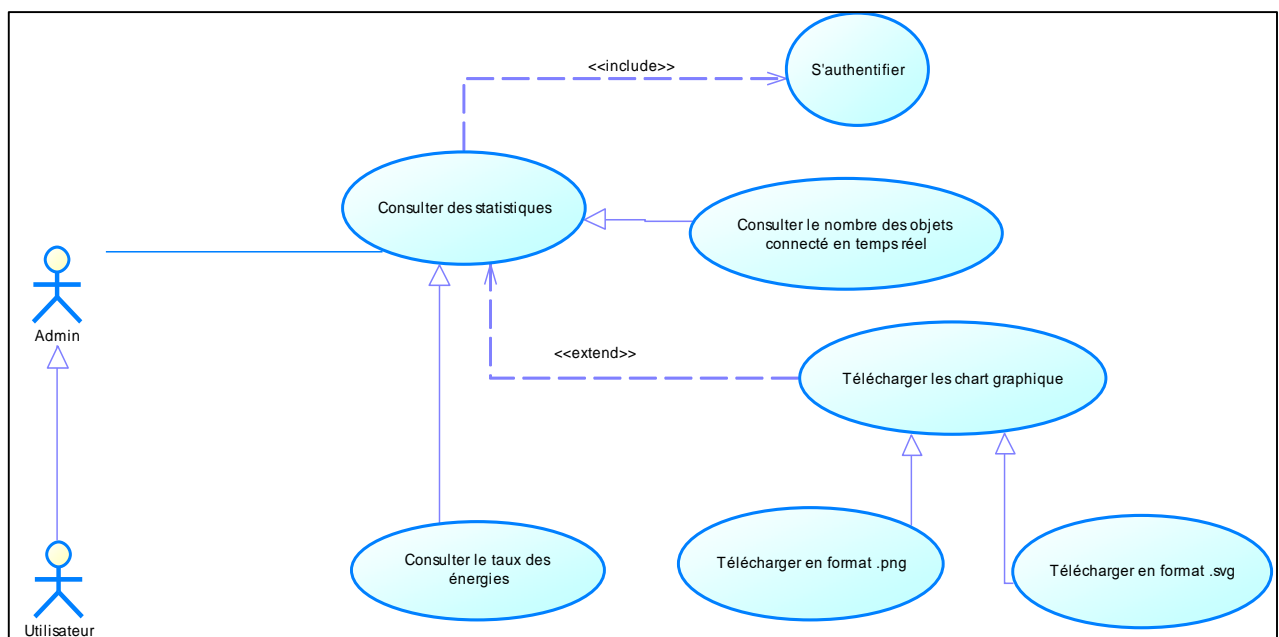


Figure 19 : Diagramme cas d'utilisation consulter des statistiques



La figure N° 20 indique que l'administrateur ou l'utilisateur peut gérer les notifications, tout d'abord il doit authentifier au système.

Donc il peut consulter les notifications ou supprimer une notification.

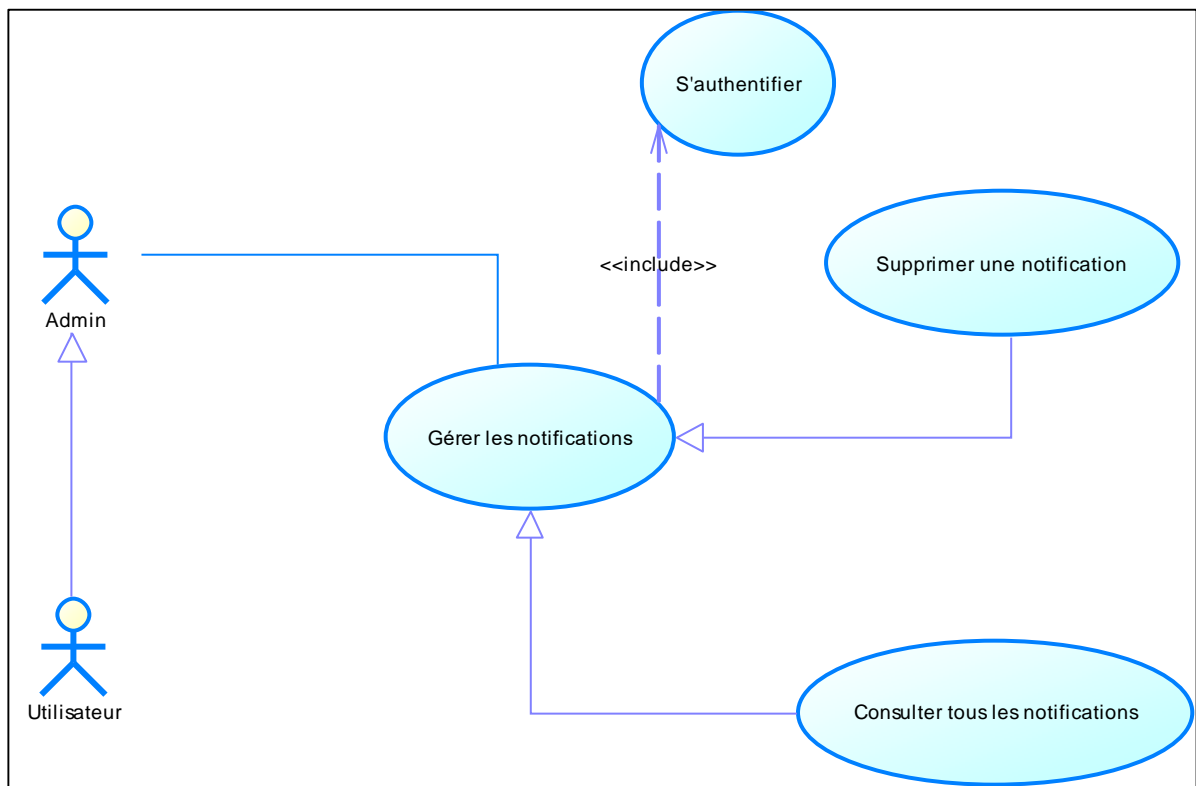


Figure 20 : Diagramme cas d'utilisation gérer les notifications

## **II. Les descriptions textuelles :**

La description détaillée du cas d'utilisation « Modifier le mot de passe » est présentée dans ci-dessous

**Titre :** s'authentifier

**Acteur :** Admin, Utilisateur

**Pré condition :** S'authentifier pour accéder au système.

**Enchaînement nominal :**

1. L'utilisateur demande le système d'afficher l'interface
2. Le système affiche l'interface
3. L'utilisateur saisit le l'ancien mot de passe
4. L'utilisateur saisit le nouveau mot de passe et le mot de passe de confirmation
5. L'utilisateur demande la validation des champs
6. Le système valide les champs
7. Le système retourne un message

**Enchaînement alternative :**

A1 : L'ancien mot de passe est incorrect

L'enchaînement A1 démarre au point 3 du scénario nominal. Le scénario nominal reprend au point 3.

A2 : Le nouveau mot de passe et le mot de passe de confirmation ne sont compatibles

L'enchaînement A2 démarre au point 4 du scénario nominal. Le scénario nominal reprend au point 4.

**Post-condition(s) :**

Mot de passe est modifié

## **III. Diagramme de séquence de sprint 2**

### **Modifier mot de passe**

Afin d'avoir le droit de modifier le mot de passe, l'utilisateur ou l'admin doit s'authentifier tout d'abord pour pouvoir se connecter aux différentes interfaces. Une fois il est authentifié, il peut manipuler ces interfaces. Pour modifier son mot de passe il doit saisir l'ancien mot de passe, nouveau mot de passe et mot de passe de confirmation.

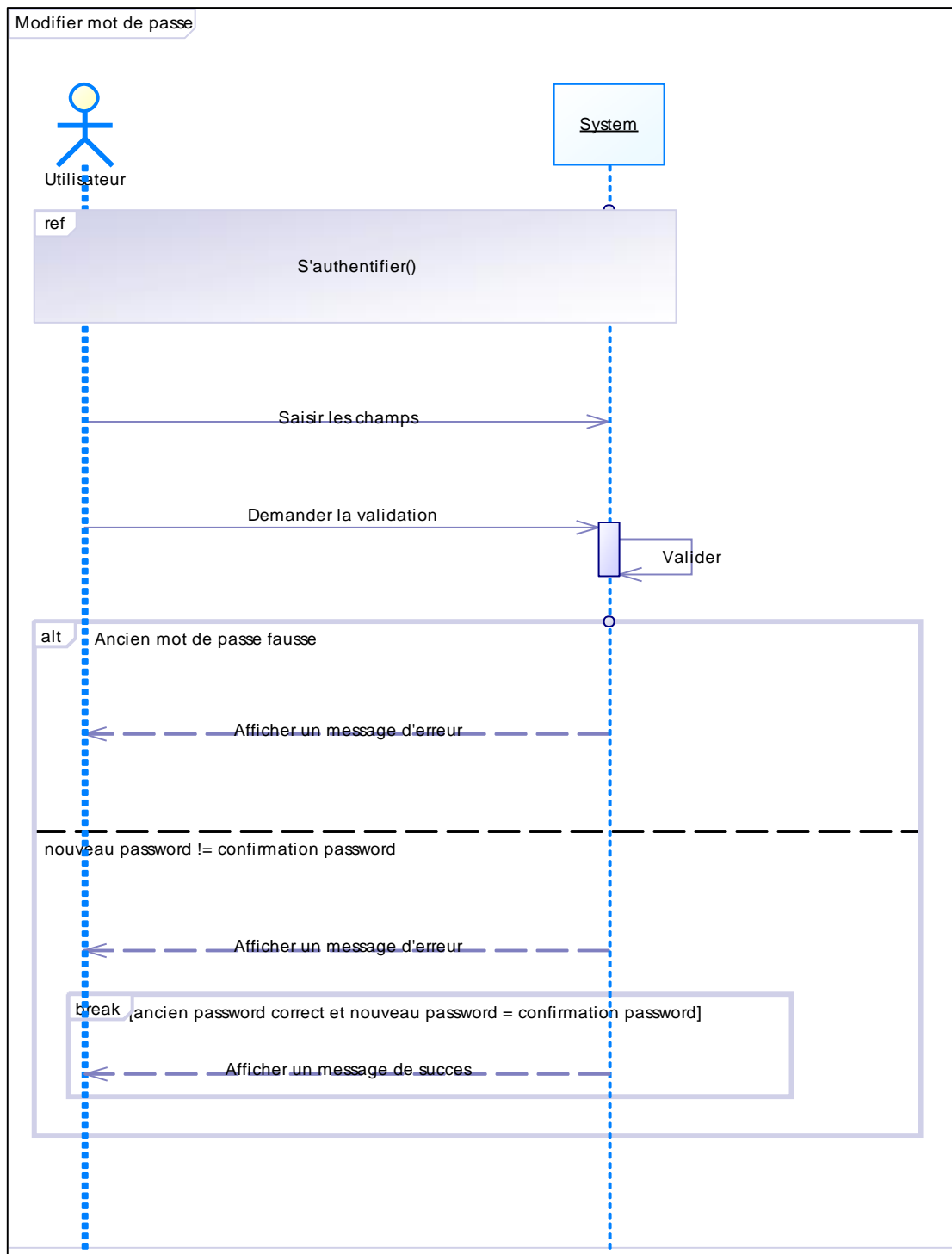


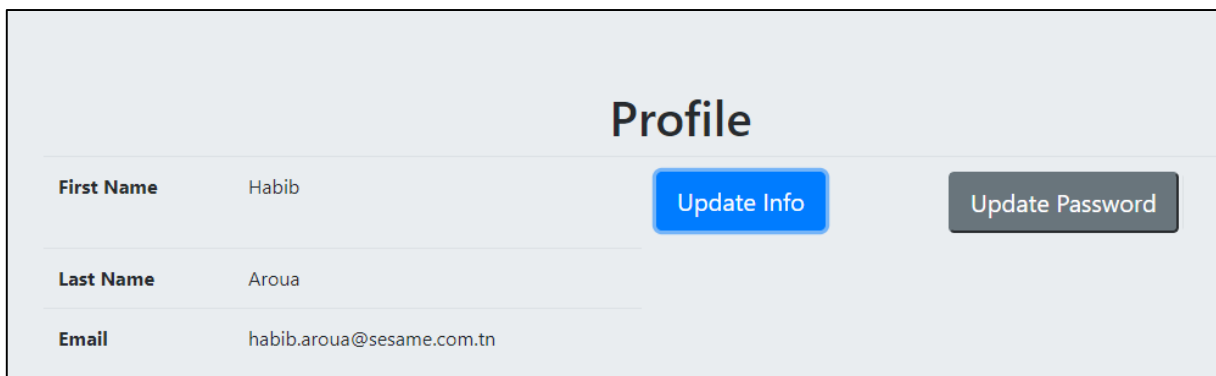
Figure 21 : Diagramme de séquence modifier mot de passe

#### IV. Les maquettes :

Voici les maquettes (Sprint 1) de notre application présentées dans les figures 22, 23, 24, 25, 26.

Parmi les fonctionnalités de notre application, un administrateur ou utilisateur peut modifier ses informations personnelles tel que ses info (nom et prénom) et son mot de passe.

- Pour modifier ses informations, il doit cliquer sur le bouton << Update Info >> ensuite le système affiche le formulaire de mise à jour de ses informations (Figure 24).
- Pour modifier son mot de passe, il doit cliquer sur le bouton << Update Password >> ensuite le système affiche le formulaire de mise à jour du mot de passe (Figure 23).

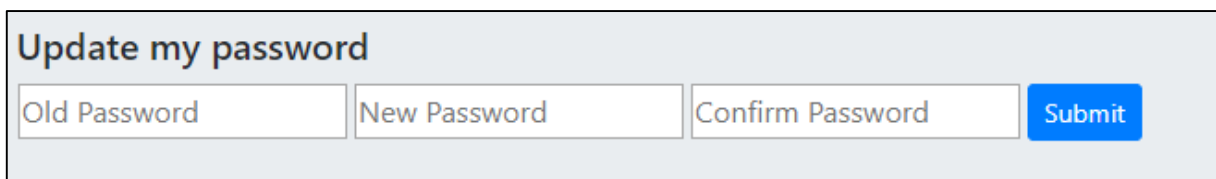


La maquette de la page 'Profile' est présentée dans un rectangle gris clair. Au centre, le titre 'Profile' est écrit en noir. En dessous, il y a trois lignes de texte : 'First Name' suivi de 'Habib', 'Last Name' suivi de 'Aroua', et 'Email' suivi de 'habib.aroua@sesame.com.tn'. À droite de ces informations, il y a deux boutons : un bouton bleu 'Update Info' et un bouton gris 'Update Password'.

Figure 22 : Page consulter le profil

Pour modifier le mot de passe l'administrateur ou l'utilisateur doit saisir l'ancien mot de passe puis il doit saisir le nouveau mot de passe et saisir encore le nouveau mot de passe pour le confirmer.

Comme indique la figure 23.



La maquette de l'interface pour modifier le mot de passe est présentée dans un rectangle gris clair. Au-dessus, le titre 'Update my password' est écrit en noir. En dessous, il y a trois champs de texte : 'Old Password', 'New Password' et 'Confirm Password'. À droite de ces champs, il y a un bouton bleu 'Submit'.

Figure 23 : Interface pour modifier le mot de passe

Par la suite, il introduit les nouvelles valeurs et valide les modifications effectuées. Comme indique la figure 24.

Figure 24 : Interface pour modifier les informations

L'administrateur ou l'utilisateur peut consulter les différentes statistiques, comme indique la figure 25.

À gauche, il peut consulter l'énergie de chaque objet.

À droite, il peut consulter le taux d'énergie de chaque objet.

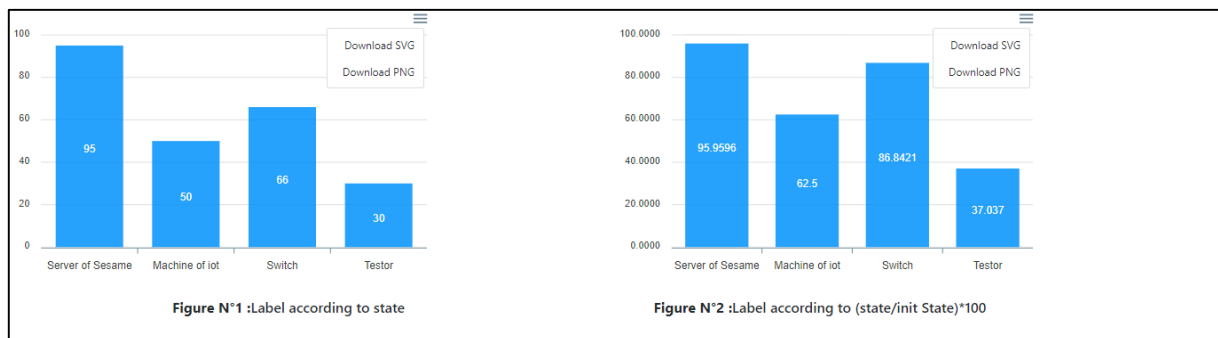


Figure 25 : Page de consulter les statistiques

L'administrateur ou l'utilisateur peut consulter les notifications des objets, il peut supprimer une notification

Home My Profile User Management Statistics Notifications <b>1)</b> Order Map Logout			
Id	Title	Content	Delete
1	problem 1	text of the problem 1	Delete

Figure 26 : Page de gestion des notifications

## Conclusion

Nous avons réussi donc à développer le sprint 2 de notre application et terminer l'application et on a développé une application fonctionnelle.

## Conclusion générale

Dans ce rapport, nous avons commencé par parler du cadre de notre projet qui est le développement d'une application en IoT et on a dégagé les particularités de notre projet suite à un comparatif fait par rapport à des projets existants. Nous avons ensuite donné les raisons de nos choix technologiques et méthodologiques et aussi nous avons analysé les besoins et les spécifications. Et enfin, nous avons détaillé les spécifications de notre projet à travers les sprints.

Durant ce projet, nous avons rencontré des grands défis , surtout en ce qui concerne quelques le développement du map dynamique et des notifications et aussi le développement du mécanisme des commandes vis à vis avec les objets connectés en addition du développement des autres fonctionnalités tels que les différentes gestions des utilisateurs et des objets etc. mais nous avons su trouvé des solutions à toutes ses problèmes , et en même temps nous avons appris beaucoup de choses dans le développement web basé sur javascript en Node JS (le framework Express js) et ReactJS comme c'est la première fois que nous développons à l'aide de ceux-ci.

En addition, nous avons envisagé des améliorations dans cette application telles que l'ajout de la fonctionnalité qui permet de donner des commandes vocales au lieu des commandes écrites et aussi avoir la trajectoire des objets connectés sur la carte et pas juste leurs positions en temps réel et développer une version mobile (Application Android et IOS).

## Références

- [1][IOT] <https://www.digora.com/fr/blog/definition-iot-et-strategie-iot> [le 10/04/2019]
- [2][UML] <https://openclassrooms.com/fr/courses/2035826-debutez-lanalyse-logicielle-avec-uml/2035851-uml-c-est-quoi> [le 15/04/2019]
- [3][WebStorm] <https://fr.wikipedia.org/wiki/WebStorm> [le 02/05/2019]
- [4][PostMan] <https://openclassrooms.com/fr/courses/4668056-construisez-des-microservices/5123020-testez-votre-api-grace-a-postman> [le 02/05/2019]
- [5][MySql] <https://fr.wikipedia.org/wiki/MySQL> [le 07/02/2019]
- [6][Diagramme] [https://fr.wikipedia.org/wiki/UML\\_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique)) [le 16/05/2019]
- [7][Deploiement] [https://fr.wikipedia.org/wiki/Diagramme\\_de\\_d%C3%A9ploiement](https://fr.wikipedia.org/wiki/Diagramme_de_d%C3%A9ploiement) [le 16/05/2019]