

RAPPORT DE PROJET DE FIN D'ETUDES

Présenté en vue de l'obtention de la
LICENCE FONDAMENTALE EN SCIENCES ET TECHNOLOGIES

Mention : Sciences de l'Informatique
Spécialité : Sciences de l'Informatique

Intitulé du projet

Par
PRÉNOM NOM

Réalisé au sein de Tunisie Télécom



Soutenu publiquement le ? mai 2019 devant le jury composé de :

Président : Prénom NOM, University Relations Leader, IBM
Rapporteur : Prénom NOM, Enseignant, ISTIC
Examinateur : Prénom NOM, Enseignant, ISTIC
Encadrant professionnel : Prénom NOM, Ingénieur, VERMEG
Encadrant académique : Prénom NOM, Enseignant, ISTIC

Année Universitaire : 2018-2019

RAPPORT DE PROJET DE FIN D'ETUDES

Présenté en vue de l'obtention de la
LICENCE FONDAMENTALE EN SCIENCES ET TECHNOLOGIES

Mention : Sciences de l'Informatique
Spécialité : Sciences de l'Informatique

Intitulé du projet

Par
PRÉNOM NOM

Réalisé au sein de Tunisie Télécom



Autorisation de dépôt du rapport de Projet de Fin d'Etudes :

Encadrant professionnel :

Le :

Signature :

Encadrant académique :

Le :

Signature :

Dédicaces

A mes parents,
A mes frères et sœurs,
A mes amis

Remerciements

Je tiens à remercier

Table des matières

Dédicaces	i
Remerciements	ii
Introduction Générale	1
1 Cadre général du projet	2
1.1 Introduction	2
1.2 Présentation de l'organisme d'accueil	2
1.2.1 Présentation de l'organisme d'accueil	2
1.2.2 Solutions	2
1.2.3 Domaines et compétences	3
1.2.4 Collaborateurs	3
1.2.5 Certifications	3
1.2.6 Organigramme	3
1.3 Cadre du projet	4
1.3.1 Contexte du projet	4
1.3.2 Problématique	4
1.3.3 Etude de l'existant	5
1.4 Etat de l'art	6
1.4.1 Analyse et critique des solutions existantes dans le marché	6
1.4.2 Virtualisation	8
1.4.3 Cloud Computing	8
1.4.4 Google Cloud plateform	10
1.5 Méthodologie de gestion de travail	10
1.5.1 Méthodes Agiles	11
1.5.2 Présentation de la méthodologie Scrum	11
1.6 Conclusion	12
2 Sprint0 : Analyse et conception globales	13
2.1 Introduction	13
2.2 Identification des acteurs	13
2.3 Identification des besoins	13
2.3.1 Besoins fonctionnels	13
2.3.2 Besoins non fonctionnels	14
2.4 Étude conceptuelle	14
2.4.1 Diagramme de cas d'utilisation global	14
2.5 Étude conceptuelle globale	15
2.5.1 Diagramme de classe global	15

2.6	Spécificités techniques	16
2.6.1	Backlog produit	16
2.6.2	Planification du projet	18
2.6.3	Environnement de travail	19
2.7	Architecture de l'application	21
2.7.1	Architecture physique de l'application	21
2.7.2	Architecture logique de l'application	22
2.8	Conclusion	23
3	Sprint1 : Gestion des comptes et des utilisateurs	24
3.1	Introduction	24
3.2	Étude fonctionnelle	24
3.2.1	Backlog du sprint	24
3.2.2	Diagramme de cas d'utilisation du sprint 1	26
3.3	Gestion des comptes	27
3.3.1	Analyse	27
3.3.2	Conception	29
3.4	Gestion des utilisateurs	33
3.4.1	Analyse	33
3.4.2	Conception	34
3.5	Réalisation	36
3.5.1	Sécurité	36
3.5.2	Interfaces	37
3.6	Conclusion	39
4	Sprint2 : Management des projets et des VMs	40
4.1	Introduction	40
4.2	Etude fonctionnelle	40
4.2.1	Backlog du Sprint	40
4.2.2	Diagramme de cas d'utilisation du sprint 2	42
4.3	Management des projets	43
4.3.1	Analyse	43
4.3.2	Conception	45
4.4	Management des VMs	47
4.4.1	Analyse	47
4.4.2	Conception	48
4.5	Réalisation	51
4.5.1	Interfaces	51
4.6	Conclusion	53
5	Sprint3 : Gestion des planifications et logs	54
5.1	Introduction	54
5.2	Etude fonctionnelle	54
5.2.1	Backlog du Sprint	54
5.2.2	Diagramme de cas d'utilisation du sprint 3	55
5.3	Gestion des planifications	56
5.3.1	Analyse	56
5.3.2	Conception	56
5.4	Gestion des logs	59

5.4.1	Analyse	59
5.4.2	Conception	59
5.5	Réalisation	60
5.5.1	Interfaces	60
5.6	Conclusion	61
Bibliographie		63
Annexe 1, Les candidats classés par ordre alphabétique		64

Table des figures

1.1	Les collaborateurs de TRITUX	3
1.2	Organigramme de TRITUX	4
1.3	Les services du Cloud Computing	9
2.1	Diagramme de cas d'utilisation global	15
2.2	Diagramme de classe global	16
2.3	Découpage du projet	18
2.4	Logo Nodejs	21
2.5	Logo Git	21
2.6	Logo Postman	21
2.7	Architecture physique de l'application	21
2.8	Architecture logique Front-end de l'application	22
2.9	Architecture logique Back-end de l'application	23
3.1	Diagramme de cas d'utilisation du sprint 1	26
3.2	Raffinement du cas d'utilisation : S'authentifier	27
3.3	Raffinement du cas d'utilisation : Gérer le profil	28
3.4	Diagramme de séquence : S'authentifier	30
3.5	Diagramme de séquence : Réinitialiser mot de passe	31
3.6	Diagramme de séquence : Mettre à jour profil	32
3.7	Raffinement du cas d'utilisation : Gérer les utilisateurs	33
3.8	Diagramme de séquence : Supprimer un utilisateur	35
3.9	Structure de JWT	36
3.10	Fonctionnement de JWT	37
3.11	Interface de connexion à l'application	37
3.12	Interface d'inscription	37
3.13	Interface de réinitialisation du mot de passe	38
3.14	Interface d'affectation d'un nouveau mot de passe en cas d'oubli	38
3.15	Email de réinitialisation du mot de passe	38
3.16	Interface de mise à jour du profil	38
3.17	Interface de gestion des utilisateurs	39
4.1	Diagramme de cas d'utilisation du sprint 2	42
4.2	Raffinement du cas d'utilisation : Gérer les membres du projet	43
4.3	Raffinement du cas d'utilisation : Gérer les projets	44
4.4	Diagramme de séquence : Mettre à jour le budget restant du projet en temps réel	46
4.5	Raffinement du cas d'utilisation : Orchestrer le fonctionnement des VMs	47
4.6	Diagramme de séquence : Créer une nouvelle instance de VM	49

4.7	Diagramme de séquence : Modifier le projet d'une machine virtuelle	50
4.8	Interface de gestion des projets	51
4.9	Interface de création d'un nouveau projet	51
4.10	Interface de gestion des membres du projet	52
4.11	Interface de gestion des instances de machine virtuelle	52
4.12	Interface de création d'une nouvelle instance de machine virtuelle	53
4.13	Interface de détails des instances de machine virtuelle	53
5.1	Diagramme de cas d'utilisation du sprint 3	55
5.2	Raffinement du cas d'utilisation : Gérer les planifications	56
5.3	Diagramme de séquence : Mettre à jour l'état d'une machine virtuelle automatiquement	57
5.4	Diagramme de séquence : Créer un planificateur	58
5.5	Diagramme de séquence : Consulter les logs	59
5.6	Interface de gestion des planifications	60
5.7	Interface de création d'un planning	60
5.8	Interface de consultation des logs	61

Liste des tableaux

1.1	Etude comparative des solutions existantes dans le marché	7
2.1	Backlog produit	18
2.2	Comparaison des Frameworks les plus populaires du Front-end	20
3.1	Backlog du Sprint 1	26
3.2	Description textuelle du « Créer un compte »	28
3.3	Description textuelle du « Mettre à jour le profil »	28
3.4	Description textuelle du « Supprimer un utilisateur »	34
4.1	Backlog du Sprint 2	42
4.2	Description textuelle du « Ajouter un membre au projet »	44
4.3	Description textuelle du « Créer un nouveau projet »	45
4.4	Description textuelle du « Activer une machine virtuelle »	48
4.5	Description textuelle du « Lister les instances de VM »	48
5.1	Backlog du Sprint 3	55
5.2	Description textuelle du « Supprimer un planning »	57
5.3	Description textuelle du « Consulter les logs »	59

Introduction Générale

Les solutions technologiques autour des infrastructures distribuées n'arrêtent pas de progresser, les sociétés consommatrices deviennent de plus en plus nombreux, ils sont aujourd’hui au centre de tous ces changements.

Actuellement, une nouvelle tendance a fait son apparition dans le monde des technologies de l’information. Il s’agit du «Cloud computing». Ce modèle a récemment émergé comme un nouveau paradigme pour l’hébergement et la prestation de services sur Internet. Le Cloud computing est le nouveau pas dans l’évolution des services web et des produits informatiques à la demande.

A ce niveau, le Cloud computing se déplace comme un remède offrant une architecture distante dont la gestion est garantie par une tierce partie. Le fournisseur de cette architecture garantit par conséquent l’utilisation et le suivi des services à travers des plateformes tel que Google Cloud Plateform pour le fournisseur Google.

Bien que le genre de ces plateformes semble idéal pour la gestion des ressources Cloud. Cependant, ils ne peuvent pas être personnalisées au besoin de l’entreprise. C’est ainsi qu’une mauvaise utilisation des ressources augmente les dépenses de l’entreprise. Le contrôle des ressources devient alors impensable pour les entreprises consommatrices du Cloud.

C'est dans ce cadre que l'équipe Tritux a eu l'idée de créer une application permettant de gérer leurs ressources Cloud par projet et budget grâce aux nouvelles technologies de l'information.

Ce projet s'est déroulé selon l'esprit de la méthode agile du Scrum ainsi la répartition du rapport va simuler les différents sprints réalisés. Dans cette optique, nous introduisons le plan de ce rapport :

Nous introduisons dans le premier chapitre le contexte, les notions théoriques liées au projet et ses objectifs à savoir les fonctionnalités envisagées, Nous présentons de même l’entreprise d'accueil, ses domaines et la solution actuelle.

Le deuxième chapitre sera dédié à l'analyse et les spécifications fonctionnelles et non fonctionnelles pour la réalisation de notre application.

Ensuite, nous entamerons la classification des chapitres selon la méthode Scrum c'est à dire en sprint et chaque sprint abordera une ou plusieurs fonctionnalités de notre application de l'analyse jusqu'à la réalisation en passant par la conception.

Enfin nous terminerons par une conclusion qui établit le bilan du travail effectué et ouvre des nouvelles perspectives pour améliorer l'application.

Chapitre 1

Cadre général du projet

1.1 Introduction

Dans ce premier chapitre, nous présenterons le cadre général du projet. D'abord, nous commencerons par la présentation de l'organisme d'accueil. Ensuite, nous discuterons la problématique. Nous développerons une étude de l'existant. Enfin, nous proposerons une solution à cette problématique en exposant la méthodologie de travail que nous avons suivi.

1.2 Présentation de l'organisme d'accueil

Dans la partie qui suit, nous allons présenter l'organisme d'accueil, ses domaines, ses compétences ainsi que ses certifications.

1.2.1 Présentation de l'organisme d'accueil

TriTUX est une société internationale spécialisée dans l'ingénierie logicielle, le développement des systèmes d'informations, le conseil en informatique et l'externalisation. Avec plus de 13 ans d'expérience, elle est l'un des principaux fournisseurs de solutions informatiques et de télécommunication. Grâce à une infrastructure solide et à des multiples certifications dans les nouvelles technologies, TriTUX fournit des services d'ingénierie et de conseil en innovation et en agilité commerciale. Elle perçoit l'environnement commercial et la rentabilité de ses clients en proposant des solutions adaptées et personnalisées avec un ensemble complet de services, de la conception à la mise en œuvre.

1.2.2 Solutions

TriTUX est un partenaire stratégique dans :

- **IoT** : Une large gamme de solutions IoT puissantes répondant aux besoins spécifiques des entreprises
- **VAS** : Une large gamme de solutions à valeur ajoutée allant au-delà des normes vocales, de messagerie et de télécopie.
- **Roaming** : Des outils multifonctions gérant les défis opérationnels et commerciaux.
- **Solutions informatiques** : Des solutions d'itinérance innovantes pour renforcer l'image de marque de ses clients et maximiser leurs revenus.

1.2.3 Domaines et compétences

Tritux propose un large éventail de services de conseil, d'ingénierie et d'analyse. Ces services s'adressent à la chaîne de valeur complète d'ingénierie couvrant divers secteurs :

- Consulting et administration Unix et Base des données.
- Datawarehousing et business intelligence.
- Messagerie multimédia SMS/MMS/FAX/WAP/WEB.
- Conception de solutions critiques et à fortes charges.
- Systèmes d'information et services spécialisés dans l'architecture SOA.

1.2.4 Collaborateurs

TriTUX a su se hisser dans la cour des grands en se positionnant sur le marché et en fidélisant de nombreux clients importants en Europe et en Afrique. Parmi celles-ci, figurent notamment :



FIGURE 1.1 – Les collaborateurs de TRITUX

1.2.5 Certifications

TRITUX a déployé une infrastructure solide et des certifications reconnues dans le domaine des nouvelles technologies répondant à toutes les exigences, en appliquant les meilleures pratiques des secteurs public et privé du monde entier et en exploitant la technologie innovante.

Parmi les certificats acquis par TriTUX on note :

- Certification en management de la qualité ISO 9001.
- International Software Testing Qualifications Board ISTQB.
- Oracle Java Certified Professional.
- Linux Red Hat certified engineer.

1.2.6 Organigramme

L'organisation hiérarchique de Tritux est présentée dans l'organigramme de la figure 1.2 suivante :

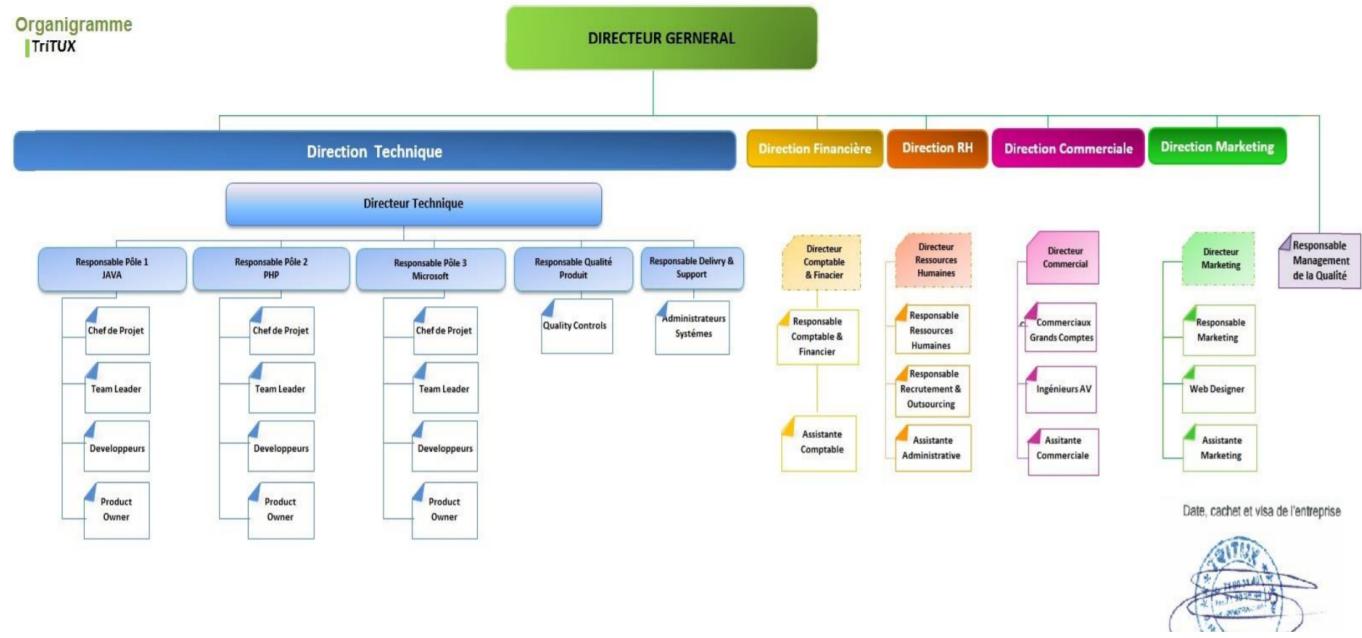


FIGURE 1.2 – Organigramme de TRITUX

1.3 Cadre du projet

1.3.1 Contexte du projet

Dans le cadre de l'optimisation de consommation de ses ressources et la diminution de ses dépenses, Tritux vise à mettre en place une plateforme de budgétisation des ressources Cloud. Pour ce fait, l'entreprise d'accueil nous a confié cette tâche de conception et développement d'une application de gestion des ressources Cloud par projet et budget.

1.3.2 Problématique

Avant l'utilisation du Cloud les entreprises ont beaucoup souffert de la complexité de la gestion de leurs systèmes informatiques. Quelle que soit la taille de l'entreprise, elle avait toujours besoin d'investir sur le matériel, sur son installation et sa maintenance afin d'offrir un produit de qualité et de s'imposer sur le marché. Ce qui traduit la migration immense des architectures traditionnelles vers le Cloud.

Bien que l'utilisation du Cloud renforce la scalabilité de l'entreprise à moindre coût et améliore sa rentabilité. La mauvaise utilisation des ressources provoque des coûts imprévisibles. Ceci s'explique par le fait que le Cloud suit un modèle de paiement à l'usage et que la quantité de ressources consommées est plus onéreuse.

À cet égard, Tritux est présenté comme étant l'une des entreprises qui souffrent du surcoût de consommation due à une utilisation incontrôlée des machines virtuelles(VM). Elle cherche donc à planifier et restreindre l'utilisation des instances de VMs en se basant sur le budget affecté à chaque projet ainsi que le rôle et les permissions de chaque employé membre du projet récemment énuméré.

1.3.3 Etude de l'existant

L'étude approfondie de la politique actuellement utilisée par l'entreprise d'accueil pour la gestion des ressources Cloud présente une phase indispensable pour la réussite de notre travail. Dans cette section, nous allons présenter la solution actuelle adoptée par TRITUX pour gérer ses ressources Cloud . Nous allons critiquer cette solution afin d'envisager les points faibles et développer une application qui répond aux besoins de l'entreprise d'accueil.

Solution actuelle

Pour orchestrer le fonctionnement des instances de VM et minimiser leurs consommations, TRITUX a utilisé la console de la plateforme de Google Cloud en y intégrant un libellé qui permet d'introduire la durée d'allocation de la machine virtuelle. Dès qu'active, la facturation est donc à l'usage, pendant la durée sollicitée.

Critiques de l'existant

Après avoir exposé la directive adopté par TRITUX pour la gestion de ces instances, nous avons énumérer quelques limites pour l'adoption de cette solution. Tout d'abord, l'utilisation directe de la console de la plateforme de google Cloud n'en- gendre pas vraiment l'économie de l'argent. En fait , oublier une VM active hors échéance épouse rapidement et sans intérêt la durée sollicité. L'entreprise se trouve obligé d'étaler cette durée pour assurer l'avancement du projet. Ceci provoque des frais supplémentaires . Ensuite, la plateforme de Google Cloud ne permet pas de restreindre la gestion des VMs par privilège. En d'autre terme, elle n'a aucune vision sur les projets ni sur les rôles des membres. Elle active et désactive les instances suite à des requêtes émises par l'entreprise. Ainsi cette dernière, n'offre pas les fonctionnalités suivantes :

- Authentification sécurisé et gestion d'accès (compte employé, oublie mot de passe,etc).
- Affectation des instance de VM à un projet bien spécifique. .
- Contrôle de budget du projet .
- Gestion des employés (profil, rôle,etc).
- Restriction d'accès aux projets et aux instances de VMs.

Objectifs du projet

Après avoir dégagé les problèmes du système existant , l'entreprise d'accueil nous a proposé dans le cadre de notre projet de fin d'étude de concevoir et d' implémenter une application de gestion des ressources Cloud par projet et budget. Pour ce faire, la manipulation et l'intégration des APIs du fournisseur Google Cloud platform constitue la base de notre projet. Ces API répondent aux besoins d'automatisation de fonctionnement des machines virtuelles basée sur des critères de décisions. Parmi ces critères nous développons un planificateur de temps et de budget afin d'empêcher l'utilisation anarchique des instances. Nous visons aussi à intégrer d'autres modules qui facilitent l'utilisation de notre application. Nous citons :

- Module d'authentification et de gestion de compte utilisateur :
- Module de gestion des utilisateurs.
- Module de gestion de projet.
- Module de création des machines virtuelles : permet de se débarrasser de l'obligation de la création des VMs à partir de l'interface du fournisseur

1.4 Etat de l'art

Dans cette partie, nous allons présenter et analyser en premier lieu les solutions existantes dans le marché. En second lieu, nous allons définir les notions de base indispensable à la compréhension de à notre projet notamment la virtualisation et le Cloud Computing. Nous allons présenter en dernier lieu le fournisseur de service Cloud avec lequel collabore l'entreprise d'accueil et les services qu'on va utiliser.

1.4.1 Analyse et critique des solutions existantes dans le marché

Dans cette section, nous présentons une analyse des solutions de surveillance et de gestion des instances de machine virtuelle (VM). Ces applications ont pour but de démarrer/arrêter les VM,s , de surveiller leurs états et leurs performances et de planifier leurs fonctionnements. Nous citons :

- CloudCheckr.
- Skeddly.
- ParkMyCloud.

Le tableau 1.2 ci-dessous souligne une analyse comparative des fonctionnalités de ces solutions.

TABLE 1.1 – Etude comparative des solutions existantes dans le marché

Applications	Points forts	Limites
ParkMyCloud	<ul style="list-style-type: none"> — Réduction du coût de la VM par le contrôle et la planification des ressources. — Contrôle d'accès et de permissions à travers les structures d'équipes et les rôles. 	<ul style="list-style-type: none"> — Manque de possibilité d'ajout de nouvelles instances Cloud directement à partir de la plateforme. — Gestion des instances guidée par le temps uniquement.
Skeddly	<ul style="list-style-type: none"> — Simplicité d'utilisation offre aux clients une gestion guidées des services Cloud. — Prix abordable pour les petites et moyennes entreprises. 	<ul style="list-style-type: none"> — Dépendance aux ressources Cloud d'Amazon uniquement. — Manque de l'analyse de capacité qui permet d'optimiser la performance et l'efficacité des ressources existantes.
CloudChekr	<ul style="list-style-type: none"> — Respect d'engagement de services par le "Service level agreement" (SLA). — Documentation automatisée des logs. — Identification proactive des risques pour gérer la sécurité du Cloud public. — Analyses de facturation détaillées avec des alertes. 	<ul style="list-style-type: none"> — Interface utilisateur complexe. — Quelques fonctionnalités ont tendance à se rompre sans avertissement. — Coût relativement cher.

Comme l'illustre le tableau 1.2, ces applications sont bien riches en fonctionnalités intéressantes. Cependant, Tritux n'a pas adopté l'une de ces plateformes vu que certaines fonctionnalités ne sont pas conformes aux besoins de l'entreprise. Parmi les frein de l'adoption nous citons :

- Allocations des instances de VMs non budgétisées.
- Solution pour Multi-cloud Cloud or Tritux coopère avec un unique fournisseur.
- Grande quantité de données qui rend l'utilisation de plateforme difficile et lourde.
- Inaptitude à créer des instances du fournisseur Cloud via la plateforme.
- Une inaptitude à créer des instances du fournisseur Cloud via la plateforme.
- Solutions payantes.
- Absence de gestion des projets.

1.4.2 Virtualisation

Toute entreprise cherche à améliorer l'efficacité et la disponibilité de ses ressources. Elle cherche à remplacer l'ancien modèle "un serveur dédié pour une application" par « un serveur dédié à plusieurs applications » grâce à la virtualisation. Du coup, la virtualisation a entamé le monde de l'informatique avec beaucoup plus d'avantages tels que :

- Utilisation optimale des ressources.
- Allocation dynamique de la puissance de calcul en fonction des besoins.
- Economie sur le matériel par mutualisation.

Selon Redhat « La virtualisation est une technologie qui permet de créer plusieurs environnements simulés ou ressources dédiées à partir d'un seul système physique. Son logiciel, appelé hyperviseur, est directement relié au matériel et permet de fragmenter ce système unique en plusieurs environnements sécurisés distincts. C'est ce que l'on appelle les machines virtuelles, ou VM. Ces dernières exploitent la capacité de l'hyperviseur à séparer les ressources du matériel et à les distribuer convenablement. »[redhat].

: Il existe plusieurs types de virtualisation parmi lesquels nous citons :

- **Virtualisation matérielle** : Le type de virtualisation le plus courant L'hyperviseur crée des versions virtuelles des ordinateurs et des systèmes d'exploitation et les consolide en un seul grand serveur physique.
- **Virtualisation du serveur** : Un regroupement des serveurs physiques sous-employés sur un seul hôte qui exécute des systèmes virtuels.
- **Virtualisation du stockage** : Une virtualisation économique permettant de regrouper un ensemble de périphériques de stockage afin qu'il rassemble à un seul périphérique.
- **Virtualisation du système d'exploitation** : Le noyau autorise l'existence de plusieurs instances d'espaces utilisateurs, cette virtualisation est utilisée principalement pour tester les applications sur différentes plateformes de système d'exploitation.

La virtualisation et le cloud computing sont deux notions complémentaires. Ceci s'explique par le fait que la délivrance des services Cloud se repose principalement sur la virtualisation. Le Cloud Computing n'est pas donc une invention mais une évolution des technologies. Nous allons exposer dans la section suivante ce nouveau paradigme.

1.4.3 Cloud Computing

De plus en plus utilisé, le cloud computing est jusqu'à présent considéré comme l'évolution majeure de l'informatique du 21ème siècle. Ce dernier est devenu la fourniture des services informatiques, permettant un accès omni présent ,à la demande, à travers un réseau à un pool de ressources informatiques partagé. Cet accès obéit à la règle n'importe où n'importe quand et n'importe comment. Ces ressources peuvent être des applications, puissances de calcul, espaces de stockage ou serveurs indépendamment de leurs localisations géographiques.

Pour réussir à présenter une innovation et un meilleur produit à leurs clients, Les grandes entreprises du secteur informatique sont massivement impliquées dans des activités liées au Cloud en faveur de profiter de ses services.

En se basant sur une couche d'abstraction proposée par la virtualisation. Cloud Computing offre trois types de services définies par la figure 1.2.

- **Software-as-a-service (SaaS)** : Des applications, aux abonnements, installées ou accessibles via un navigateur. En profitant de ce type de service les fournisseurs

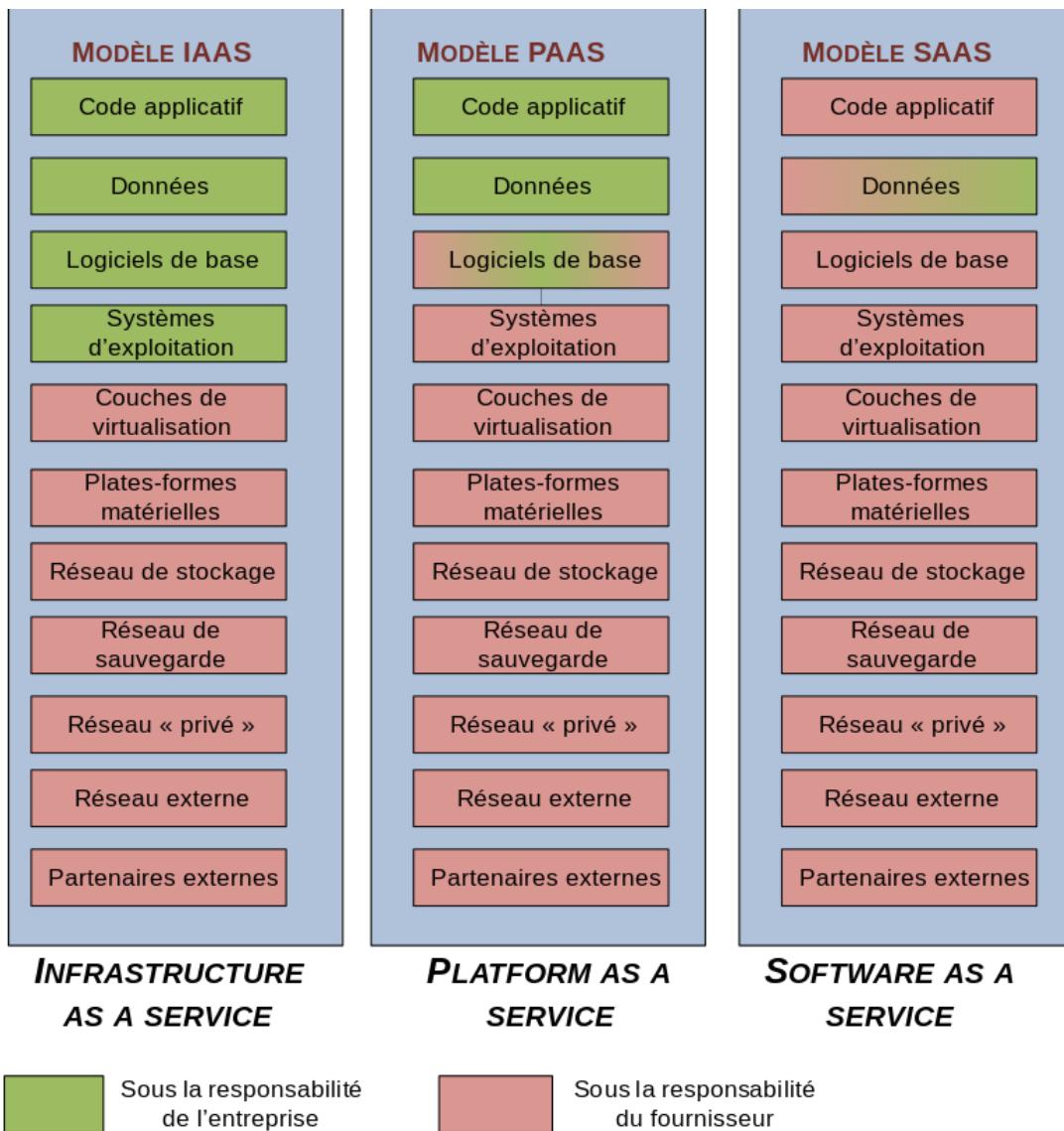


FIGURE 1.3 – Les services du Cloud Computing

gèrent de manière transparente l'ensemble des aspects techniques de l'application. Quant aux clients, ils se limitent à effectuer quelques paramétrages de l'application.

- **Plateform-as-a-service (PAAS)** : Des plateformes de développement ou de test dont le système d'exploitation et les outils sont gérés par les fournisseurs du Cloud. Or les clients gardent la main sur l'installation des applications et leurs paramétrages.
- **Infrastracture-as-a-service (IAAS)** : Les fournisseurs offrent des ressources informatiques virtualisées (stockage, réseaux, virtualisation, etc.) sur lesquelles le consommateur garde le contrôle sur le système d'exploitation, les applications déployées et certains composants de réseau. Les domaines d'exploitation des infrastructures sont divers, Ils peuvent être utilisées comme un réseau privé virtuel dans les entreprises ou pour héberger des sites sur des serveurs virtuels ou pour l'utilisation des data centers virtuels.

Ces services peuvent être déployer sur différents modes :

- **Cloud privé** : C'est un modèle très répandu dédié à une seule entreprise. Le déploiement de ce dernier peut se faire sous deux autres formats possibles :

- Cloud privé internalisé : Dans ce cas, le Cloud est interne à l'entreprise ou entièrement dédié à cette même entreprise. Il est accessible via des réseaux sécurisés et opérés par les équipes internes.
- Cloud privé externalisé : Dans ce cas, le Cloud offre des services similaires au Cloud privé internalisé. Ce dernier est entièrement dédié à l'entreprise, mais hébergé chez un tiers.
- **Cloud public** : Dans ce cas, le Cloud est externe à l'entreprise et géré par un opérateur externe propriétaire des infrastructures, avec des ressources totalement partagées entre tous ses clients.
- **Cloud hybride** : Dans ce cas, il s'agit de l'association de deux ou plusieurs modèles de cloud pour assurer une portabilité des applications et des données.

Bien que le marché du Cloud public est plein de fournisseurs, la concurrence est toujours concentrée entre Amazon Web Services, Microsoft Azure et Google Cloud Platform. Pour des raisons d'efficacité et de disponibilité, le choix de fournisseur pour Tritux a été fixé sur Google Cloud Platform.

1.4.4 Google Cloud plateform

La plateforme Cloud de Google (GCP) est un ensemble de ressources informatiques, mises à la disposition du grand public via des services modulaires basés sur le Cloud, sous la forme d'une offre de cloud public.

Celle-ci propose des services de calcul, stockage, apprentissage automatique et d'internet des objets, ainsi que des outils de gestion et de sécurité. Les principaux produits de Cloud computing de Google Cloud Platform sont répartis en trois grandes familles :

- **Google App Engine** : Un service Cloud de type PAAS permet aux développeurs d'accéder à l'hébergement évolutif de Google. Les développeurs peuvent également utiliser un kit de développement logiciel pour implémenter des produits logiciels s'exécutant sur App Engine.
- **Google Storage** : Une plateforme de stockage conçue pour stocker des Données de taille importante.
- **Google Compute Engine** : Une offre IaaS qui permet aux clients d'exécuter des charges de travail sur le matériel physique de Google. Il fournit un nombre évolutif de machines virtuelles (VM) pouvant servir de grands clusters de calcul à cette fin. Google Compute Engine peut être géré via une API REST, une interface de ligne de commande (CLI) ou une console Web. Compute Engine est un service à la carte avec un minimum de 10 minutes.

Ainsi ces ressources virtuelles devront être instanciées, allouées et orchestrées via des plateformes d'administration et de gestion des ressources Cloud.

1.5 Méthodologie de gestion de travail

Dans le processus de réalisation d'un projet complexe, l'utilisation d'une méthodologie de management de projet est indispensable afin d'assurer la réussite de ce dernier. En effet, une bonne méthodologie de projet fournit le cadre, les procédés, les lignes directrices et les techniques pour gérer à la fois le personnel et le travail.

Il existe de nombreuses méthodologies de gestion de projet, assez différentes les unes des autres de par leurs fonctionnements, mais ils se rejoignent tous afin de garantir le succès de la conduite du projet.

Pour notre projet nous avons opté pour une méthode agile qui permet de se focaliser sur la valeur métier du livrable, d'améliorer constamment la qualité de ce que nous réalisons et de réagir efficacement aux changements. De ce fait, nous allons donner tout d'abord un aperçu sur les méthodologies agiles. Puis, nous focalisons sur la méthodologie Scrum, qui a été utilisée dans le présent projet.

1.5.1 Méthodes Agiles

Les besoins, dans une équipe de développement logiciel qui travaille sur un projet interne sont parfois susceptibles d'être changés ou modifiés. De ce fait, l'équipe se trouve en train de développer une application avec des spécifications parfois non précises, ce qui peut entraîner des retards dans le déploiement du projet. Ces problématiques ont poussé les ingénieurs à réinventer les méthodes de gestion de projet et de conception en introduisant ce que nous appelons la méthode Agile.

1.5.2 Presentation de la méthodologie Scrum

Scrum est la méthodologie suivie par la société TRITUX pour la gestion de ses projets. C'est une méthodologie agile itérative basée sur des itérations d'une durée de 2 à 4 semaines appelées Sprints. Durant chacune de ces itérations, une partie du produit nommée « Incrément » est réalisée en se basant sur les parties créées lors des itérations précédentes et livrées à la fin du Sprint.

Elle offre les avantages suivants :

- Flexibilité.
- Pas de distinction des rôles au sein d'une équipe « Scrum ».
- Auto confiance au sein de l'équipe en l'isolant durant le Sprint et en la rendant autogérée.
- Équipes soudées.

Il existe différents rôles participants dans la mise en œuvre

du projet selon la logique de Scrum :

- **Product Owner** : C'est un membre à part entière de l'équipe Scrum dont la responsabilité principale est de définir un produit qui apportera le maximum de valeur métier aux utilisateurs dans le temps et le budget. Il présente les besoins du client et propose de nouveaux objectifs. Il contrôle aussi la qualité et la date de délivrance de chaque sprint
- **Scrum Master** : Il aide l'équipe à travailler de façon autonome et à s'améliorer constamment. Il est le garant de l'application du processus, Scrum en l'occurrence.
- **Scrum Team** : L'équipe de développement

Les artefacts de cette méthodologie sont de deux types : [Dup09]

1. **Backlog produit** : liste ordonnée de tout ce qui pourrait être requis dans le produit et elle est l'unique source des besoins pour tous les changements à effectuer sur le produit sous la responsabilité du Product Owner.
2. **Backlog sprint** : est constitué à partir des thèmes du Backlog produit. Lors de la réunion de planification de sprint, l'équipe de développement choisit les éléments du Backlog produit qui seront réalisés. Il est sous la responsabilité de l'équipe et elle est seule à pouvoir le modifier en cours d'itération.

1.6 Conclusion

Ce chapitre nous a permis d'introduire notre projet, de présenter l'organisme d'accueil et la problématique qui nous suscite à la réalisation de ce projet, ainsi que la solution proposée. Notre rapport sera organisé en respectant les normes de la méthode Agile SCRUM. En vue de mieux comprendre le métier, une description de l'environnement fera l'objet du prochain chapitre.

Chapitre 2

Sprint0 : Analyse et conception globales

2.1 Introduction

Ce chapitre sera dédié à l'analyse et à la spécification des besoins. En premier lieu, nous identifierons les acteurs principaux du projet. En second lieu, nous définirons les différents besoins fonctionnels et non fonctionnels de notre application. Par la suite, nous présenterons une étude conceptuelle globale. Enfin, nous clôturons par définir l'architecture utilisée.

2.2 Identification des acteurs

Un acteur présente une personne, une entité ou un système agissant d'une façon directe sur l'application.

Les différents acteurs modélisant notre système sont :

- *Utilisateur* : un employé de l'entreprise qui peut gérer son profil, consulter ses instances, gérer les planifications, etc.
- *Chef de projet* est un membre de la société, qui peut en plus des fonctionnalités récemment citées créer de nouvelle instance cloud, gérer des membres de son projet, etc.
- *Administrateur* directeur technique de l'entreprise et la personne la plus privilégiée, il possède le droit de manipuler toutes les fonctionnalités offertes par notre application.

L'ensemble de ces acteurs sont liés par une relation d'héritage.

2.3 Identification des besoins

Dans cette section, nous allons développer les besoins fonctionnels et non fonctionnels de l'entreprise d'accueil.

2.3.1 Besoins fonctionnels

Les besoins fonctionnels expriment les principales fonctionnalités de l'application sans se préoccuper de la façon de l'implémentation.

Une étude détaillée de système nous a permis de dégager les principaux exigences fonctionnelles des différents acteurs de notre application.

- Authentification et gestion de mot de passe.
- Consultation et modification du profil.
- Gestion des projets notamment création, suppression et modification.
- Gestion des utilisateurs qui contient le volet de consultation et de suppression.
- Gestion des planifications notamment création et suppression.
- Affectation/ retrait d'un projet à un utilisateur.
- Affectation/retrait d'un planning des instances de machines virtuelles.
- Manipulation des instances de machine virtuelle. Un utilisateur peut consulter, activer, arrêter une instance.
- Attribution des droits d'accès différents selon l'utilisateur.
- Gestion des machines virtuelles.
- Gestion des instances de machine virtuelle qui contient le volet de création et de suppression réalisée que par le chef de projet et l'administrateur.

2.3.2 Besoins non fonctionnels

Pour garantir un bon fonctionnement de l'application, cette dernière doit assurer les besoins non fonctionnels présentés ci-dessous.

- Ergonomie : Les interfaces doivent être conviviales, ergonomiques et facile à exploiter par l'utilisateur.
- Fiabilité : L'application doit fournir des résultats correctes.
- Sécurité : L'accès à l'application ainsi qu'aux données doit être sécurisé. D'une part par la gestion des autorisations aux différents modules de l'application grâce aux priviléges gérés par le module d'administration et d'une autre part par l'intégration d'un module d'authentification basée sur l'échange de jetons entre le serveur et le client.
- Extensibilité : Le système doit être extensible et permet de s'intégrer facilement sur le réseau existant de l'entreprise. Il doit aussi supporter l'intégration d'autres fonctionnalités
- Maintenabilité : La maintenabilité et l'évolutivité sont des priorités. Le code sera lisible, commenté, divisé en fonction des pages (des interfaces) et en fonction des tâches abordées.

2.4 Étude conceptuelle

Nous allons exposer dans cette section le diagramme de cas d'utilisation et de classe globales modélisant notre projet.

2.4.1 Diagramme de cas d'utilisation global

Dans cette partie, nous présentons le diagramme de cas d'utilisation global qui modélise l'interaction entre le système informatique à développer et les acteurs interagissant avec le système. Également, il permet de recenser les besoins des utilisateurs et les fonctionnalités du système.

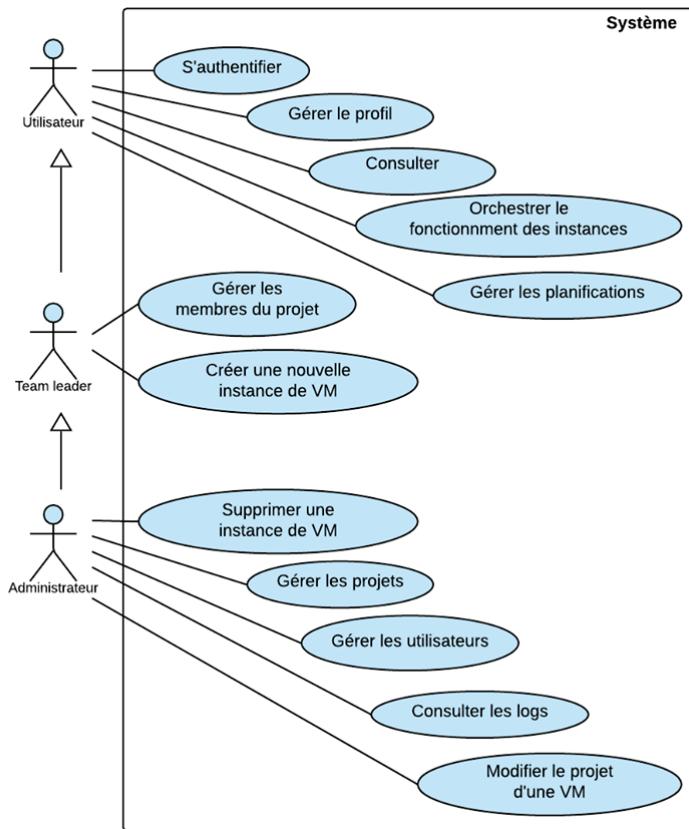


FIGURE 2.1 – Diagramme de cas d'utilisation global

La figure 2.1 ci-dessus représente le diagramme de cas d'utilisation global de notre projet où on y trouve les acteurs principaux et leurs rôles. Ce diagramme décrit de manière globale les différentes fonctionnalités de chaque acteur. En effet, tout utilisateur est apte à s'authentifier, gérer son profil, consulter les projets, gérer les planifications de fonctionnement des machines virtuelles et orchestrer leurs fonctionnements. Notre deuxième acteur est un Team leader qui est utilisateur privilégié par la gestion des membres du projet et la création des machines virtuelles. De plus que les fonctionnalités de Team leader et d'utilisateur, l'administrateur assure la gestion des utilisateurs et la consultation des logs.

2.5 Étude conceptuelle globale

2.5.1 Diagramme de classe global

Le diagramme de classe est considéré parmi les diagrammes les plus importants dans la modélisation de l'UML(Unified Modeling Language), comme présenté dans la figure 2.2, il permet de décrire la structure statique d'un système en présentant ses différents classes, attributs, méthodes et relations entre ses objets.

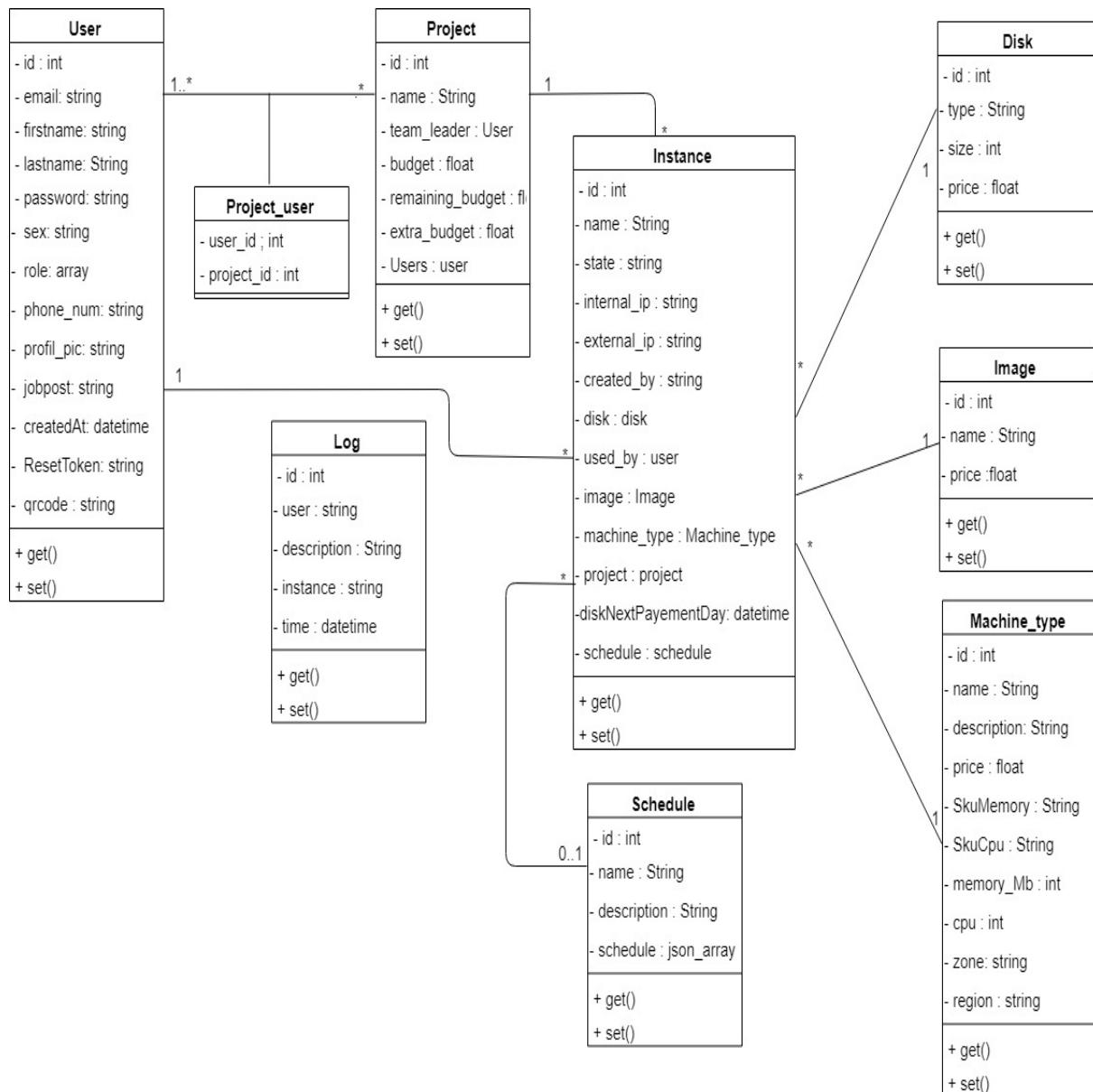


FIGURE 2.2 – Diagramme de classe global

2.6 Spécificités techniques

2.6.1 Backlog produit

Le Backlog est un artefact très important dans Scrum. C'est l'ensemble des caractéristiques fonctionnelles ou techniques qui constituent le produit souhaité. Le tableau 2.1 résume le backlog produit de notre application où chaque user story est caractérisé par un identifiant, un nom et une estimation.

ID	Tâches	User Story	Esti.
1	Gestion des comptes	1.1 En tant que utilisateur, je souhaite m'inscrire à l'application.	3
		1.2 En tant que utilisateur, je souhaite m'authentifier à l'application.	5
		1.3 En tant que utilisateur je souhaite recevoir un mail de réinitialisation du mot de passe en cas d'oubli.	5
		1.4 En tant que utilisateur, je souhaite consulter mon profil.	3
		1.5 En tant que utilisateur, je souhaite mettre à jour mes données personnelles.	3
2	Gestion des utilisateurs	2.1 En tant que administrateur, je souhaite consulter la liste des utilisateurs de l'application	2
		2.2 En tant que administrateur, je souhaite supprimer un utilisateur.	2
		2.3 En tant que administrateur, je souhaite affecter un projet à un utilisateur.	3
		2.4 En tant que administrateur, je souhaite retirer un projet d'un utilisateur.	3
3	Management des projets	3.1 En tant que Team leader, je souhaite consulter mes projets.	2
		3.2 En tant que administrateur, je souhaite supprimer un projet.	2
		3.3 En tant que administrateur, je souhaite créer un projet.	2
		3.4 En tant que administrateur, je souhaite modifier un projet.	2
		3.5 En tant que Team leader, je souhaite ajouter un membre à mon projet.	3
		3.6 En tant que Team leader, je souhaite retirer un membre de mon projet.	3
4	Management des VMs	4.1 En tant que utilisateur je souhaite consulter la liste des VMs.	5
		4.2 En tant que utilisateur je souhaite orchestrer le fonctionnement des VMs.	5
		4.3 En tant que Team leader je souhaite créer de nouvelles instances de machine virtuelle.	7
		4.4 En tant que administrateur, je souhaite supprimer une machine virtuelle.	7
		4.5 En tant que administrateur, je souhaite modifier le projet d'une machine virtuelle.	3

ID	Tâches	User Story	Esti.
5	Gestion des planifications	5.1 En tant que utilisateur, je souhaite consulter la liste des plannings.	2
		5.2 En tant que utilisateur, je souhaite créer un planning.	4
		5.3 En tant que utilisateur, je souhaite supprimer un planning.	3
		5.4 En tant que utilisateur, je souhaite affecter un planning à une VM.	5
		5.5 En tant que utilisateur, je souhaite retirer un planning d'une VM.	3
6	Gestion des logs	6.1 En tant que utilisateur, je souhaite consulter les activités des utilisateurs.	3

TABLE 2.1 – Backlog produit

2.6.2 Planification du projet

La planification de notre projet est résumée par la figure 2.3

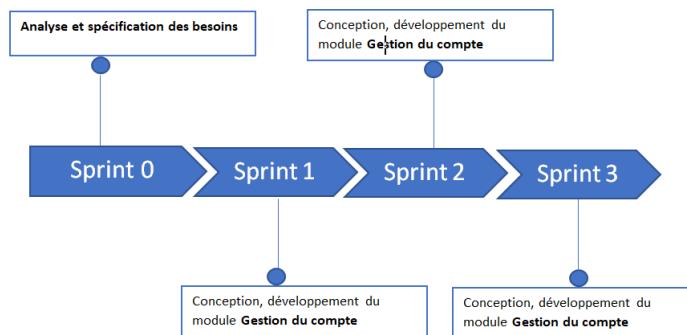


FIGURE 2.3 – Découpage du projet

- **Sprint 0 :** contient une étude préalable du projet.
- **Sprint 1 :**
 - Gestion des comptes : contient toutes les fonctionnalités liées à l'accès à l'application et le contrôle des données personnelles.
 - Gestion des utilisateurs : contient les fonctionnalités liées au contrôle des utilisateurs.
- **Sprint 2 :**
 - Management des projets : contient les fonctionnalités liées à la gestion des projets de Tritux.
 - Management des ressources Cloud : contient les fonctionnalités liées aux contrôle et gestion des ressources Cloud.
- **Sprint 3 :**
 - Gestion des Planifications.
 - Gestion des logs

2.6.3 Environnement de travail

La réussite de notre travail dépend énormément du choix des technologies et outils utilisés tout en tenant compte du besoin de l'entreprise.

Choix techniques pour le Front-end

Angular 7

Angular 7 est la version la plus récente du framework open source Angular, lancé le 18 octobre 2018 par Google. Ce dernier est un framework modulaire à base de components et modules. Il assure la création des applications mono-pages (SPA : Signle Page Application), web et mobiles.

Angular 7 propose plusieurs fonctionnalités parmi eux :

- Pipe : Composant permettant la transformation et la personnalisation de l'affichage des données dans les balises HTML.
- Vue : Des fichiers de HTML5 et CSS3.
- HTML5 : Le standard HTML est l'acronyme de « HyperText Mark-Up Language », c'est un langage de balisage développé pour la formalisation de l'écriture d'un document.
- CSS3 : Les feuilles de styles sont un langage qui permet de gérer la présentation et la mise en forme d'une page web.
- Contrôleur : Classe TypeScript représente la logique de la vue. Elle traite les différentes tâches à réaliser. Ainsi elle fait le lien entre la vue et les services en invoquant ses méthodes.
- TypeScript : Langage de programmation open-source développé par Microsoft. Il s'agit d'un sur-ensemble syntaxique strict de JavaScript, qui ajoute un typage statique optionnel au langage. TypeScript tend à améliorer et à sécuriser la production du code JavaScript.
- Services : Composant assurant l'émission, le traitement et la réception des requêtes http, en vue d'assurer l'échange des données JSON entre les services Web et le contrôleur.
- JSON : JavaScript Object Notation est un format d'échange de données léger. Ainsi, Il est facile pour la lecture et écriture des données. Basé sur le Javascript, le JSON est un format de texte totalement indépendant du langage, c'est-à-dire, il permet de faire communiquer deux langages de programmation différents.

Framework	ReactJS	Angular	Vue
Courbe d'apprentissage	Difficile	Moyen	Facile
Scalabilité	Haute	Haute	Faible
Communauté et popularité	Le plus populaire, entraîne de devenir un bon choix pour les applications mobiles natives	Il se développe tellement vite	Une communauté très active et entraîne de se développer
Emplois	Fortement demandé	Fortement demandé	Moins populaire et n'est pas supporté par une grande entreprise comme Facebook ou Google.
Performance	Haute	moyenne	Haute

TABLE 2.2 – Comparaison des Frameworks les plus populaires du Front-end

En se basant sur le tableau comparatif 2.2, le choix du Framework à utiliser a été fixé sur Angular 7 puisqu'il répond parfaitement à nos besoins et aux exigences techniques du développement de notre application.

Choix techniques pour le Back-end

Symfony 4

Symfony est un puissant Framework de PHP qui permet de réaliser des sites complexes rapidement, mais de façon structurée et avec un code clair, maintenable et sécurisé en respectant les normes de programmation. Ainsi Symfony bénéficie d'une richesse des bundles (plugins) développés et permettent de séparer la partie métier de la partie données. Ce dernier est utilisé par Tritux pour la plupart des projets web.

Outils

— Cron job

Cron est un utilitaire qui planifie l'exécution automatique d'une commande ou d'un script sur le serveur à des temps prédéfinis ou après certains intervalles prédéfinis. Un travail cron est la tâche planifiée elle-même. Les tâches Cron peuvent être très utiles pour automatiser des tâches répétitives tels que le contrôle du budget restant du projet et l'automatisation du fonctionnement des VMs.

— Node.js

Node est une plateforme de développement open source permettant d'exécuter du code JavaScript côté serveur. Node est utile pour développer des applications nécessitant une connexion persistante du navigateur au serveur et exécutant sur un serveur HTTP dédié.



FIGURE 2.4 – Logo Nodejs

— **Git**

Git est un système de contrôle de version distribué gratuit et à source ouverte, conçu pour gérer tout projet, du plus petit au plus grand, avec rapidité et efficacité.



FIGURE 2.5 – Logo Git

— **Postman**

Postman est un puissant client HTTP pour tester les services Web, il facilite le test, le développement et la documentation des API en permettant aux utilisateurs de créer rapidement des requêtes HTTP simples et complexes.



FIGURE 2.6 – Logo Postman

2.7 Architecture de l'application

2.7.1 Architecture physique de l'application

Notre application respecte la logique de l'architecture 3-tiers : nous avons un serveur pour la Base de données, un serveur Web pour recevoir et émettre les requêtes du client et un serveur client.

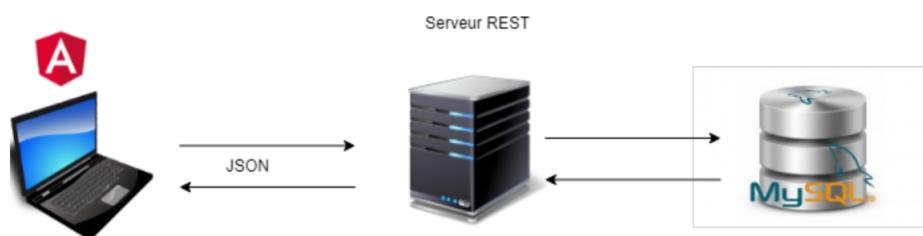


FIGURE 2.7 – Architecture physique de l'application

- Couche de présentation : Elle correspond à la partie visible et interactive de l'application pour les utilisateurs.
- Couche de logique métier : Elle permet d'appliquer les règles du métier gérées par l'application. Elle agit sur les données capturées à partir de la couche d'accès des données.
- Couche d'accès aux données : Elle correspond aux données qui sont destinées à être conservées.

La communication entre le client et le serveur REST est assurée par le biais d'échange des APIs REST.

API REST

REST est une API qui utilise les méthodes HTTP pour accéder à des ressources distantes sur le web. Chaque ressource est représentée par une URI 6 qui permet d'avoir un système universel d'identification des éléments de l'application. Elle supporte plusieurs formats de données comme JSON, XML, YML.

2.7.2 Architecture logique de l'application

Architecture Front-end

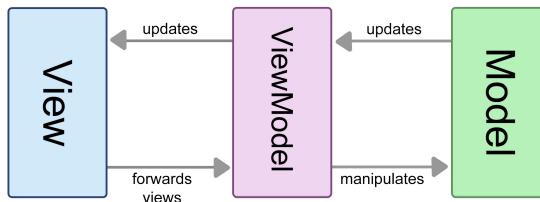


FIGURE 2.8 – Architecture logique Front-end de l'application

MVVM pour model view view-Model est un design pattern qui facilite la séparation du développement de l'interface utilisateur, c'est-à-dire de la couche de présentation. Comme le montre la figure 2.12 ViewModel (VM) est chargé d'exposer les objets de données du modèle de manière à ce que les objets soient plus facilement gérés et présentés. Pour son architecture, Angular adopte le modèle MVVM, cette implémentation est présentée comme suit :

- *Model* est implémenté en tant que service angulaire.
- *View* est implémentée à l'aide d'un modèle angulaire.
- *ViewModel* est implémenté en tant que contrôleur angulaire. La directive 'ngController' permet de spécifier un contrôleur pour une vue.

Architecture Back-end

Symfony est basée sur l'architecture MVC (modèle, vue et contrôleur) qui est un modèle architectural très puissant, il tire sa puissance de son concept de base qui est la séparation des données (modèle), de l'affichage (vue) et des actions (contrôleur).

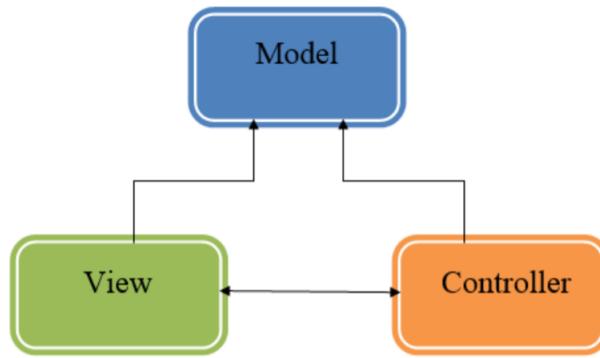


FIGURE 2.9 – Architecture logique Back-end de l’application

2.8 Conclusion

Ce chapitre nous a été utile pour montrer notre objectif, nos besoins et éclaircir notre démarche à travers l’identification des acteurs et des besoins. Il nous a offert une vision plus détaillée en réalisant une étude conceptuelle globale et en présentant les spécificités techniques de l’application. Le chapitre suivant sera consacré à réaliser le premier incrément de notre projet.

Chapitre 3

Sprint 1 : Gestion des comptes et des utilisateurs

3.1 Introduction

L'objectif de ce chapitre est de présenter la première itération du cycle de vie de notre projet. Nous entamons par identifier les tâches à réaliser dans le Backlog du Sprint pour passer par la suite aux phases d'analyse et de conception et nous finirons par exposer la phase de réalisation de ce module.

Nous avons découpés ce sprint en deux parties, une pour la gestion des comptes et une autre pour la gestion des utilisateurs.

3.2 Étude fonctionnelle

3.2.1 Backlog du sprint

Le Backlog du sprint présentés par le tableau 3.1 contient une liste des tâches de chaque user story identifiées par l'équipe Scrum et qui devront être réalisées pendant ce sprint.

ID	User Story	Description	Esti.
1.1	En tant que utilisateur, je souhaite m'inscrire à l'application.	Créer la vue d'inscription sécurisée par reCAPTCHA.	3
		Générer l'API de création d'un compte	3
		Consommer l'API	3

ID	User Story	Description	Esti.
1.2	En tant que utilisateur, je souhaite m'authentifier à l'application.	Créer la vue d'authentification	5
		Créer l'action d'interception de toutes les requêtes du front-end et de les affecter un Bearer Token correspondant.	4
		Implémenter l'API d'authentification en émettant le JWT (Json Web Token) vers le front-end	4
1.3	En tant que utilisateur je souhaite recevoir un mail de réinitialisation du mot de passe en cas d'oubli.	Consommer l'API et sauvegarder le token.	3
		Créer les vues de saisie du mail et de création d'un nouveau mot de passe.	3
		Implémenter l'API de réinitialisation du mot de passe.	5
1.4	En tant que utilisateur, je souhaite consulter mon profil.	Consommer l'API.	3
		Créer la vue de consultation du profil.	3
		Générer l'API permettant de retourner des données d'un utilisateur.	3
1.5	En tant que utilisateur, je souhaite mettre à jour mes données personnelles.	Consommer l'API	3
		Créer la vue de modification du profil, avec modal de confirmation par mot de passe	3
		Créer l'API de modification du profil.	3
		Créer l'API de confirmation du modification.	3
2.1	En tant que administrateur, je souhaite consulter la liste des utilisateurs de l'application	Consommer les deux APIs.	3
		Créer la "datatable" de sélection de tous les utilisateurs.	2
		Implémenter l'API de sélection des utilisateurs.	3
		Consommer l'API.	3

ID	User Story	Description	Esti.
2.2	En tant que administrateur, je souhaite supprimer un utilisateur.	Implémenter l'API de suppression d'un utilisateur.	2
		Consommer l'API.	3
2.3	En tant que administrateur, je souhaite affecter un projet à un utilisateur.	Créer la modale de consultation, affectation et de détachement des projets d'un utilisateur	3
		Implémenter l'API d'affectation d'un projet à un utilisateur.	3
		Consommer l'API.	3
2.4	En tant que administrateur, je souhaite retirer un projet d'un utilisateur.	Implémenter l'API de retrait d'un projet de l'utilisateur.	3
		Consommer l'API	3

TABLE 3.1 – Backlog du Sprint 1

3.2.2 Diagramme de cas d'utilisation du sprint 1

Le diagramme de cas d'utilisation du premier sprint présenté par la figure 3.1 a pour objectif de déterminer les besoins, les résultats attendus et les objectifs les plus prioritaires de la première valeur métier.

La détermination des besoins est basée sur la représentation de l'interaction fonctionnelle entre l'acteur et le système.

Tous les cas d'utilisation de ce sprint sont précédés par une opération d'authentification.

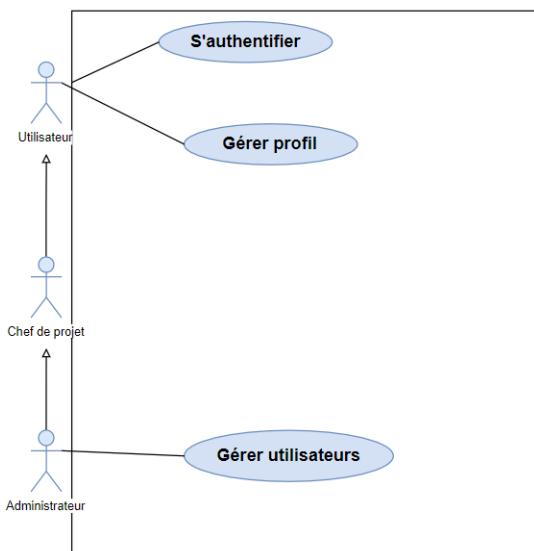


FIGURE 3.1 – Diagramme de cas d'utilisation du sprint 1

Comme le montre la figure 3.1, ce sprint permet aux utilisateurs d'accéder facilement à l'application et de gérer leurs données personnelles sur leurs profils. Ainsi, pendant

cette itération, l'administrateur devient capable de gérer tous les utilisateurs inscrits dans l'application.

3.3 Gestion des comptes

Cette partie est consacrée à la gestion des comptes personnels des utilisateurs. Ce dernier est composé par les cas d'utilisation "s'authentifier" et "gérer profil".

3.3.1 Analyse

Afin de mieux détailler le fonctionnement et assimiler les cas d'utilisation de la première partie constituant ce sprint, nous allons établir, dans cette section, leurs raffinements en livrant une description sur les différents scénarios possibles.

— Raffinement du cas d'utilisation "S'authentifier"

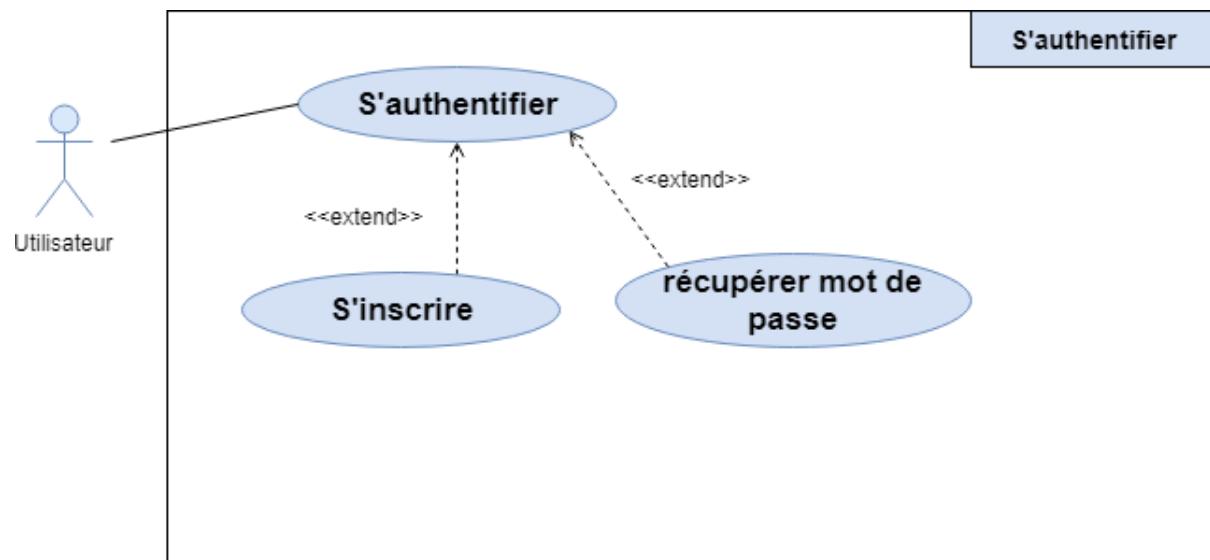


FIGURE 3.2 – Raffinement du cas d'utilisation : S'authentifier

Comme le montre la figure 3.2 un utilisateur peut soit s'authentifier ou bien créer un compte s'il s'agit d'un nouveau utilisateur. L'application offre la possibilité de récupérer son mot de passe en cas d'oubli en envoyant un nouveau mot de passe par email. Si l'utilisateur est authentifié avec succès, une page d'accueil sera ouverte, sinon, il sera redirigé vers la page d'authentification.

Description textuelle du cas d'utilisation "Créer un compte"

Acteur	Utilisateur
Description	Ajouter un utilisateur au système.
Préconditions	Disponibilité d'accès au serveur.
Post-conditions	Utilisateur inscrit.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur remplit le formulaire, valide reCAPTCHA puis confirme. 2. Le système vérifie l'unicité de l'utilisateur. 3. Le système génère un mot de passe aléatoire et l'envoie à l'utilisateur par mail. 4. Le système ajoute l'utilisateur .
Scénario alternatif	<ol style="list-style-type: none"> 2.a Le système retourne un message d'erreur indiquant l'existence d'un utilisateur avec des coordonnées pareil.

TABLE 3.2 – Description textuelle du « Crée un compte »

— Raffinement du cas d'utilisation "Gérer le profil"

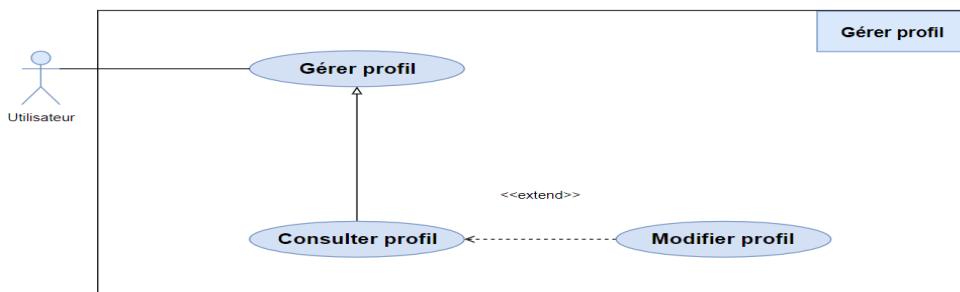


FIGURE 3.3 – Raffinement du cas d'utilisation : Gérer le profil

Le diagramme de cas d'utilisation de la figure 3.3 est une illustration des détails de la gestion du profil. En outre, ce diagramme montre la relation d'inclusion entre la consultation du profil et sa mise à jour.

Description textuelle du sous cas d'utilisation : "Mettre à jour le profil"

Acteur	Utilisateur.
Description	Mettre à jour profil.
Préconditions	Utilisateur authentifié.
Post-conditions	Profil mis à jour.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur clique sur l'icône de mise à jour du profil. 2. Le système afficher un formulaire. 3. L'utilisateur remplit le formulaire. 4. Le système met à jour le profil. 5. Le système affiche le profil modifié.
Scénario alternatif	<ol style="list-style-type: none"> 3.a L'utilisateur laisse un champ vide, ou saisit de données invalides. 3.b Le système répond avec des messages d'erreurs.

TABLE 3.3 – Description textuelle du « Mettre à jour le profil »

3.3.2 Conception

Les diagrammes de séquence objet permettent de représenter les vues dynamiques du système en montrant les collaborations entre les objets selon un point de vue temporel. Dans ce qui suit, nous allons représenter les diagrammes de séquences les plus importants de la première partie de cette itération.

Diagramme de séquences du cas d'utilisation "S'authentifier"

La figure 3.4 présentée ci dessous décrit la procédure d'authentification à l'application. Tout d'abord, l'utilisateur saisit son email et mot de passe. Une validation des champs se produit automatiquement, s'ils sont invalides, un message d'erreur s'affiche, sinon l'utilisateur continue sa procédure en validant recaptcha. Une fois validé et le bouton de connexion est appuyé, les données s'envoient au contrôleur pour vérifier l'existence et la correspondance des coordonnées. Si c'est le cas, le contrôleur génère un token et l'émet pour qu'il soit décodé et utilisé dans l'envoie des prochaines requêtes, sinon il retourne une requête http portant le statut 401 indiquant la non autorisation d'accès.

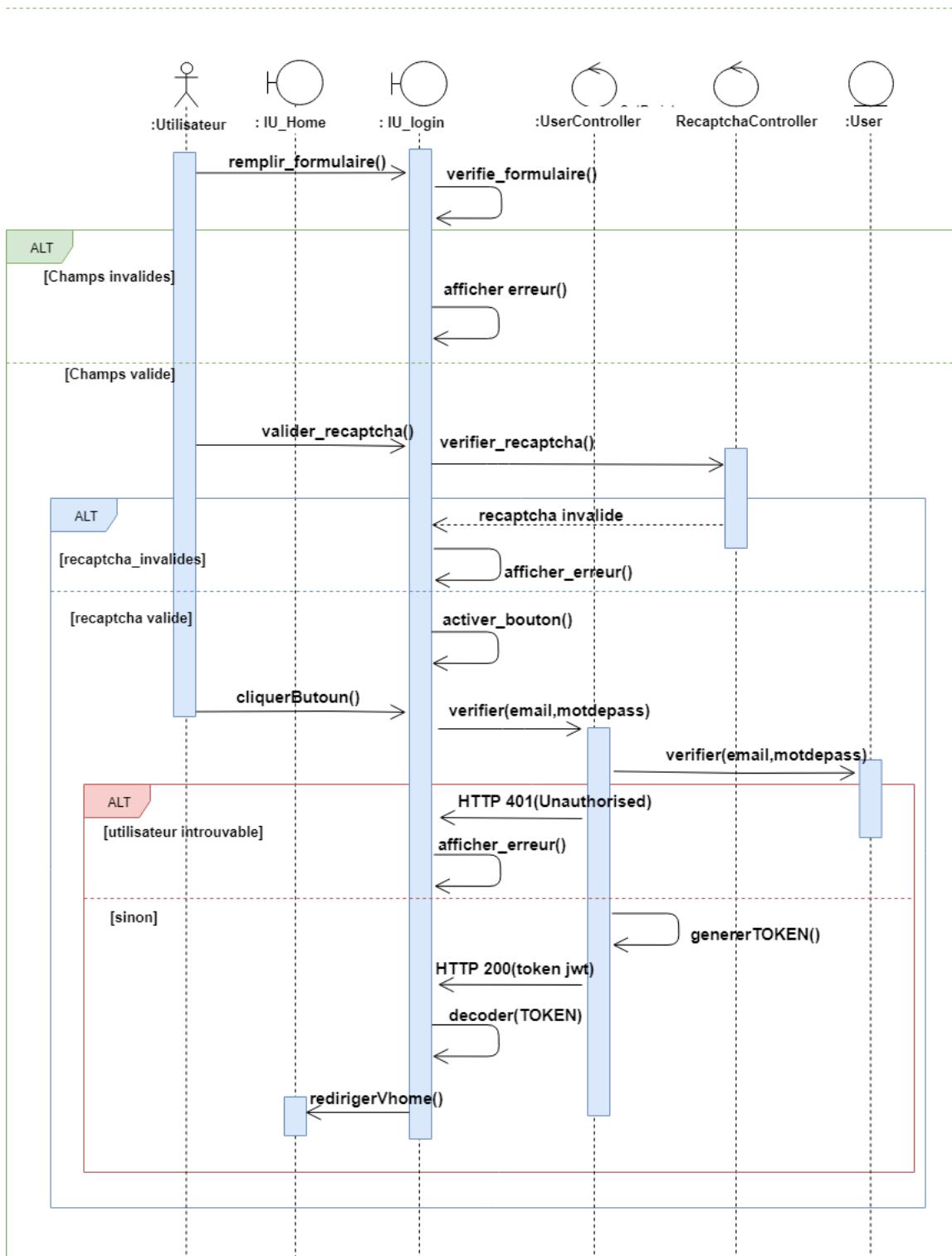


FIGURE 3.4 – Diagramme de séquence : S'authentifier

Diagramme de séquence du sous cas d'utilisation "Réinitialiser mot de passe"

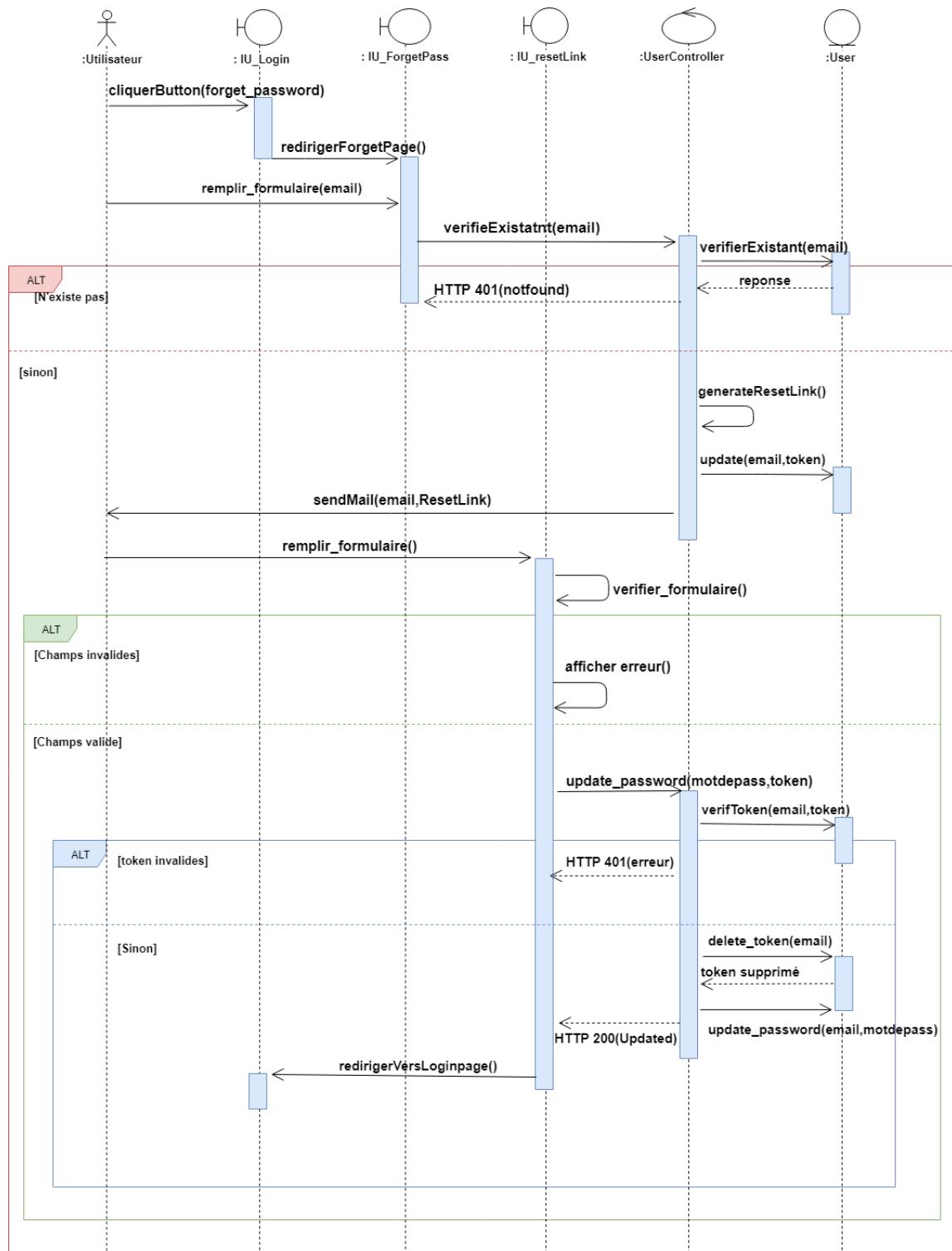


FIGURE 3.5 – Diagramme de séquence : Réinitialiser mot de passe

Le diagramme de la figure 3.5 ci dessus, met en évidence la procédure de réinitialisation du mot de passe. Tout d'abord, l'utilisateur demande un nouveau mot de passe en appuyant sur le bouton correspondant. Ensuite, il sera redirigé vers l'interface de réinitialisation du mot de passe, là où il devra saisir son email. Une fois son existence est validée, le contrôleur régénère un token et envoie un email contenant un lien de régénération du mot de passe. L'utilisateur choisit alors un nouveau mot de passe et l'émet au contrôleur

pour mettre à jour l'ancien token et l'ancien mot de passe. Le contrôleur répond avec une requête HTTP de statut 200 si tout est bien.

Diagramme de séquence du cas d'utilisation : "Mettre à jour son profil"

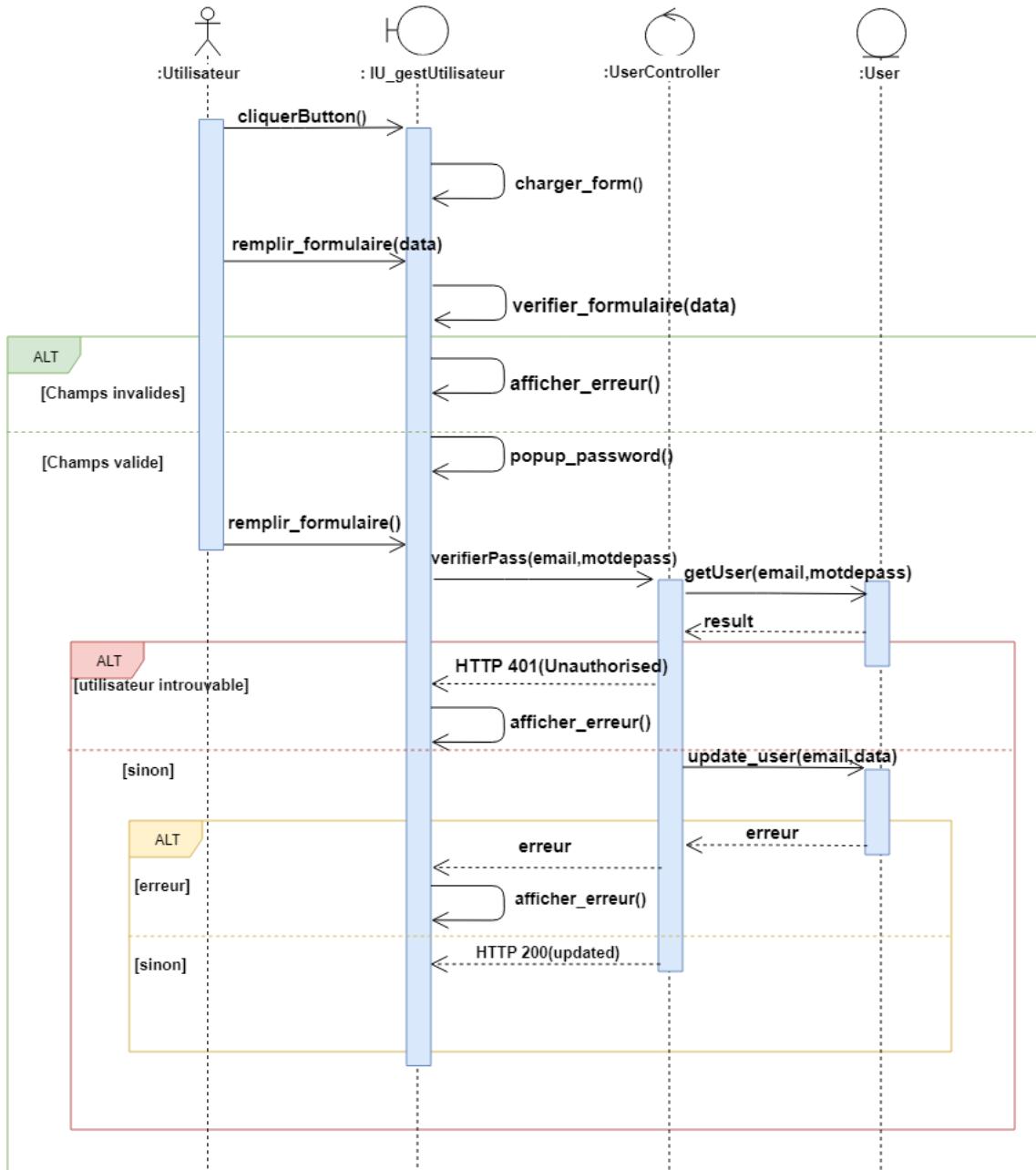


FIGURE 3.6 – Diagramme de séquence : Mettre à jour profil

Le diagramme de la figure 3.6 décrit la procédure de modification des données personnelles. En effet, l'utilisateur accède à son profil, là où il clique sur l'icône d'activation de mode édition. Il saisit les données à modifier et confirme sa modification en cliquant sur le bouton confirme. Une fois cliqué, une modale s'ouvre demandant de confirmer encore par le saisie du mot de passe. L'utilisateur saisit alors son mot de passe. Ce dernier est envoyé au contrôleur pour valider son existence. S'il est valide les changements seront pris

en compte et le contrôleur retourne une requête http de statut 200 sinon il retourne une requête de statut 401.

3.4 Gestion des utilisateurs

Cette partie est consacrée à la gestion des utilisateurs. Elle est composée par le cas d'utilisation "gérer les utilisateurs".

3.4.1 Analyse

Dans ce qui suit, nous allons offrir une vue d'ensemble des différentes fonctionnalités relatives à la deuxième partie de ce sprint. Ensuite, nous allons raffiner et décrire les cas d'utilisation les plus prioritaires.

Raffinement du cas d'utilisation : "Gérer utilisateurs"

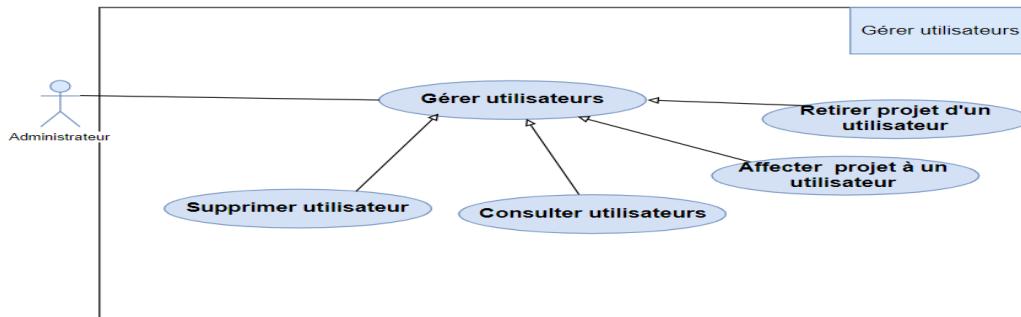


FIGURE 3.7 – Raffinement du cas d'utilisation : Gérer les utilisateurs

Le diagramme de la figure 3.7 présente un raffinement relatif au cas d'utilisation "gérer les utilisateurs". En effet, à travers la consultation de la liste des utilisateurs nous pouvons soit supprimer, soit affecter un projet à un utilisateur ou bien retirer un projet de l'utilisateur.

Description textuelle du sous cas d'utilisation : "Supprimer un utilisateur"

Acteur	Administrateur.
Description	Supprimer un compte utilisateur du système.
Préconditions	Administrateur authentifié.
Post-conditions	Utilisateur supprimé.
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur clique sur l'icône de suppression. 2. Le système affiche une modale de confirmation. 3. L'administrateur clique sur confirme son choix. 4. Le système supprime l'utilisateur correspondant et affiche un message manifestant la réussite de l'opération. 5. Le système affiche la nouvelle liste des utilisateurs.
Scénario alternatif	<ol style="list-style-type: none"> 2.a L'administrateur décide d'annuler l'opération de suppression. 2.b Le système ferme la modale et affiche la liste des utilisateurs mise à jour.

TABLE 3.4 – Description textuelle du « Supprimer un utilisateur »

3.4.2 Conception

Afin de décortiquer et détailler les cas d'utilisation précédemment cités, nous présentons dans ce qui suit les diagrammes de séquences des cas d'utilisation les plus importants

Diagramme de séquence du cas d'utilisation : "Supprimer un utilisateur"

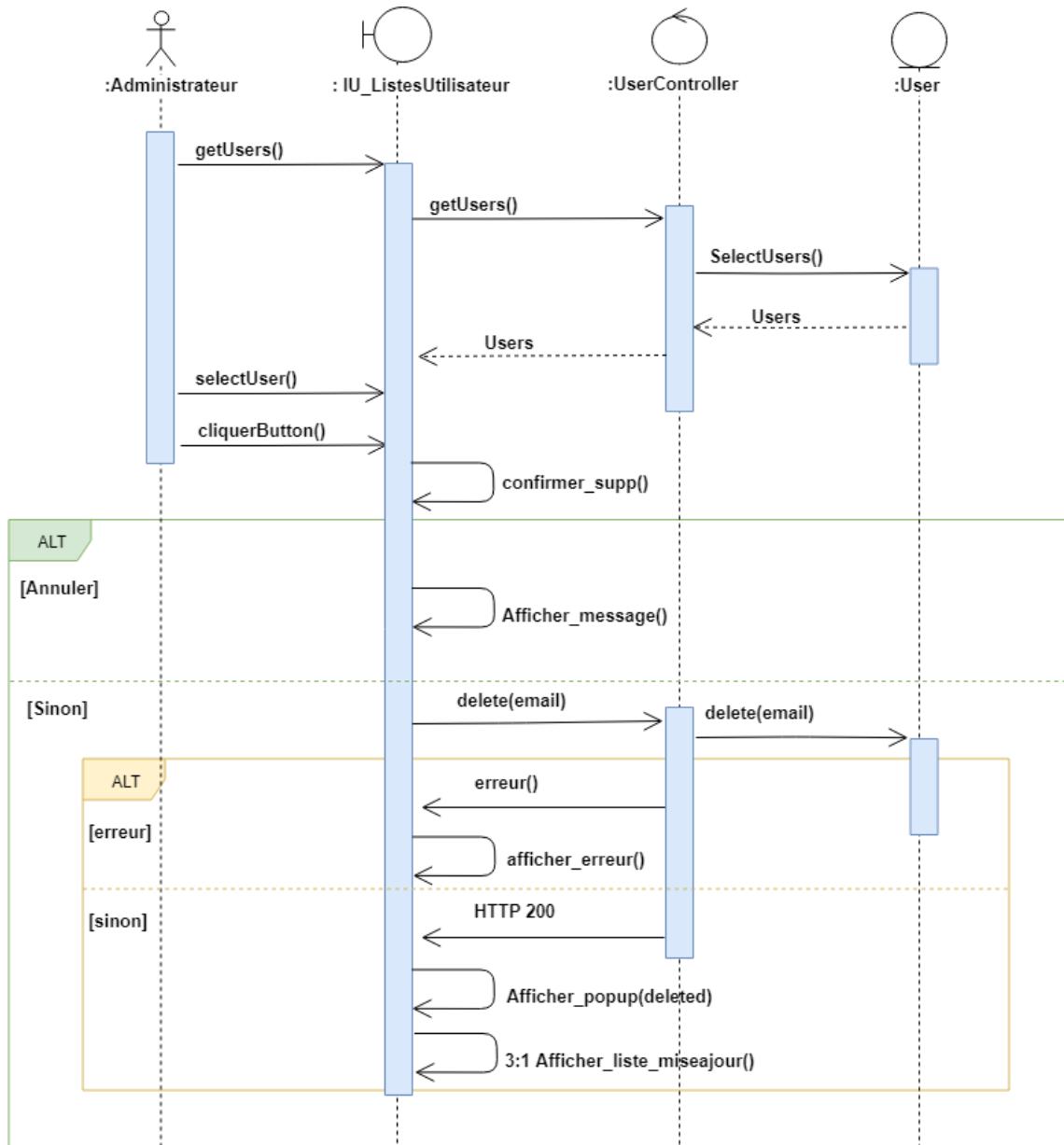


FIGURE 3.8 – Diagramme de séquence : Supprimer un utilisateur

Le diagramme de la figure 3.8 décrit l'opération de suppression d'un utilisateur. Tout d'abord, l'administrateur accède à la liste des utilisateurs. Ensuite, Il clique sur le bouton de suppression correspondant. Une fois cliqué, une modale de confirmation s'ouvre. Là où il peut soit annuler l'opération, soit la confirmer. S'il appuie sur le bouton de confirmation les données nécessaires à la suppression seront émises au contrôleur pour vérifier l'existence de l'utilisateur à partir de son email. Par la suite, le contrôleur retourne une requête http de statut 200, dans le cas où l'utilisateur existe. Il retourne une erreur, dans le cas contraire.

3.5 Réalisation

3.5.1 Sécurité

- **Authentification avec reCAPTCHA** : Permettant de vérifier l'identité annoncée, s'assurer de la non usurpation d'identité de la personne en question, empêcher les accès automatisés par les robots et toute tentative de spam.
- **Autorisation** : Permettant la vérification de priviléges de chaque utilisateur pour accéder à la ressource demandée. Dans Symfony 4, il s'agit d'un fichier de configuration "Security.yaml" réparti en sections dont chaque section assure un rôle bien important dans la sécurité
- Section "firewalls" met en évidence la méthode d'authentification elle peut être soit par login et mot de passe, soit par la méthode HTTP ou bien à travers un compte de réseau social.
- Section "access control" permet de gérer les droits d'accès des utilisateurs.
- Section "encoders" permet de spécifier l'algorithme de hashage à utiliser pour stocker le mot de passe.
- **Cryptage des mots de passe** : Pour se protéger contre les attaques et assurer la confidentialité et l'intégrité des mots de passe des utilisateurs, nous avons opté pour le chiffrement de ses derniers par l'algorithme "argon2i" à travers la section "encoders" de fichier "Security.yaml" de Symfony.
- **Sécurisation des APIs** : L'échange des informations concernant l'utilisateur couramment authentifié est sécurisé au moyen des jetons JWT pour (Json Web Token) généré par le serveur.

Structure de JWT

Ce standard de transmission des informations est composé de trois parties définies par la figure 3.5.

- Header : contient le type de jeton et l'algorithme utilisé pour la signature.
- Payload : contient toutes les données personnelles de l'utilisateur.
- Signature : hachage d'en-tête, de charge utile de la partie Payload et d'une clé secrète codée.

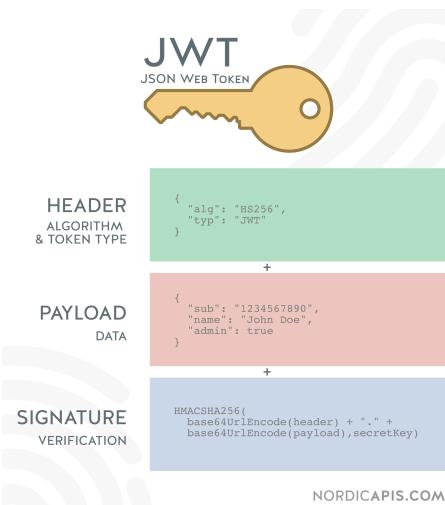


FIGURE 3.9 – Structure de JWT

Fonctionnement de JWT

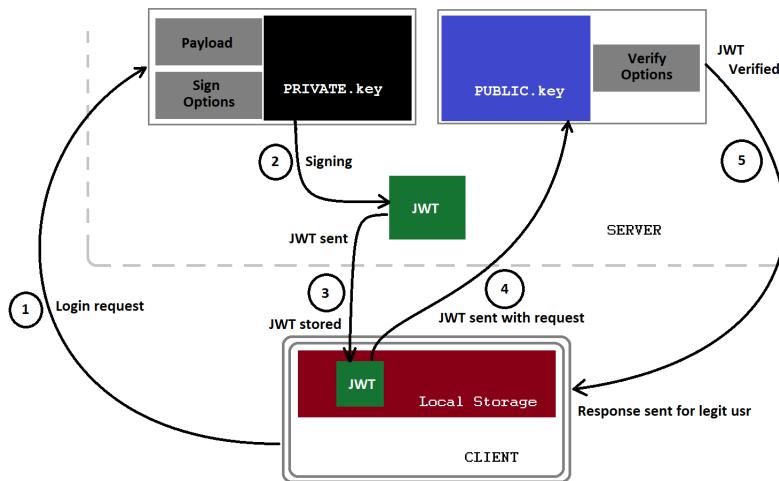


FIGURE 3.10 – Fonctionnement de JWT

Comme présenté par la figure 3.10 lorsqu'un utilisateur envoie une requête avec les paramètres requis tels que le mail d'utilisateur et son mot de passe. L'application vérifie si le mail d'utilisateur et le mot de passe sont valides. Lors de la validation, l'application créera un jeton à l'aide d'un payload et d'une clé secrète. Il renverra ensuite le jeton à l'utilisateur pour l'enregistrer à "local storage" et l'envoyer à chaque demande. Lorsque l'utilisateur envoie une requête avec ce jeton, l'application vérifie la validité avec la même clé secrète. Si le jeton est valide, la demande est envoyée, sinon l'application enverra un message d'erreur approprié.

3.5.2 Interfaces

Nous montrons le résultat du sprint courant, dans la partie qui suit, à travers des écrans d'imprimante.

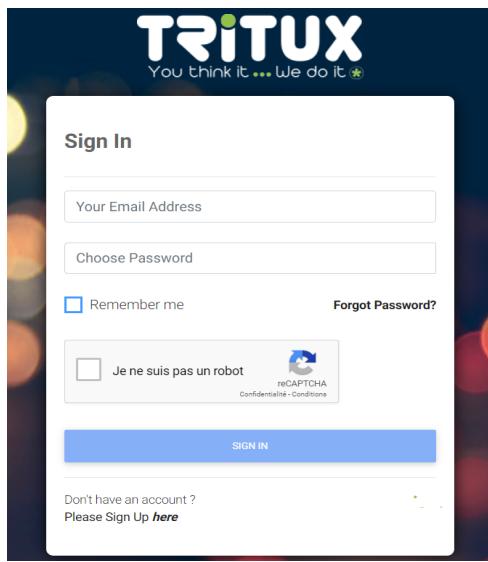


FIGURE 3.11 – Interface de connexion à l'application

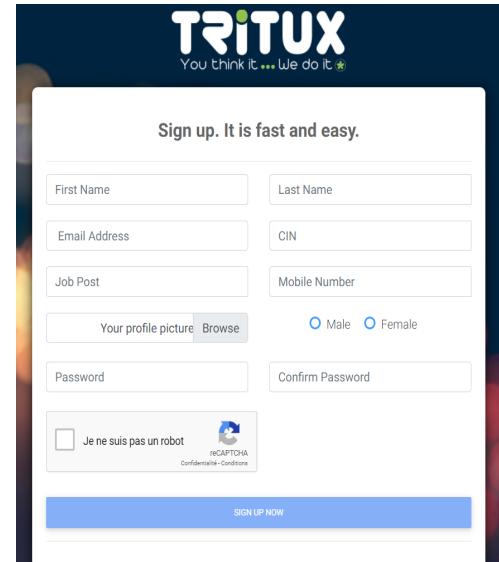


FIGURE 3.12 – Interface d'inscription

La figure 3.11 montre l'interface de l'ouverture de l'application. Si un utilisateur ne possède pas de compte il peut s'inscrire à travers la figure 3.12 ou bien s'il est déjà inscrit

mais il a oublié son mot de passe, il peut le réinitialiser à travers les interfaces décrites dans les figures 3.13 et 3.14.

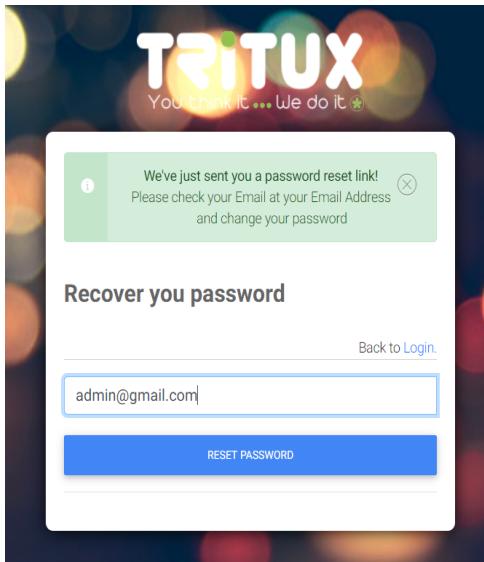


FIGURE 3.13 – Interface de réinitialisation du mot de passe



FIGURE 3.14 – Interface d'affectation d'un nouveau mot de passe en cas d'oubli

Hello **ibtissem lengliz**,

We have received your password reset request.

Please click the following link to reset your password: http://localhost:4200/auth/recoverpass/?token=a9Z3Vj_PnvXy2sp95wqUPjPVwRTVxhlon9_nxFwu5a0
Thank you,

FIGURE 3.15 – Email de réinitialisation du mot de passe

La figure 3.15 représente l'interface de gestion du profil. Dedans l'utilisateur aura la possibilité de modifier ses données personnelles comme le montre la figure 3.16.

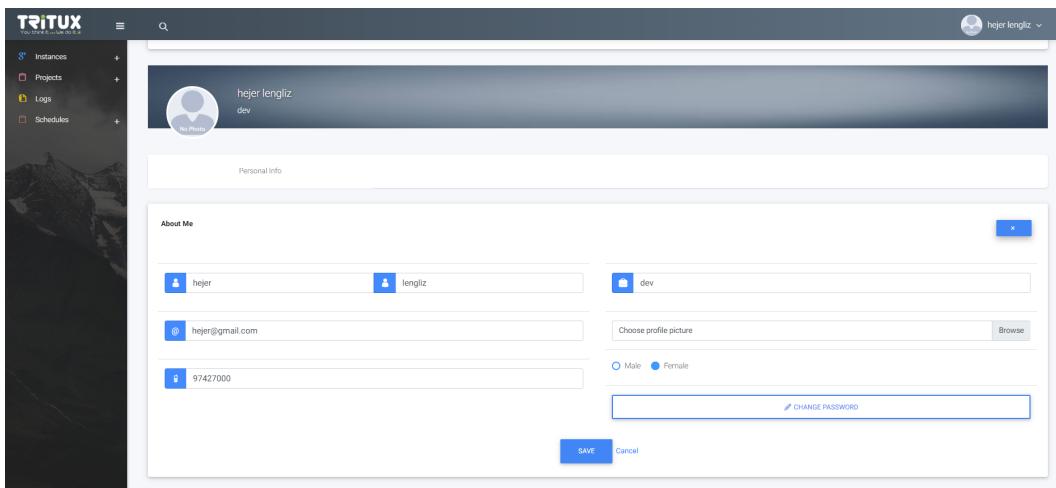


FIGURE 3.16 – Interface de mise à jour du profil

Picture	Firstname	Lastname	Email	Job Post	Projects	Actions
	Admin	Admin	admin@gmail.com	root		
	hejer	lengliz	hejer@gmail.com	dev		
	ibtissam	lengliz	ibtilengliz@gmail.com	developer		
	kais	zaatir	kais@gmail.com	dev		

Showing 1 to 4 of 4 entries

FIGURE 3.17 – Interface de gestion des utilisateurs

La figure 3.16 représente l’interface de gestion des utilisateurs, grâce à laquelle l’administrateur pourra rechercher , supprimer un utilisateur et affecter ou retirer un projet de l’utilisateur.

3.6 Conclusion

Le résultat d’un sprint est un produit livrable au client. Tout au long 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 une nouvelle valeur métier couvrant les fonctionnalités de management des projets et des machines virtuelles.

Chapitre 4

Sprint2 : Management des projets et des VMs

4.1 Introduction

L'objectif de ce chapitre est de présenter la seconde itération du cycle de vie de notre projet. Nous allons entamer par identifier les tâches à réaliser dans le Backlog du Sprint pour passer par la suite aux phases d'analyse et de conception. Nous finirons par exposer la phase de réalisation de ce module.

Nous avons découpés ce sprint en deux parties, une pour la gestion des projets et une autre pour le management des instances de VM qui est un élément majeur de notre application.

4.2 Etude fonctionnelle

4.2.1 Backlog du Sprint

Le Backlog du sprint dévoilé par le tableau 3.1 comprend une liste des travaux, de chaque user story, qui seront faites durant ce sprint.

ID	User Story	Description	Esti.
3.1	En tant que Team leader, je souhaite consulter mes projets.	Créer la "datatable" de sélection de tous les projets.	3
		Implémenter l'API de sélection des utilisateurs.	3
		Consommer l'API.	3
3.2	En tant que administrateur, je souhaite supprimer un projet.	Créer la modale de confirmation de suppression	2
		Implémenter l'API de suppression d'un utilisateur.	3
		Consommer l'API.	3

ID	User Story	Description	Esti.
3.3	En tant que administrateur, je souhaite créer un projet.	Créer la vue de création d'un projet.	2
		Générer l'API de création d'un projet.	3
		Consommer l'API.	3
3.4	En tant que administrateur, je souhaite modifier un projet.	Créer la modale de modification d'un projet.	2
		Générer l'API de modification d'un projet.	3
		Consommer l'API.	3
3.5	En tant que Team leader, je souhaite ajouter un membre à mon projet.	Créer la modale de consultation, affectation et de détachement d'un membre du projet.	2
		Générer l'API d'ajout d'un membre à un projet.	3
		Consommer l'API.	3
3.6	En tant que Team leader, je souhaite retirer un membre de mon projet.	Créer l'API de retrait d'un membre d'un projet.	3
		Consommer l'API.	3
4.1	En tant que utilisateur je souhaite consulter la liste des VMs.	Spécifier le nom de domaine de l'application auprès de Google Cloud plateform.	1
		Générer les clés nécessaires pour autoriser les requêtes auprès de Google Cloud plateform.	1
		Créer la "datatable" de sélection de toutes les instances de VMs.	3
		Générer l'API de sélection des VMs.	4
		Consommer l'API.	3
4.2	En tant que utilisateur je souhaite orchestrer le fonctionnement des VMs.	Générer l'API d'orchestration de fonctionnement des VMs.	5
		Consommer l'API.	3
4.3	En tant que Team leader je souhaite créer de nouvelles instances de machine virtuelle.	Trouver les options utilisées dans Google Cloud plateform pour créer une VM.	4
		Implémenter l'API de création d'une VM.	5
		Consommer l'API.	3

ID	User Story	Description	Esti.
4.4	En tant que administrateur, je souhaite supprimer une machine virtuelle.	Créer la modèle de confirmation de suppression.	2
		Générer l'API de suppression d'une VM.	4
		Consommer l'API.	4
4.5	4.5 En tant que administrateur, je souhaite modifier le projet d'une machine virtuelle.	Générer l'API de modification du projet d'une VM.	3
		Consommer l'API.	3

TABLE 4.1 – Backlog du Sprint 2

4.2.2 Diagramme de cas d'utilisation du sprint 2

Le diagramme de cas d'utilisation du second sprint exposé par la figure 4.1 a pour but d'établir les besoins, les résultats espérés et les missions les plus prioritaires de la seconde valeur métier.

Tous les cas d'utilisation de ce sprint sont précédés par une opération d'authentification.

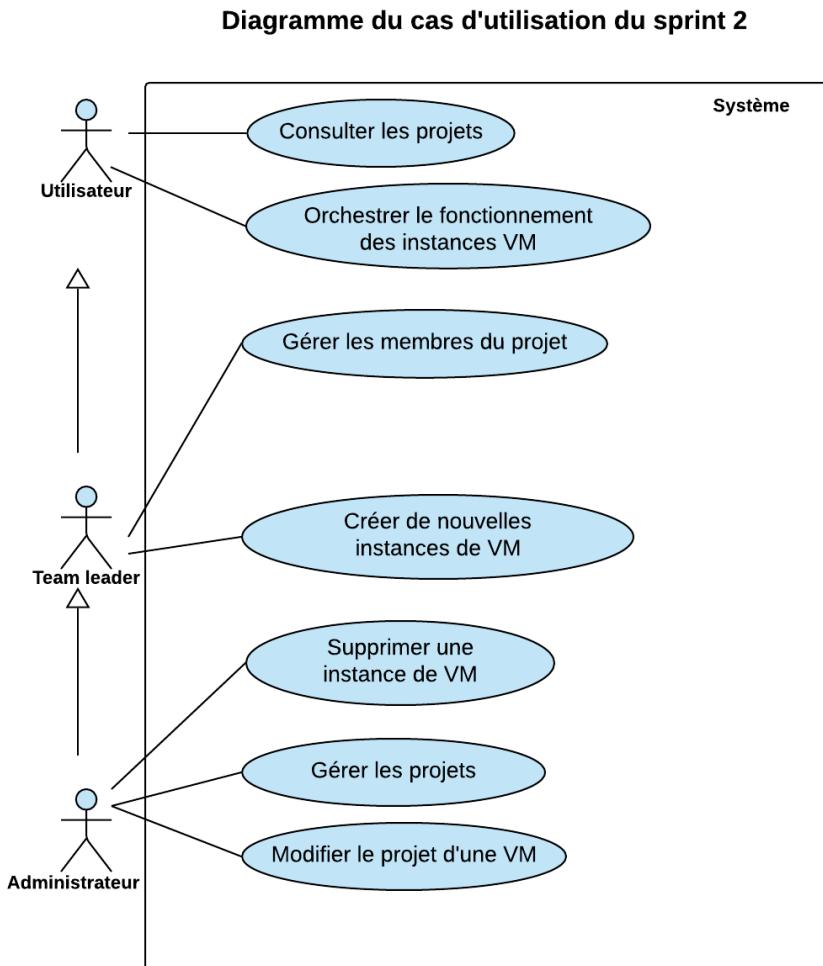


FIGURE 4.1 – Diagramme de cas d'utilisation du sprint 2

Tel qu'en atteste la figure 4.1, ce sprint permet aux utilisateurs de visualiser leurs projets et les informations appropriées. Ainsi, il les permet d'orchestrer le fonctionnement des instances de machines virtuelles. Encore plus, durant cette itération, le Team leader est apte à créer de nouvelles instances de machines virtuelles. Quant à l'administrateur, de plus que toutes les fonctionnalités déjà citées il serait capable de gérer les projets, de créer et de supprimer les VMs non utilisées.

4.3 Management des projets

Cette section est vouée au management des projets. Celui-ci est constitué par les cas d'utilisation "gérer les projets", "visualiser les projets" et "gérer les membres du projet".

4.3.1 Analyse

Pour mieux assimiler les cas d'utilisation de la première partie de ce sprint, nous allons définir, dans cette section, leurs raffinements en fournissant une description sur les scénarios.

- Raffinement du cas d'utilisation "Gérer les membres du projet"

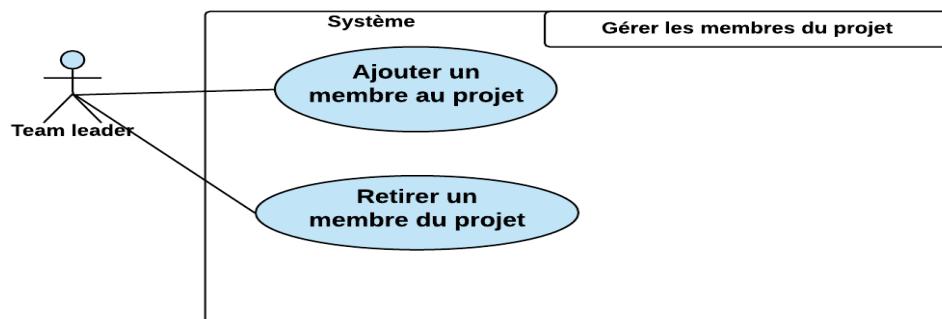


FIGURE 4.2 – Raffinement du cas d'utilisation : Gérer les membres du projet

Le diagramme de la figure 4.3 ci dessus, présente les sous cas d'utilisation qui étendent de "Gérer les membres du projet".

Description textuelle du sous cas d'utilisation : "Ajouter un membre au projet"

Acteur	Team leadder
Description	Ajouter un nouveau membre au projet.
Préconditions	Team leader authentifié.
Post-conditions	Membre ajouté.
Scénario principal	<ol style="list-style-type: none"> 1. Le Team leader remplit les champs relatifs à l'affectation d'un nouveau membre puis valide. 2. Le système vérifie l'existence de l'utilisateur saisi. 3. Le système enregistre le membre ajouté. 4. Le système affiche la liste des membres mises à jour.
Scénario alternatif	<ol style="list-style-type: none"> 1.a Le Team leader saisit un mail d'utilisateur inexistant dans le système. 1.b Le système affiche un message d'erreur.

TABLE 4.2 – Description textuelle du « Ajouter un membre au projet »

— Raffinement du cas d'utilisation "Gérer les projets"

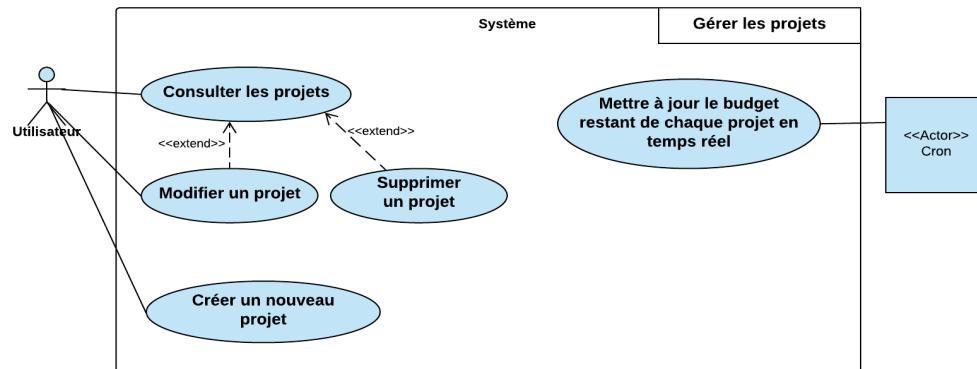


FIGURE 4.3 – Raffinement du cas d'utilisation : Gérer les projets

Le diagramme de la figure 4.3 représente un raffinement relatif au cas d'utilisation "gérer les projets". En effet, ce cas d'utilisation fait intervenir deux acteurs. L'administrateur pour la création d'un nouveau projet et/ou la consultation de la liste des projets et le Cron pour la mise à jour automatique du budget restant de chaque projet. En consultant la liste des projets, l'administrateur a la possibilité de modifier ou de supprimer un projet.

Description textuelle du sous cas d'utilisation : "Créer un nouveau projet"

Acteur	Administrateur
Description	Ajouter un projet au système.
Préconditions	Administrateur authentifié.
Post-conditions	Projet créé.
Scénario principal	1. L'administrateur remplit les champs relatifs à la création d'un nouveau projet puis valide. 2. Le système vérifie l'unicité de projet. 3. Le système ajoute le nouveau projet.
Scénario alternatif	2.a Le système affiche un message d'erreur indiquant l'existence d'un sujet pareil.

TABLE 4.3 – Description textuelle du « Créeer un nouveau projet »

4.3.2 Conception

Dans ce qui suit, nous allons détailler et décortiquer les cas d'utilisation les plus prioritaires, de la première partie de ce sprint, à travers les diagrammes de séquences objet.

Diagramme de séquence du cas d'utilisation : "Mettre à jour le budget restant de chaque projet en temps réel"

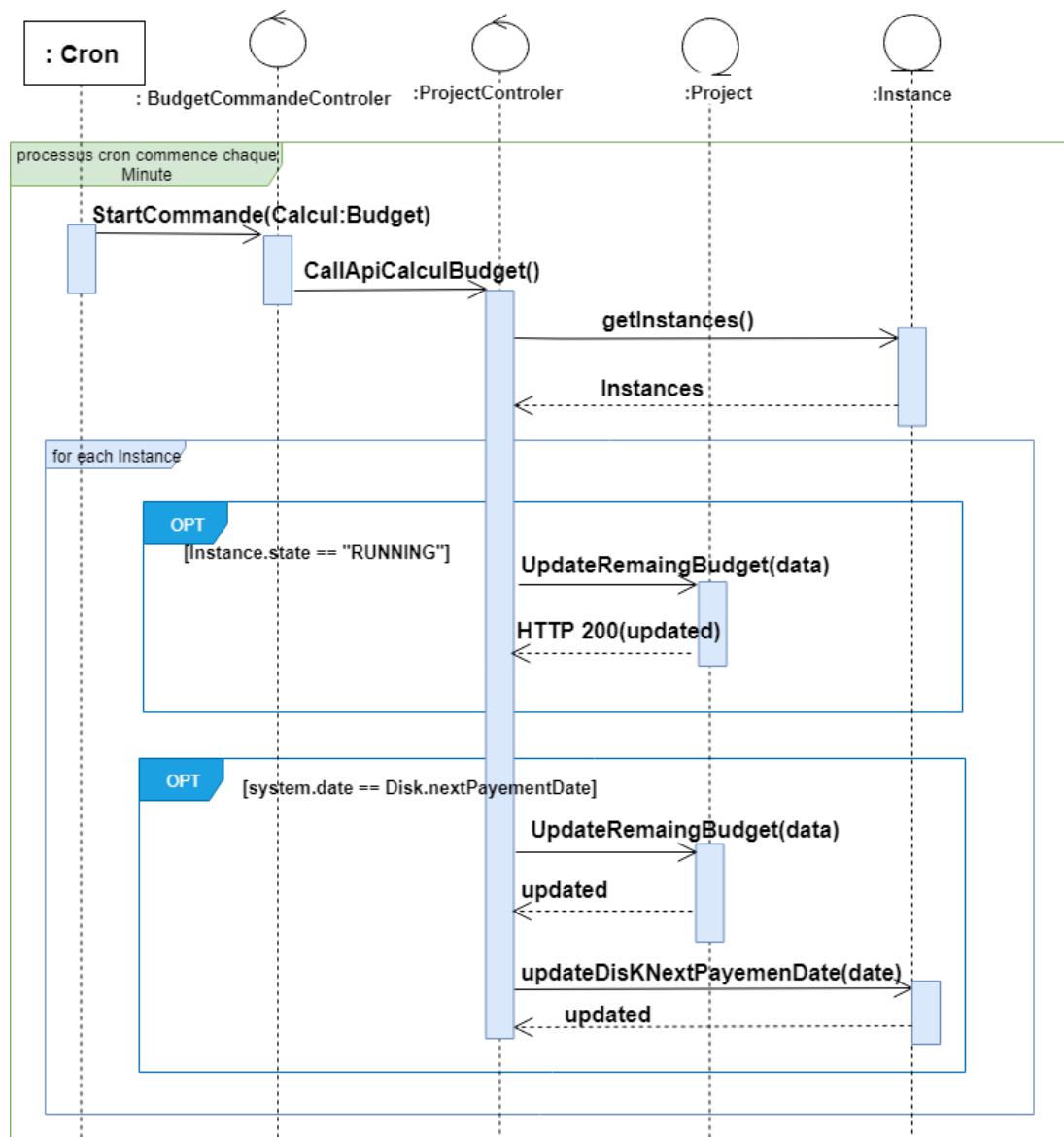


FIGURE 4.4 – Diagramme de séquence : Mettre à jour le budget restant du projet en temps réel

Le diagramme de la figure 4.4 détaille la procédure de mise à jour automatique du budget restant de chaque projet. Tout d'abord, Cron lance la commande responsable à la consommation de l'API permettant de calculer le budget restant. Cette opération se répète périodiquement chaque minute. En effet cet API consiste à sélectionner toutes les instances de VM et de faire deux tests pour chaque instance. Le premier consiste à vérifier l'état de la machine, si elle est en marche. Le budget restant sera mis à jour. Le second consiste à vérifier les dates de système et du prochain paiement. S'ils coïncident le contrôleur met à jour le budget restant et la date du prochain paiement. Vu que le paiement du disque ne s'effectue qu'une seule fois par mois.

4.4 Management des VMs

Cette section est vouée au management des instances de machines virtuelles qui sont une entité phare dans notre projet. Celui-ci est constitué par les cas d'utilisation "Orchestrer les instances de VM", "Créer de nouvelles instances de VM" , "Supprimer une VM" et "Modifier le projet d'une VM".

Toutes les tâches de cette partie nécessite une préalable configuration au près du Google Cloud plateform pour autoriser l'émission des requêtes à partir de notre application.

4.4.1 Analyse

Dans ce qui suit, nous allons offrir une vue d'ensemble des différentes fonctionnalités relatives à la deuxième partie de ce sprint. Ensuite, nous allons raffiner et décrire les cas d'utilisation les plus prioritaires.

- Raffinement du cas d'utilisation "Orchestrer les instances de VM"

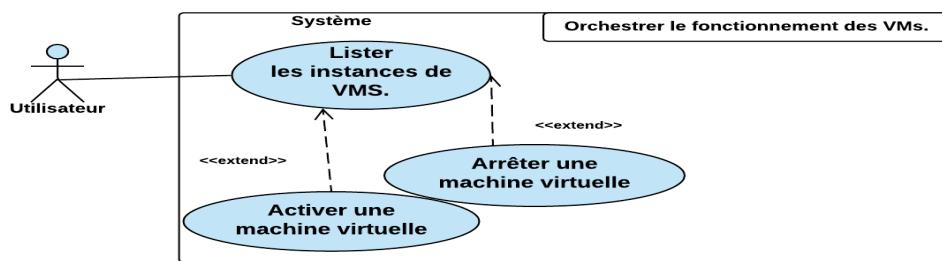


FIGURE 4.5 – Raffinement du cas d'utilisation : Orchestrer le fonctionnement des VMs

Le diagramme de cas d'utilisation de la figure 4.5 met en évidence un raffinement lié au cas d'utilisation "Orchestrer le fonctionnement des VMs". Cette opération est assurée à travers la consultation de la liste des machines virtuelles ou l'utilisateur peut activer ou arrêter une VM.

Description textuelle du sous cas d'utilisation : "Activer une machine virtuelle"

Acteur	Utilisateur.
Description	Activer une machine virtuelle.
Préconditions	1.Utilisateur authentifié. 2. Requêtes autorisées.
Post-conditions	Machine virtuelle activée.
Scénario principal	1. L'utilisateur clique sur le bouton d'activation d'une VM. 2. Le système récupère le budget restant du projet de la VM 3. Le système demande au serveur du Google Compute engine d'activer la VM. 4. Le système met à jour l'état de la VM dans la base de données.
Scénario alternatif	2.a Le budget restant du projet insuffisant pour démarrer une VM. 2.b Le système affiche un message d'erreur indiquant l'insuffisance du budget.

TABLE 4.4 – Description textuelle du « Activer une machine virtuelle »

Description textuelle du cas d'utilisation : "Lister les instances de VM"

Acteur	Utilisateur.
Description	Récupérer les instances de VM
Préconditions	1.Utilisateur authentifié.
Post-conditions	Liste des instances de VM récupérée.
Scénario principal	1. L'utilisateur demande l'accès à la liste des instances de VM. 2. Le système récupère la liste des projets auquel appartient l'utilisateur courant. 3. Le système affiche la liste des instances de VM des projets retournés
Scénario alternatif	2.a L'utilisateur n'appartient à aucun projet. 2.b Le système affiche un message indiquant que l'utilisateur n'appartient à aucun projet.

TABLE 4.5 – Description textuelle du « Lister les instances de VM »

4.4.2 Conception

Afin de décortiquer et détailler les cas d'utilisation précédemment cités, nous présentons dans ce qui suit les diagrammes de séquences des cas d'utilisation les plus importants

Diagramme de séquence du cas d'utilisation : "Créer une nouvelle instance de VM"

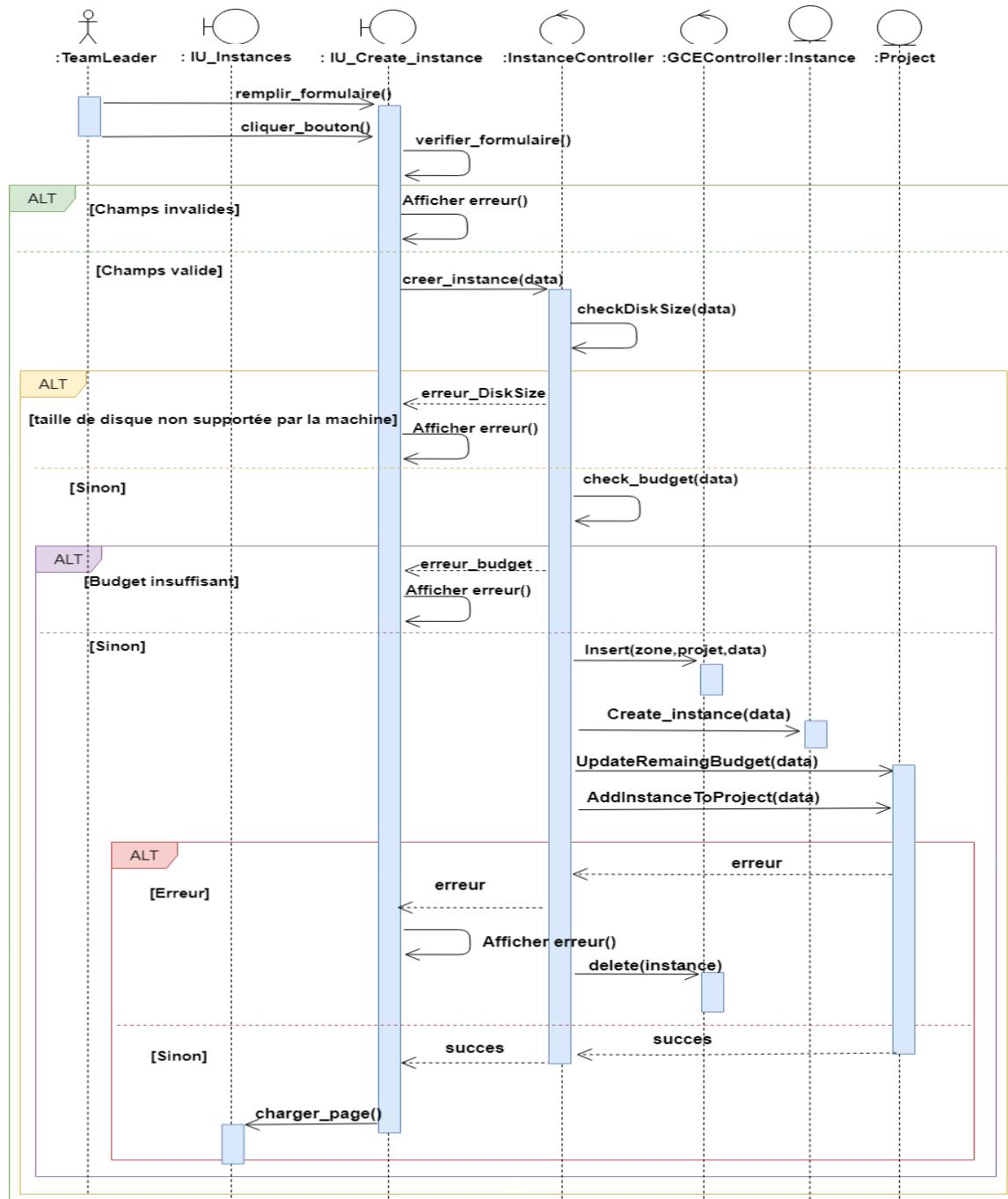


FIGURE 4.6 – Diagramme de séquence : Créer une nouvelle instance de VM

Le diagramme de la figure 4.6 illustre la procédure de création d'une instance de machine virtuelle. Tout d'abord, le Team leader remplit le formulaire puis valide et confirme. Une fois confirmé, le contrôleur vérifie la compatibilité entre la taille du disque et le type de la machine émis. S'ils sont incompatibles le contrôleur retourne une erreur, sinon il vérifie le budget restant du projet. S'il est suffisant pour créer une VM le contrôleur demande au serveur du Google Compute Engine de créer la VM puis il l'insère dans l'entité Instance et met à jour le projet associé et son budget restant. Dans le cas où une er-

reur s'est produite au niveau de l'insertion dans la base de données, le contrôleur annule l'opération en supprimant la VM de la part de Google Compute Engine afin d'assurer la synchronisation.

Diagramme de séquence du cas d'utilisation : "Modifier le projet d'une machine virtuelle"

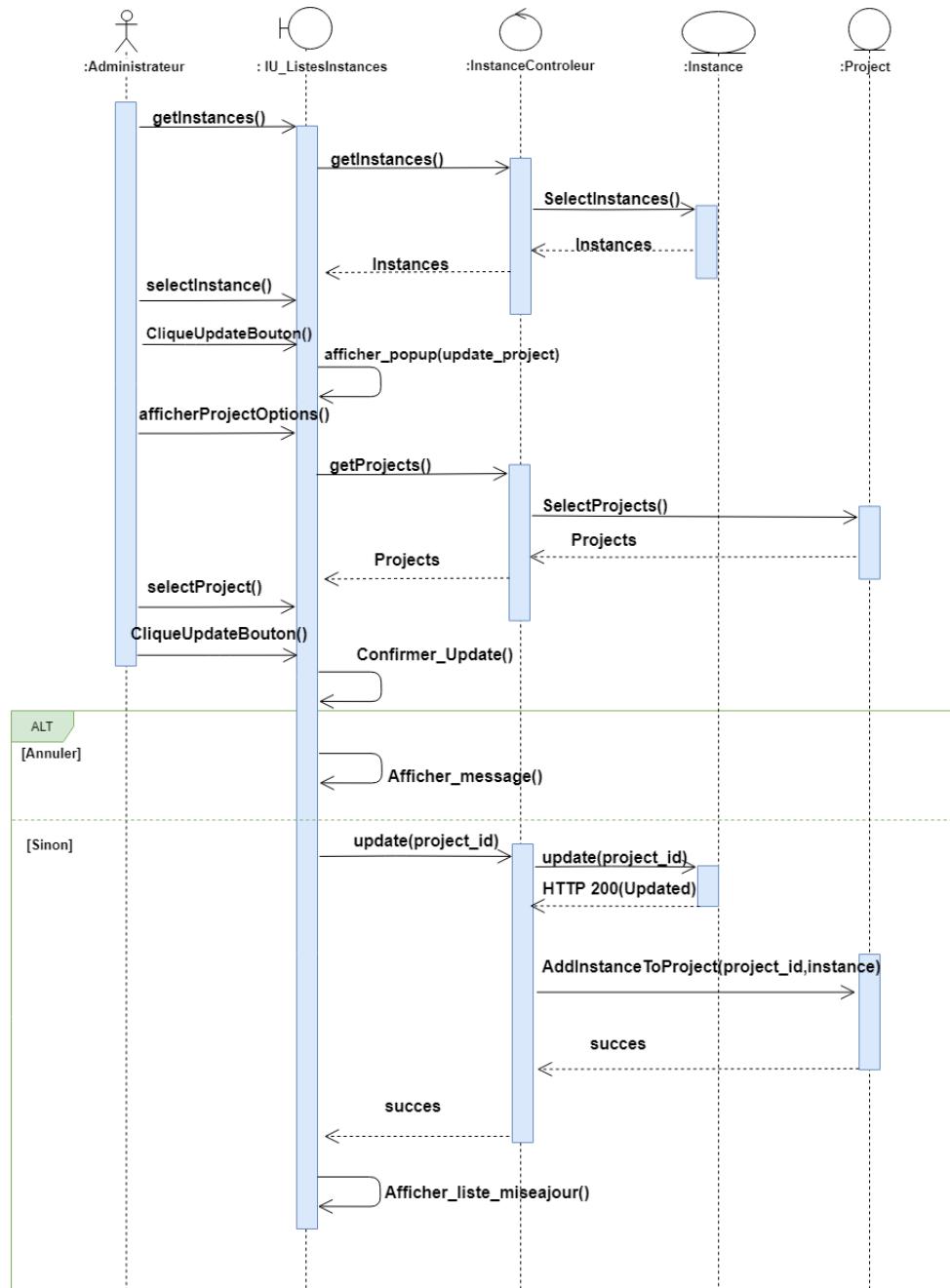


FIGURE 4.7 – Diagramme de séquence : Modifier le projet d'une machine virtuelle

Le diagramme de la figure 3.8 décrit l'opération de modification du projet d'une machine virtuelle. Tout d'abord, l'administrateur récupère toutes les instances de machine virtuelle. Ensuite, Il sélectionne la machine virtuelle adéquate et choisit le nouveau projet

puis valide. Une fois validé, une modale de confirmation s'ouvre. Là où il peut soit annuler l'opération, soit la confirmer. S'il confirme les données nécessaires à la modification seront émises au contrôleur des instances pour effectuer les changements dans les entités Instance et projet. Finalement le contrôleur retourne une requête Htp de statut 200 en cas de succès.

4.5 Réalisation

Passons à la partie réalisation, nous allons capturer quelques interfaces de ce deuxième sprint.

4.5.1 Interfaces

Name	Budget	Extra Budget	Remaining Budget	Team Leader	Collaborators	Actions	
Gestion des ressources cloud1	1000	0	499.6	admin@gmail.com			
gestion des SMS1	1500	0	0.59918198333334	admin@gmail.com			

FIGURE 4.8 – Interface de gestion des projets

La figure 4.8 représente l'interface de gestion des projets, grâce à laquelle l'administrateur pourra rechercher, supprimer, modifier un projet et créer un nouveau projet comme le montre la figure 4.9 ci-dessous.

Application d'amélioration de la qualité de vie au travail	2500
500	hejer@gmail.com

FIGURE 4.9 – Interface de création d'un nouveau projet

The screenshot shows a project management application. On the left, there's a sidebar with 'All Projects' and navigation links like 'Projects / All Projects'. The main area has a modal titled '# Collaborators' listing three users: admin@gmail.com, hejer@gmail.com, and ibtilengliz@gmail.com, each with a red 'Remove' button. Below this is a table of projects:

Name	Budget	Extra Budget	Remaining Budget	Team Leader	Collaborators	Actions
Gestion des ressources cloud1	1000	0	499.6	admin@gmail.com		
gestion des SMS1	1500	0	0.99918196333334	admin@gmail.com		

At the bottom, it says 'Showing 1 to 2 of 2 entries' and has 'Previous' and 'Next' buttons.

FIGURE 4.10 – Interface de gestion des membres du projet

La figure 4.10 ci dessus met en évidence la gestion des membres du projet. Dedans, le team leader peut affecter ou retirer des collaborateurs.

The screenshot shows a virtual machine instance management interface. On the left, there's a sidebar with 'Instances' and other navigation links like 'Projects', 'Users', 'Logs', and 'Schedules'. The main area has a table of VM instances:

Name	State	Schedules	Project	Location	Internal IP	External IP	Actions
instance-1				asia-south1-a	10.160.0.3	NULL	
instance-2				europe-west3-a	10.156.0.2	NULL	
instance-4				europe-west2-a	10.154.0.43	NULL	

At the bottom, it says 'Showing 1 to 3 of 3 entries' and has 'Previous' and 'Next' buttons. A progress bar at the bottom right indicates 'STARTING VM INSTANCE IN PROGRESS'.

FIGURE 4.11 – Interface de gestion des instances de machine virtuelle

La figure 4.11 représente l'interface de gestion des machines virtuelles, grâce à laquelle l'administrateur pourra rechercher , supprimer , orchestrer le fonctionnement des VMs , affecter un planning, modifier le projet de chaque VM et créer une machine virtuelle comme le montre la figure 4.12. Ce dernier il peut aussi consulter les détails de chaque instance de machine virtuelle grâce à la figure 4.13.

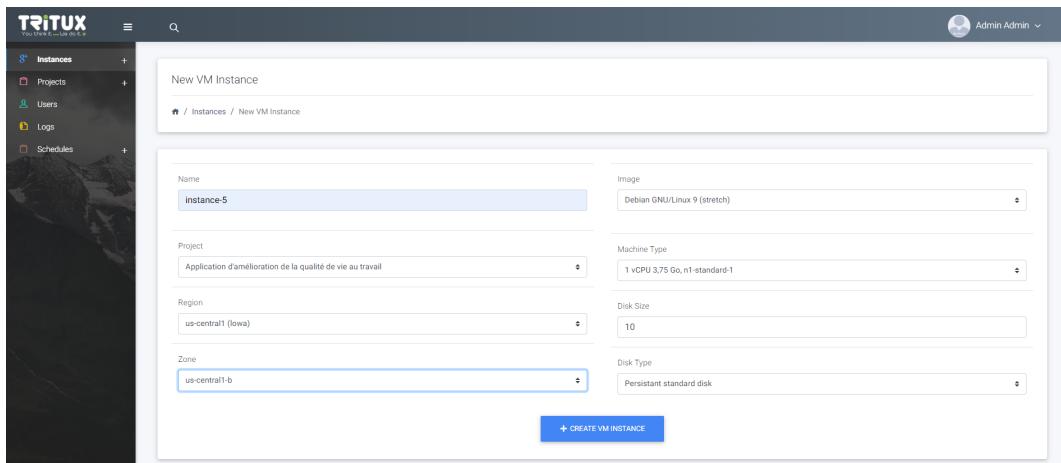


FIGURE 4.12 – Interface de création d'une nouvelle instance de machine virtuelle

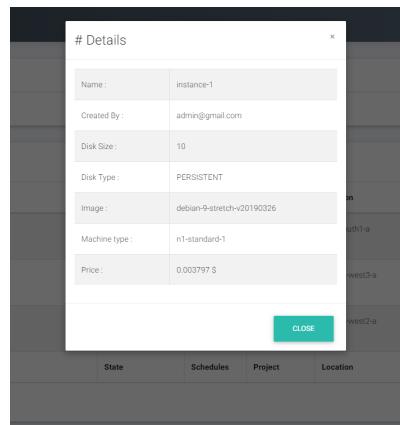


FIGURE 4.13 – Interface de détails des instances de machine virtuelle

4.6 Conclusion

Au cours de ce sprint nous avons pu détaillé les principaux fonctionnalités de la partie la plus importante de notre application qui est la gestion des machines virtuelle et des projets. Cette partie met en évidence la contrôle des VMs par projet et son budget. Dans le chapitre suivant nous allons attaquer avec la même approche le troisième sprint.

Chapitre 5

Sprint 3 : Gestion des planifications et logs

5.1 Introduction

L'objectif de ce chapitre est de présenter la dernière itération du cycle de vie de notre projet. Nous allons entamer par identifier les tâches à réaliser dans le Backlog du Sprint pour passer par la suite aux phases d'analyse et de conception. Nous finirons par exposer la phase de réalisation de ce module.

Nous avons décidé de découper ce sprint en deux parties, une partie pour la gestion des planifications et une autre partie pour les logs qui permettent à l'administrateur d'avoir une vision globale sur l'activité des utilisateurs.

5.2 Etude fonctionnelle

5.2.1 Backlog du Sprint

Le Backlog du sprint présentés par le tableau 3.1 contient une liste des tâches de chaque user story identifiées par l'équipe Scrum et qui devront être réalisées pendant ce sprint.

ID	User Story	Description	Esti.
5.1	En tant que utilisateur, je souhaite consulter la liste des plannings.	Créer la "datatable" de sélection de tous les plannings.	2
		Créer l'API de sélection des plannings.	3
		Consommer l'API	3
5.2	En tant que utilisateur, je souhaite créer un planning.	Implémenter la vue de création d'un planning.	5
		Générer l'API de création d'un planning.	4
		Consommer l'API.	3

ID	User Story	Description	Esti.
5.3	En tant que utilisateur, je souhaite supprimer un planning.	Créer la modale de confirmation de suppression d'un planning	3
		Créer l'API de suppression d'un planning.	3
		Consommer l'API.	3
5.4	En tant que utilisateur, je souhaite affecter un planning à une VM.	Mettre en place l'outil Cron.	2
		Créer la modale de consultation, affectation et de détachement d'un planning d'une VM.	5
		Créer l'API d'affectation d'un planning à une VM.	4
		Consommer l'API.	3
5.5	En tant que utilisateur, je souhaite retirer un planning d'une VM.	Créer l'API de retrait d'un planning d'une VM.	3
		Consommer l'API.	3
6.1	En tant que utilisateur, je souhaite consulter les activités des utilisateurs.	Créer la "datatable" de sélection des logs.	3
		Créer l'API de sélection des logs.	3
		Consommer l'API.	3

TABLE 5.1 – Backlog du Sprint 3

5.2.2 Diagramme de cas d'utilisation du sprint 3

Le diagramme de cas d'utilisation du troisième sprint exposé par la figure 5.1 a pour but d'établir les besoins, les résultats espérés et les missions les plus prioritaires de la troisième valeur métier.

Tous les cas d'utilisation de ce sprint sont précédés par une opération d'authentification.

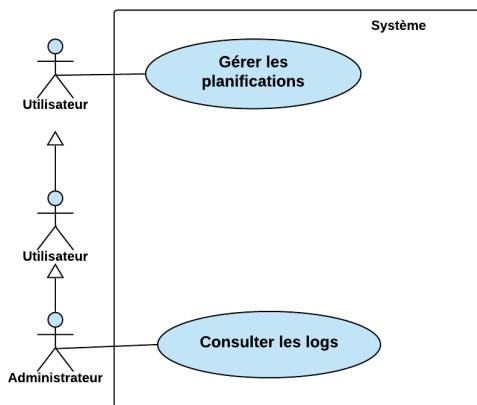


FIGURE 5.1 – Diagramme de cas d'utilisation du sprint 3

Tel qu'en atteste la figure 5.1, ce sprint permet aux utilisateurs de gérer les planifications.

Ainsi, il permet à l'administrateur de consulter les logs.

5.3 Gestion des planifications

Cette section est dédiée à la gestion des planifications. Elle est constituée par le cas d'utilisation "Gérer les planifications".

La réalisation de ce cette partie nécessite l'utilisation du planificateur de tâches Cron. Ce dernier est un utilitaire linux qui planifie l'exécution automatique d'une commande ou d'un script sur le serveur à une date et une heure spécifiées. Il est utile pour automatiser des tâches répétitives. Dans notre cas, il va planifier le fonctionnement des machines virtuelles.

5.3.1 Analyse

Afin de mieux détailler le fonctionnement et assimiler les cas d'utilisation de ce sprint, nous allons établir, dans cette section, leurs raffinements en livrant une description sur les scénarios les plus importants.

— Raffinement du cas d'utilisation "Gérer les planifications"

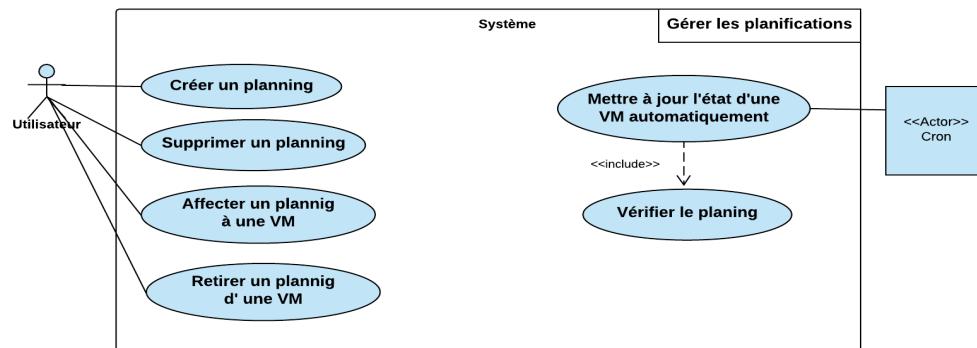


FIGURE 5.2 – Raffinement du cas d'utilisation : Gérer les planifications

Comme le montre la figure 5.2, la création, suppression d'un planning et son retrait et affectation à une VM sont tous des sous cas d'utilisation qui s'étendent de la gestion des planifications. En outre, la mise à jour automatique de l'état de la VM avec une vérification du planning affecté sont des cas d'utilisations assurés à travers notre second acteur qui est le Cron.

Description textuelle du cas d'utilisation : "Supprimer un planning

5.3.2 Conception

Afin de décortiquer et détailler les cas d'utilisation précédemment cités, nous présentons dans ce qui suit les diagrammes de séquences des cas d'utilisation les plus importants

Acteur	Utilisateur
Description	Supprimer un planning.
Préconditions	Utilisateur authentifié
Post-conditions	Planning supprimé.
Scénario principal	<ol style="list-style-type: none"> L'utilisateur sélectionne le planning et clique sur le bouton de suppression. Le système affiche une modale de confirmation. L'utilisateur confirme son choix. Le système supprime le planning correspondant.
Scénario alternatif	2.a L'utilisateur décide d'annuler son choix.

TABLE 5.2 – Description textuelle du « Supprimer un planning »

Diagramme de séquence du cas d'utilisation : "Mettre à jour l'état d'une machine virtuelle automatiquement"

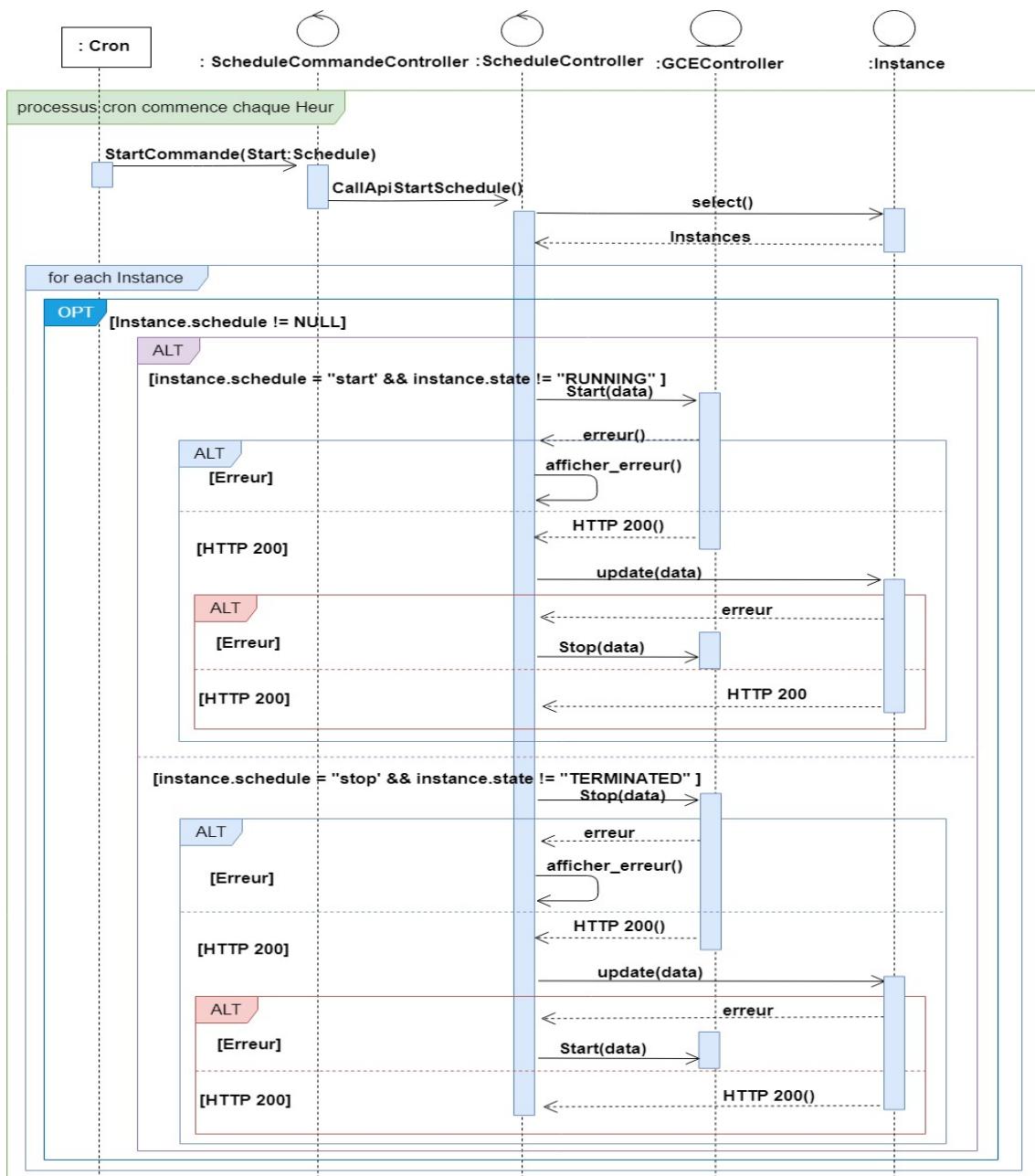


FIGURE 5.3 – Diagramme de séquence : Mettre à jour l'état d'une machine virtuelle automatiquement

Le diagramme de la figure 5.3 détaille la procédure relative à la mise à jour automatique de l'état des VMs. Cette opération nécessite la mise en place de l'outil Cron de linux. Chaque heure, cron demande la consommation de l'API responsable au fonctionnement automatique des machines virtuelles. Au moment de l'exécution de cet API, le contrôleur des planifications récupère la liste des instances et pour chaque instance qui a un planning il vérifie si elle est en état de repos et si son planning l'oblige pour qu'elle soit activée. Si c'est le cas, le contrôleur émet une requête au Google Compute engine pour l'activer et une autre pour changer son état dans la base de données. Dans le cas où une erreur s'est produite au niveau de la mise à jour dans la base de données et pour en assurer la synchronisation le contrôleur annule l'opération en arrêtant la VM auprès de Google Compute Engine. Pareil pour l'arrêt automatique d'une machine mise en marche.

Diagramme de séquence du cas d'utilisation : "Créer un planning"

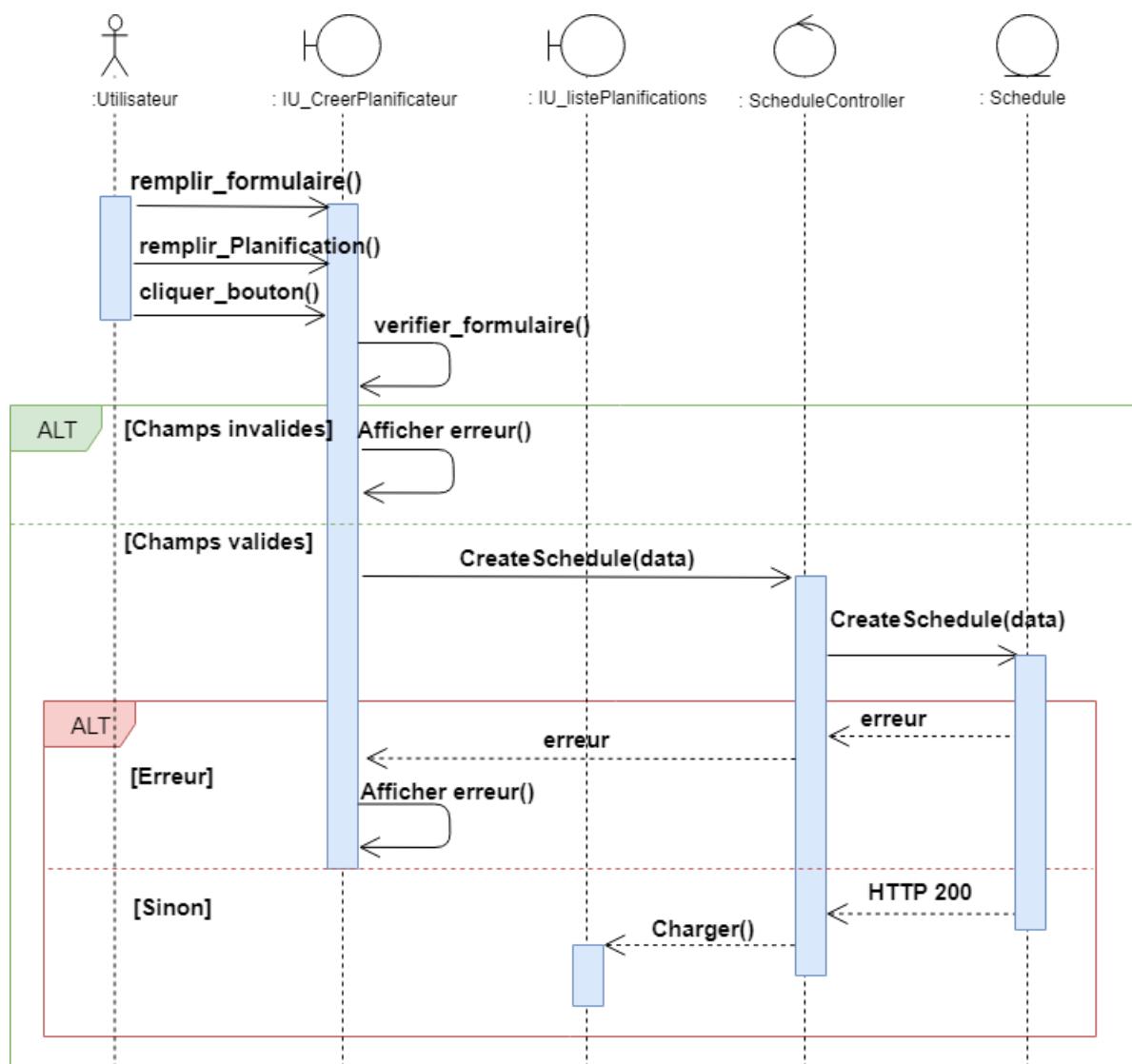


FIGURE 5.4 – Diagramme de séquence : Créer un planificateur

Le diagramme de la figure décrit l'opération de création d'un planning. Tout d'abord, l'utilisateur remplit le formulaire correspondant et choisit l'horaire du fonctionnement de

la machine virtuelle puis valide. Une fois validé, le contrôleur émet une requête de création d'un planning. Il retourne une requête de statut 200 si la création est effectuée avec succès et une erreur dans le cas contraire.

5.4 Gestion des logs

Cette section est dédiée à la gestion des logs. Elle est constituée par le cas d'utilisation "Consulter les logs".

5.4.1 Analyse

Dans ce qui suit, nous allons offrir une vue d'ensemble des fonctionnalités relatives à la deuxième partie de ce sprint.

Description textuelle du cas d'utilisation : "Consulter les logs"

Acteur	Administrateur.
Description	Afficher les logs.
Préconditions	Administrateur authentifié.
Post-conditions	Liste des logs affichée.
Scénario principal	<ol style="list-style-type: none"> L'administrateur demande la page des logs. Le système charge les données et les retourne.
Scénario alternatif	2. Le système retourne une erreur du chargement de la page

TABLE 5.3 – Description textuelle du « Consulter les logs »

5.4.2 Conception

Afin de décortiquer et détailler le cas d'utilisation précédemment cité, nous présentons dans ce qui suit sa conception à travers le diagramme de séquences objet.

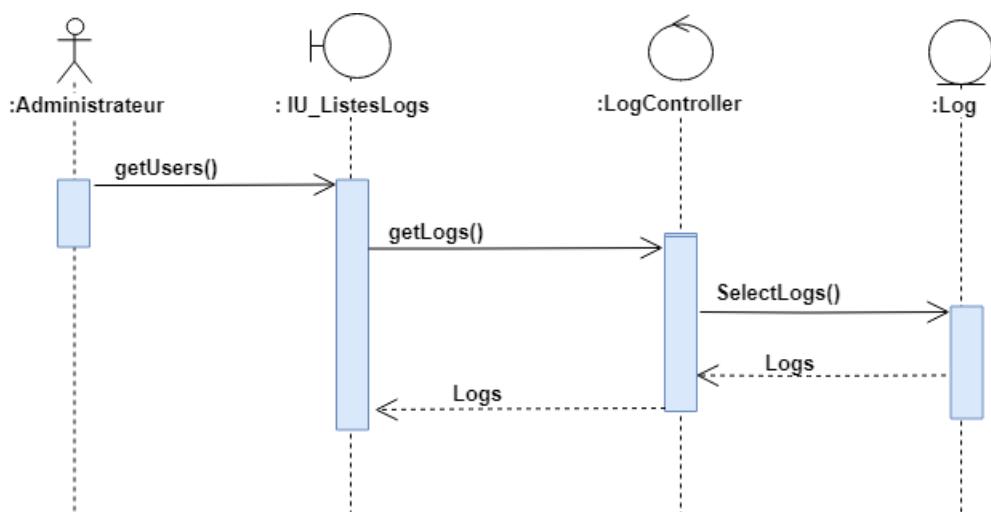


FIGURE 5.5 – Diagramme de séquence : Consulter les logs

Le diagramme de la figure décrit l'opération de sélection des logs. En effet, l'administrateur charge la pages contenant la liste des logs. A ce moment, le contrôleur sélectionne et retourne la liste des logs.

5.5 Réalisation

Après avoir achevé l'étape de la conception de ce sprint, en tenant compte des besoins fixés et des choix conceptuels effectués précédemment, nous consacrons cette section à la description du travail réalisé.

5.5.1 Interfaces

Name	Description	Instances	Actions
from 08 to 18		[Green]	[Red]
Night Time	From 17 to 22	[Green]	[Red]
schedule3	description3	[Green]	[Red]
Work Time	from 09 to 13 and from 16 to 18	[Green]	[Red]

FIGURE 5.6 – Interface de gestion des planifications

La figure 5.6 représente l'interface de gestion des planifications, grâce à laquelle l'utilisateur pourra rechercher, supprimer et créer un nouveau planning comme le montre la figure 5.7 ci-dessous.

FIGURE 5.7 – Interface de création d'un planning

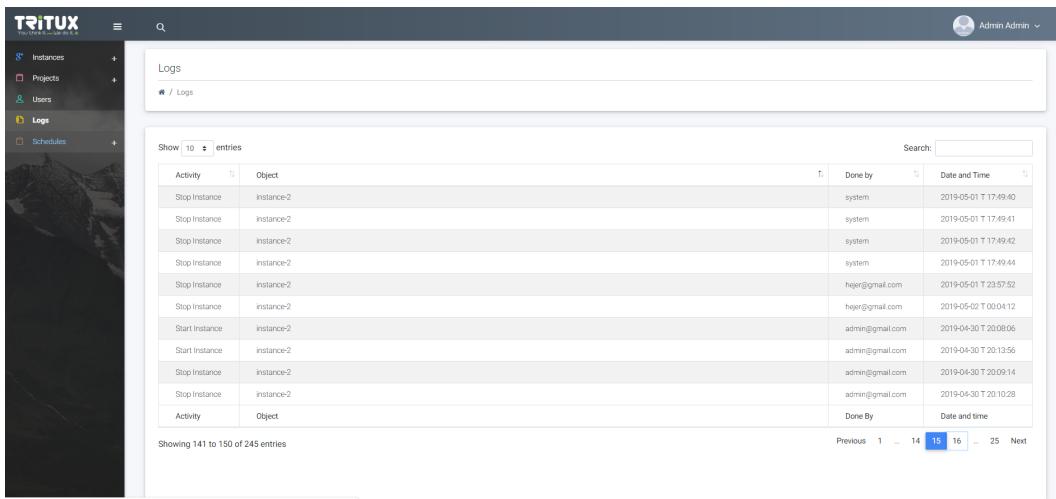


FIGURE 5.8 – Interface de consultation des logs

La figure 5.8 ci dessus, représente l'interface grâce à laquelle l'administrateur peut accéder et consulter les logs de toutes les activités des utilisateurs dans l'application.

5.6 Conclusion

Nous clôturons ici la réalisation du dernier sprint de notre application. Nous avons pu réaliser les logs grâce auquel l'administrateur aura une vue global des activités de ses collaborateurs ainsi que la partie gestion des planifications pour automatiser le fonctionnement des machines virtuelles.

Conclusion Général

Le présent document est une présentation du travail réalisé durant notre stage de fin d'études au sein de l'entreprise Tritux. Nous avons débuté par comprendre le contexte général du projet et les différentes exigences du futur système. Nous avons préparé, par la suite, un planning de travail en respectant les priorités des besoins déjà fixés

Malgré les contraintes de temps et les difficultés techniques que nous avons rencontré qui se résument principalement dans la complexité du projet, nous avons réussi à réaliser la totalité de « Gestion des ressources Cloud par projet et budget »

Le travail dans le cadre de ce projet de fin d'études, était d'une importance considérable dans la mesure où il nous a servi comme portail vers le monde professionnel et la vie d'entreprise.

De point de vue technique, il nous a permis de mettre en œuvre les acquis théoriques que nous avons appris tout au long de notre cursus universitaire et de les enrichir. Outre, ce projet était aussi enrichissant pour les bonnes pratiques de la gestion de projet vu que nous avons eu l'opportunité d'organiser son déroulement dès le début.

Loin du gain académique, ce stage nous a permis de mesurer notre capacité à apprendre et à entreprendre dans un court délai.

Finalement, notre travail ne s'arrête pas à ce niveau. En effet, plusieurs perspectives s'offrent à notre application. Parmi les fonctionnalités que nous pouvons envisager pour « Gestion des ressources Cloud par projet et budget » :

- S'ouvrir à plusieurs fournisseurs Cloud tels que AWS et Azure.
- Rajouter un volet de estimation du gain pour chaque planning utilisé.
- Rajouter un dashboard pour offrir une vue globale sur les différentes fonctionnalités.

Bibliographie

- [AV00] M. Agusti and J. M. Valiente. Feasibility of using wavelet based pyramidal analysis for visual content image description. In *Proceedings of the 3rd IASTED International Conference Visualization, Imaging and Image*, volume 2, pages 569–574, 2000.
- [Dup09] Nestor Dupont. *Reparer son vaisseau*. L’Alliance, 2009.
- [JN53] Edgar G. Johnson and Alfred O. Nier. Angular aberrations in sector shaped electromagnetic lenses for focusing beams of charged particles. *Physical Review*, 91(1), jul 1953.
- [Knu90] Donald E. Knuth. *The TeXbook*. Addison -Wesley, 1990.
- [Lav90] L. Laverdure. Gravimetrie de la ceinture volcanique de l’Abitibi. Master’s thesis, Ecole Polytechnique de Montreal, Montreal, Qc, Canada, 1990.
- [LLC99] MultiMedia LLC. MS Windows NT kernel description, 1999.
- [Rac96] Zoran Racic. *Étude et essais du spectromètre à plasma DYMIO de la mission MARS 96*. PhD thesis, 1996.
- [Rol99] Christian Rolland. *ETEX par la pratique*. O'Reilly, 1999.
- [YI05] T. Yamada and T. Inoue. Influence of phosphorus runoff from agricultural area-sion enclosed sea downsteram. *Journal of Water and Environment Technology*, 3(2) :157–164, 2005.
- [Zer09] Les Zeros. Le site du zero, jun 2009. www.siteduzero.com.

Annexe 1