

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDERRAHMANE MIRA DE BÉJAÏA



جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

FACULTÉ DES SCIENCES EXACTES
DÉPARTEMENT D'INFORMATIQUE

MÉMOIRE DE MASTER PROFESSIONNEL

OPTION : GÉNIE LOGICIEL

Thème

CONCEPTION ET RÉALISATION D'UNE APPLICATION MOBILE POUR LA GESTION DES BILANS, CAS D'ÉTUDE ” L'UNITÉ DE RECHERCHE *LaMOS* DE BÉJAÏA ”

Réalisé par :

Melle KEBBI Sonia

Devant le jury composé de :

<i>Président</i>	Mr AMROUN K.	U. A/MIRA BÉJAÏA
<i>Examinatrice</i>	Mme EL BOUHISSI H.	U. A/MIRA BÉJAÏA
<i>Encadrant</i>	Mme ADEL-AISSANOU K.	U. A/MIRA BÉJAÏA
<i>Co-Encadrant</i>	Mr ATMANI M.	U. A/MIRA BÉJAÏA

Promotion 2019-2020

DÉDICACES

Je dédie ce modeste travail ...

A mes chers parents

En signe de reconnaissance de l'immense bien que vous avez fait pour moi concernant mon éducation qui aboutit aujourd'hui à la réalisation de ce travail. Recevez à travers ce dernier, toute ma gratitude et mes profonds sentiments. Que Dieu le tout puissant soit à vos côtés et vous accorde une meilleure santé

Inchallah.

A mes très chers frères et sœurs

Samir , Saliha , M'hand et Yacine

A mes très chers neveux et nièces que j'aime autant

Melina , Adam et Yani

A mes très chers oncles, tantes, cousins et cousines

A mes chers amis

Numidia , Nesrine, Thiziri , Yasmine et tout les autres

A toutes les personnes que j'apprécie et que je n'ai pas citées

REMERCIEMENTS

Mes vifs remerciements vont d'emblée à Dieu le tout puissant qui m'a doté d'une grande volonté et d'un savoir adéquat pour mener à bien cet humble travail.

j'adresse mes remerciements tout particulièrement : Á nos chers parents, ma fierté et la source de ma réussite car ils se sont sacrifiés pour me fournir une atmosphère de travail disposant de toutes les meilleures conditions, sans eux rien n'aurait pu être facile, que dieux me les garde et les protège afin que l'on puisse leurs rendre un peu du beaucoup qu'ils nous ont procuré.

Á mon Encadreur Mme ADEL Karima pour sa disponibilité, son savoir-faire et son soutien.

Aux membres du jury qui ont accepté d'évaluer notre travail.

Á mes très chers frères, soeurs, ami (es) et à toutes les personnes qui ont contribué à la réussite de ce travail.

Table des matières

Table des matières	iv
Table des figures	vi
Liste des tableaux	vii
Glossaire	viii
Introduction Générale	1
1 Présentation de l'organisme d'accueil	3
1.1 Introduction	3
1.2 Présentation de LaMOS (Laboratoire de Modélisation et Optimisation des Systèmes)	3
1.2.1 Définition	3
1.2.2 Historique	3
1.2.3 Organigramme de l'organisme d'accueil	4
1.2.4 Structure de l'unité de recherche LaMOS	4
1.3 Problématique	6
1.4 Solutions proposées	6
1.5 Méthodologie de conception	7
1.5.1 Méthologie agile	7
1.5.2 Pourquoi Scrum	7
1.5.3 Définition de Scrum	7
1.5.4 Concepts de base de la méthode de gestion de projet Scrum	9
1.5.5 Présentation de la méthode Scrum	12
1.6 Langages de modélisation	13
1.6.1 UML (Unified Modeling Language)	13
1.6.2 Différents types de diagrammes UML	14
1.7 Conclusion	15

TABLE DES MATIÈRES

2	Généralités sur les applications mobiles	16
2.1	Introduction	16
2.2	Informatique mobile	16
2.2.1	Présentation	16
2.2.2	Terminaux mobiles	16
2.3	Applications mobiles	19
2.3.1	Définition	19
2.3.2	Avantages d'une application mobile	19
2.3.3	Inconvénients d'une application mobile	19
2.3.4	Différents types d'application mobile	20
2.4	Système d'exploitation Android	20
2.4.1	Présentation	20
2.4.2	Historique	21
2.4.3	Architecture	23
2.5	Applications Android	25
2.5.1	Présentation	25
2.5.2	État d'une activité (application)	26
2.5.3	Cycle de vie d'une activité Android	26
2.6	Conclusion	28
3	Préparation de projet	29
3.1	Introduction	29
3.2	Spécification des besoins	29
3.2.1	Besoins fonctionnels	29
3.2.2	Besoins non fonctionnels	30
3.3	Modélisation du besoin	31
3.3.1	Identification des acteurs	31
3.3.2	Diagramme de cas d'utilisation global	31
3.4	Pilotage du projet avec Scrum	33
3.4.1	Équipe et rôle	33
3.4.2	Backlog du produit	33
3.4.3	Planification de Release	36
3.5	Environnement de travail	37
3.5.1	Environnement matériel	38
3.5.2	Environnement Logiciel	38
3.6	Architecture de l'application	40
3.7	Conclusion	42

TABLE DES MATIÈRES

4	Release 1	43
4.1	Introduction	43
4.2	Développement du Sprint 1 « Authentification, Gestion des activités et génération du bilan d'un membre d'équipe »	43
4.2.1	Analyse	45
4.2.2	Conception	50
4.2.3	Réalisation	57
4.3	Développement du sprint 2 « gestion des membres d'équipe et génération du bilan de l'équipe »	60
4.3.1	Analyse	61
4.3.2	Conception	65
4.3.3	Réalisation	68
4.4	Conclusion	70
5	Release 2	71
5.1	Introduction	71
5.2	Développement du sprint 1 « Gestion des équipes et génération du bilan de l'unité »	71
5.2.1	Analyse	72
5.2.2	Conception	76
5.2.3	Réalisation	79
5.3	Développement du sprint 2 « gestion des sorties pédagogiques et la gestion des activités visiteur »	81
5.3.1	Analyse	82
5.3.2	Conception	86
5.3.3	Réalisation	89
5.4	Conclusion	91
	Conclusion Générale et Perspectives	92
	Bibliographie	93
A	Diagramme de séquence système	i
A.1	Diagramme de séquence système " Gérer les membres d'équipe "	i
A.1.1	Diagramme de séquence système " Ajouter un membre d'équipe "	ii
A.1.2	Diagramme de séquence système " Modifier un membre d'équipe "	iii
A.1.3	Diagramme de séquence système " Supprimer un membre d'équipe "	iv
A.1.4	Diagramme de séquence système " Rechercher un membre d'équipe "	v
A.1.5	Diagramme de séquence système " Générer le bilan d'un membre d'équipe "	vi
A.2	Diagramme de séquence système " Gérer les activités "	vii
A.2.1	Diagramme de séquence système " Ajouter une activité "	vii

TABLE DES MATIÈRES

A.2.2	Diagramme de séquence système " Modifier une activité "	viii
A.2.3	Diagramme de séquence système " Supprimer une activité "	ix
A.2.4	Diagramme de séquence système " Rechercher une activité "	ix
A.3	Diagramme de séquence système " Générer le bilan de l'unité "	x
A.4	Diagramme de séquence système " S'authentifier "	xi
B	Présentation des interfaces	xii
B.1	Interface "Profil d'un Membre d'équipe "	xii
B.2	Interface "Profil d'un Chef d'équipe "	xiii
B.3	Interface "Profil d'un Directeur "	xiv

Table des figures

1.1	Organigramme de l'unité de recherche LaMOS [1].	4
1.2	Cycle de vie de Scrum [39].	13
1.3	Types de diagrammes UML [40]	14
2.1	Un assistant numérique personnel [41]	17
2.2	Les smartphones [42]	18
2.3	Les tablettes [43]	18
2.4	Architecture de la plateforme Android [18]	23
2.5	La machine virtuelle Dalvik	24
2.6	Cycle de vie d'une activité Android [16]	28
3.1	Diagramme de cas d'utilisation global	32
3.2	Schéma de Releases	37
3.3	Architecture MVC	41
4.1	Décomposition du sprint 1 en items	44
4.2	Diagramme de cas d'utilisation du sprint 1	46
4.3	Diagramme de cas d'utilisation « S'authentifier »	46
4.4	Diagramme de cas d'utilisation « Gérer les activités »	48
4.5	Diagramme de cas d'utilisation « Générer le bilan d'un membre d'équipe »	49
4.6	Diagramme de séquence « S'authentifier »	51
4.7	Diagramme de séquence « Ajouter une activité »	52
4.8	Diagramme de séquence « Générer le bilan d'un membre d'équipe »	53
4.9	Diagramme de classe du sprint 1	55
4.10	Interface d'authentification	57
4.11	interface d'ajout d'une activité	58
4.12	interface du Bilan d'un membre d'équipe	59
4.13	Décomposition du sprint 2 en items	60
4.14	Diagramme de cas d'utilisation du sprint 2	62
4.15	Diagramme de cas « Gérer les membres d'équipe »	63
4.16	Diagramme de cas « Générer le bilan de l'équipe »	64
4.17	Diagramme de séquence « Ajouter un membre d'équipe »	65

TABLE DES FIGURES

4.18	Diagramme de séquence « Générer le bilan de l'équipe »	66
4.19	Diagramme de classe du sprint 2	67
4.20	Interface d'ajout d'un utilisateur	68
4.21	Interface du bilan d'équipe	69
5.1	Décomposition du sprint 1 en items	71
5.2	Diagramme de cas d'utilisation du sprint 1	73
5.3	Diagramme de cas d'utilisation « Gérer les équipes »	73
5.4	Diagramme de cas d'utilisation « Générer le bilan de l'unité »	75
5.5	Diagramme de séquence « Ajouter une équipe »	76
5.6	Diagramme de séquence « Générer le bilan de l'unité »	77
5.7	Diagramme de classes du sprint 1	78
5.8	Interface d'ajout d'une équipe	79
5.9	Interfaces de génération du bilan de l'unité	80
5.10	Décomposition du sprint 2 en items	81
5.11	Diagramme de cas d'utilisation du sprint 2	83
5.12	Diagramme de cas « Gérer les sorties pédagogiques »	84
5.13	Diagramme de cas « Gérer les activités visiteur »	85
5.14	Diagramme de séquence « Supprimer une sortie pédagogique »	86
5.15	Diagramme de séquence « Consulter le profil d'un membre d'équipe »	87
5.16	Diagramme de classe du sprint 2	88
5.17	Interface de présentation	89
5.18	Interface d'accueil	90
5.19	Interface d'ajout d'une sortie	91
A.1	Diagramme de séquence système " Ajouter un membre d'équipe "	ii
A.2	Diagramme de séquence système " Modifier un membre d'équipe "	iii
A.3	Diagramme de séquence système " Supprimer un membre d'équipe "	iv
A.4	Diagramme de séquence système " Rechercher un membre d'équipe "	v
A.5	Diagramme de séquence système " Générer le bilan d'un membre d'équipe "	vi
A.6	Diagramme de séquence système " Ajouter une activité "	vii
A.7	Diagramme de séquence système " Modifier une activité "	viii
A.8	Diagramme de séquence système " Supprimer une activité "	ix
A.9	Diagramme de séquence système " Rechercher une activité "	ix
A.10	Diagramme de séquence système " Générer le bilan de l'unité "	x
A.11	Diagramme de séquence système " S'authentifier "	xi
B.1	Interface "Profil d'un Membre d'équipe "	xii
B.2	Interface "Profil d'un Chef d'équipe "	xiii
B.3	textnterface "Profil d'un Directeur "	xiv

Liste des tableaux

1.1	Les équipes de l'unité de recherche LaMOS [1].	6
2.1	Historique de versions du système Android [18].	22
3.1	Description des acteurs	31
3.2	Présentation de l'équipe Scrum	33
3.3	Backlog du produit	36
3.4	Environnement matériel	38
4.1	Le sprint planning de sprint 1	44
4.2	Backlog du sprint 1	45
4.3	Description textuelle du cas d'utilisation « S'authentifier »	47
4.4	Description textuelle du cas d'utilisation « Ajouter une activité »	48
4.5	Description textuelle du cas d'utilisation « Générer le bilan d'un membre d'équipe »	50
4.6	Le dictionnaire de données	54
4.7	Le sprint planning du sprint 2	60
4.8	Backlog de sprint 2	61
4.9	Description textuelle du cas d'utilisation « Ajouter un membre d'équipe »	63
4.10	Description textuelle du cas d'utilisation « Générer le bilan de l'équipe »	64
5.1	Le sprint planning de sprint 1	72
5.2	Backlog du sprint 1	72
5.3	Description textuelle du cas d'utilisation « Ajouter une équipe »	74
5.4	Description textuelle du cas d'utilisation « Générer le bilan de l'unité »	75
5.5	Le sprint planning du sprint 2	81
5.6	Backlog de sprint 2	82
5.7	Description textuelle du cas d'utilisation « Supprimer une sortie pédagogique »	84
5.8	Description textuelle du cas d'utilisation « Consulter le profil d'un membre d'équipe »	85

Glossaire

-A-

APK : Android Package

API : Application Programming Interface

-C-

CSQ : Contrôle Statistiques de la Qualité

CSS : Cascading Style Skee

-D-

DBA : Data Base Administrator

Dalvik : est la machine virtuelle pour le langage Java des applications du système d'exploitation Android de Google. Le nom provient d'une ville d'Islande, d'où le créateur est originaire. Dalvik est produit par l'Open Handset Alliance

-E-

EPSIRT : Évaluation des Performances

-F-

FSE2 : Fiabilité des Systèmes Électro- Énergétiques

-H-

HTML : Hyper Text Markup Language

-I-

IDE : (Integrated Development Environment) est une interface qui permet de développer, compiler et exécuter un programme dans un langage donné.

IOS : (Internetworking Operating System) est un système d'exploitation et plate-forme logicielle pour smartphones et tablettes créé et utilisé exclusivement par Apple dont la version 1.0 est sortie le 29 Juin 2007 avec le lancement de l'iPhone.

-J-

JSP : Java Server Pages est une technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web.

JSON : JavaScript Object Notation – Notation Objet issue de JavaScript

-L-

LaMOS : Laboratoire de Modélisation et Optimisation des Systèmes

-M-

MCO : Méthodes Cybernétiques et Optimisation

MFS : Maintenance et Fiabilité des Systèmes

MPMS : Méthode d'Analyse de Perturbation des Modèles Stochastiques

MySQL : est un serveur de bases de données relationnelles Open Source.

MVC : (Modele Vue Controler) est un patron de conception, son but est de séparer l'interface, l'action et l'accès aux données.

-O-

OMG : Object Management Group

OMT : object modeling technique

OOSE : Object Oriented Software Engineering

OS : Operating system

-P-

PBI : Product Backlog Item

PDA : Personal Digital Assistant

PMA : PhpMyAdmin

PHP : HyperText Preprocessor

-S-

SR2 : Systèmes avec Rappels et Réseaux

Scrum : Systematic Customer Resolution Unraveling Meeting

SM : Scrum Master

SGBD : (Système de Gestion de Base de Données) est un logiciel système conçu pour créer et gérer des bases de données.

SQL : (Structured Query Language) est un langage permettant de communiquer avec une base de données.

-U-

UML : Unified Modeling Language

URL : Uniform Resource Locator

-X-

XML : eXtensible Markup Language

Introduction Générale

Au cours des dernières années, la révolution de l'informatique a balayé tous les pays du monde cela inclut notre pays l'Algérie qui a connu un fort développement incontestable dans ce domaine.

Parmi les outils qui ont impulsé le marché informatique, nous citons les applications mobiles, qui se définissent comme des programmes gratuits ou payants et exécutables à partir du système d'exploitation. Conjugés à l'essor d'internet, ils permettent généralement un accès plus confortable et plus efficace à des sites ou à des services.

Cela ne vous a certainement pas échappé : de plus en plus de personnes possèdent un Smartphone ou encore une tablette. Le développement mobile est un phénomène qui a pris beaucoup d'ampleur en quelques années. Pour les entreprises et les administrations, suivre cette tendance est important puisqu'elle leur permet de créer des services innovants ou encore d'améliorer leur communication.

Dans ce contexte, l'unité de recherche « LaMOS » de Béjaïa qui est l'une des plus anciennes structures de recherche en activité d'Algérie dans le domaine des mathématiques d'informatique et de la technologie ou nous avons effectué notre stage de fin de cycle, voudrait mettre en place un logiciel de gestion des bilans.

Pour atteindre ce but, nous avons pensé à développer une application mobile pour faciliter aux membres de l'unité LaMOS de rédiger les bilans scientifiques nécessaire. Pour cela, nous avons organisé notre travail en cinq chapitres, comme ceci :

Dans le premier chapitre, nous allons présenter l'organisme d'accueil de l'unité de recherche LaMOS et ses activités. Puis nous définissons la méthodologie à suivre afin de réaliser notre système.

Ensuite, dans le second chapitre, nous allons introduire l'univers Android, en décrivant c'est quoi l'informatique mobile, le système Android et les applications Android.

INTRODUCTION GÉNÉRALE

Le troisième chapitre explique notre démarche, soit, l'identification des futurs acteurs de notre système et l'analyse des besoins. Et aussi la description de l'architecture choisie pour réaliser notre solution.

Puis dans le chapitre quatre, nous avons illustré la première version (release) de mon projet en se basant sur la méthodologie Scrum. nous présentons tout au long de ce chapitre la spécification, la conception et la réalisation des deux premiers sprints.

Enfin, le cinquième chapitre, est dédié à la réalisation du deuxième release. Et nous concluons notre travail par une conclusion générale et quelques perspectives.

Chapitre 1

Présentation de l'organisme d'accueil

1.1 Introduction

Le présent chapitre, est consacré à la présentation du cadre général du projet. Tout d'abord, nous présentons l'organisme d'accueil. Ensuite, nous citons les problèmes à résoudre et les solutions proposées. Enfin, nous décrivons la méthode de travail adoptée pour développer et mettre en oeuvre notre solution.

1.2 Présentation de LaMOS (Laboratoire de Modélisation et Optimisation des Systèmes)

1.2.1 Définition

Le LaMOS (Modélisation et Optimisation des Systèmes) est l'une des plus anciennes structures de recherche en activité d'Algérie. Comprenant aujourd'hui 155 (cent cinquante-cinq) chercheurs de haut niveau, il vient d'être élevé au statut d'unité de recherche pluridisciplinaire (arrête n°002 du 04 Janvier 2014). Le LaMOS est ainsi devenu en juin 2013 la première (et la seule) unité de recherche du pays dans les domaines des mathématiques, de l'informatique et de la technologie[1].

1.2.2 Historique

Fondé en 1985, le LaMOS Bejaia (alors Laboratoire de Modélisation Stochastique) avait été initialement rattaché au projet d'unité de recherche en statistiques appliquées de l'université de Constantine (déposé en février 1985, révisé en décembre 1985 et en mai 1986). Il s'est ouvert aux applications des mathématiques :

- À la Science de l'ingénieur, en 1988 (au début de la crise économique en Algérie) ;
- Aux sciences humaines et sociales (Bibliographie et Méthodes d'Analyse, Histoire des Mathématiques) en 1991 (au début de la crise politique et identitaire en Algérie).

Un code spécifique lui avait été attribué en 1989 par les revues scientifiques internationales. La commission d'évaluation du ministère aux universités avait en 1991 souligné l'importance du projet LaMOS (Laboratoire de Modélisation et d'Optimisation des Systèmes) en tant qu'unité de recherche pluridisciplinaire. Suite à la promulgation de la loi sur la recherche, il avait été officiellement agréé par (arrêté ministériel n° 88 du 25 juin 2000) [1].

1.2.3 Organigramme de l'organisme d'accueil

La figure 1 représente l'organigramme de l'unité de recherche LaMOS de l'université de Bejaia.

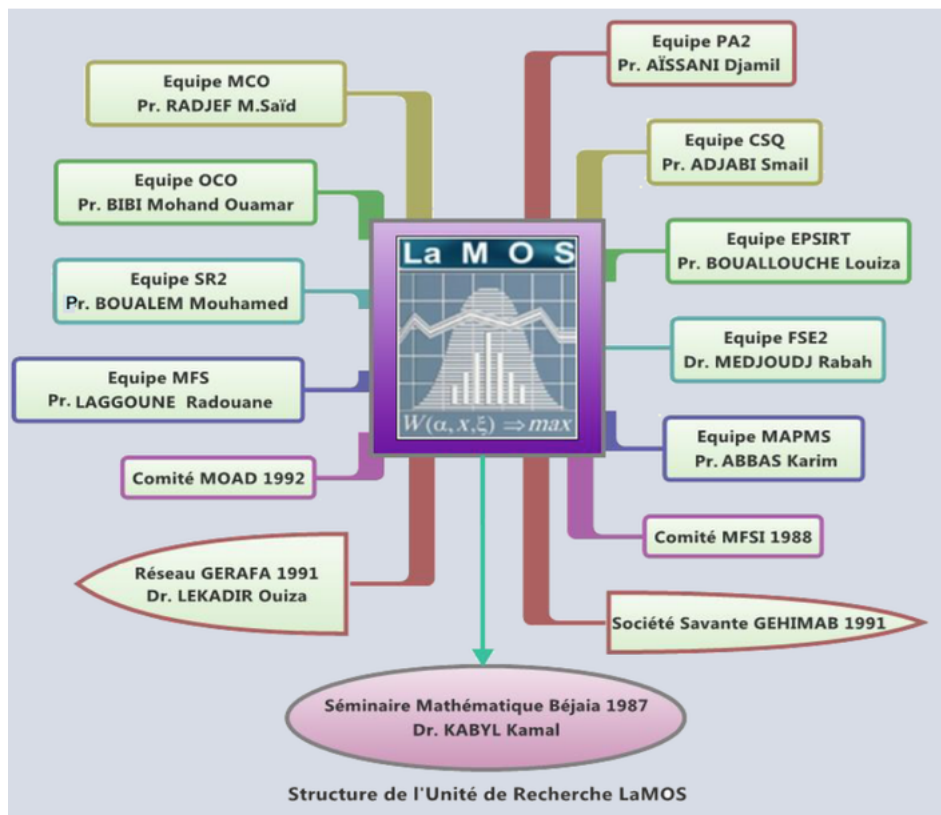


FIGURE 1.1 – Organigramme de l'unité de recherche LaMOS [1].

1.2.4 Structure de l'unité de recherche LaMOS

L'unité de recherche LaMOS a pour compétence le développement et l'application des méthodes de calcul scientifique et technique aux problèmes de modélisation, simulation et optimisation des systèmes complexes (notamment industriels et socio-économiques). Il est principalement constitué de 09 équipes et 02 groupes spécialisés :

CHAPITRE 1. PRÉSENTATION DE L'ORGANISME D'ACCUEIL

Intitule de l'équipe de recherche	Responsable de l'équipe	Axe de recherche
MCO (Méthodes Cybernétiques et Optimisation)	Pr RADJEF M. Said	<ul style="list-style-type: none">• Méthodes multicritères• Théorie des Jeux Différentiels
SR2 (Systèmes avec Rappels et Réseaux)	Pr BOUALEM Mouhamed	<ul style="list-style-type: none">• Systèmes avec rappels• Réseaux de files d'attente• Systèmes non fiables et avec vacances
CSQ (Contrôle Statistiques de la Qualité)	Pr ADJABI Smail	<ul style="list-style-type: none">• Tests et lois non paramétriques de fiabilité• Méthodes statistiques appliquées• Évaluation des performances des réseaux de télécommunication
PA2 (Processus Aléatoires et Applications)	Pr AISSANI Djamil	<ul style="list-style-type: none">• Chaînes de Markov• Gestion des stocks• Modèles de Risque
FSE2 (Fiabilité des Systèmes Électro- Énergétiques)	Pr MEDJOU DJ Rabah	<ul style="list-style-type: none">• Fiabilité Électrique et Électronique
EPSIRT (Évaluation des Performances)	Pr BOUALLOUCHE Louiza	<ul style="list-style-type: none">• Réseaux ad Hoc, Internet, Réseaux de capteurs• Évaluation de performance et qualité de services
OCO (Optimisation et Contrôle Optimal)	Pr BIBI Mohand Ouamar	<ul style="list-style-type: none">• Programmation linéaire en quadratique• Optimisation multicritère• Théorie qualitative et constructive du contrôle optimal

MFS (Maintenance et Fiabilité des Systèmes)	Pr LAGGOUNE Radouane	• Fiabilité et Optimisation des systèmes mécaniques
MPMS (Méthode d'Analyse de Perturbation des Modèles Stochastiques)	Pr ABBAS Karim	• Méthode d'Analyse de Perturbation des Modèles Stochastiques

TABLE 1.1 – Les équipes de l'unité de recherche LaMOS [1].

1.3 Problématique

LaMOS est une unité de recherche composée de chercheurs qui réalisent différentes activités (publication nationale, publication internationale, conférence . . .) afin de les publier et les mettre à la portée des lecteurs.

A l'issue d'une réunion organisée avec le responsable du service informatique , il en est sorti que les services principalement concernés par notre solution sont : les services de gestion des bilans.

Les outils actuels de prise de décision présents au sein de ces services laissent apparaître des insuffisances, dues à la fois aux technologies utilisées, aux conditions de leurs mises en œuvre, et aux méthodes de traitement de l'information, ce qui entraîne quelques problèmes :

- Des prises de décisions biaisées et/ou tardives influant négativement sur le pilotage de l'activité ;
- Redondance des mêmes activités dans un même bilan ;
- Pas de moyens de garder trace du travail fait par chaque chercheur ;
- Toutes les activités ne sont pas systématiquement suivies ;
- Manque de communication ;
- Recours à WORD et EXCEL qui n'est pas un moyen efficace.

1.4 Solutions proposées

Pendant la période de stage pratique passé au niveau de l'unité de recherche LaMOS, nous avons pu comprendre l'organisme et le fonctionnement de cette unité, ce qui nous a permis après une étude et analyse préliminaire d'opter pour la proposition suivante qui pourrait bien répondre aux questions posées précédemment.

- Conception et réalisation d'un système de gestion des bilans sous Android.

1.5 Méthodologie de conception

Pour la performance de notre système de gestion d'informations, et pour satisfaire les exigences du client en moindre coût et délai, nous avons utilisé une méthode agile pour le déroulement du projet.

1.5.1 Méthologie agile

Une fois qu'une organisation décide d'adopter une gestion de développement agile, il reste encore à choisir la méthodologie la plus adaptée à son projet. En effet, les méthodes agiles disponibles sont nombreuses et peuvent être source de confusion. Les méthodes agiles les plus populaires en usage aujourd'hui sont :

- L'eXtrême Programming (XP)
- Scrum
- Feature Driven Development (FDD)
- Lean Software Development
- Agile Unified Process (Agile UP ou AUP)
- Crystal
- Dynamic Systems Development Method (DSDM) [2].

1.5.2 Pourquoi Scrum

Dans le cadre de notre projet et afin d'assurer le bon déroulement des différentes phases de ce dernier, nous avons opté pour la méthode agile Scrum pour la conception et le développement de notre application pour des raisons bien déterminées en effet le processus Scrum s'adapte parfaitement à la décomposition de notre projet de fin de cycle, il se base sur les avantages suivants :

- Plus de souplesse et de réactivité;
- Grande capacité d'adaptation au changement grâce à des itérations courtes;
- Satisfaire au mieux les besoins du client.

1.5.3 Définition de Scrum

Scrum est une méthode de développement agile orientée projet informatique dont les ressources sont régulièrement actualisées.

La méthode Scrum tire son nom du monde du rugby, scrum = mêlée. Le principe de base étant d'être toujours prêt à réorienter le projet au fil de son avancement.

CHAPITRE 1. PRÉSENTATION DE L'ORGANISME D'ACCUEIL

C'est une approche dynamique et participative de la conduite du projet. La mêlée est une phase de jeu essentielle au rugby. Elle permet au jeu de répartir sur d'autres bases. La réunion dans la méthode Scrum relaie la métaphore [3].

Le scrum se base sur trois piliers essentiels qui le caractérisent d'une manière à vrai dire unique comparer aux autres méthodes agiles :

- **La transparence** : les rôles au sein d'une équipe Scrum ne sont pas déterminés entre les membres, chacun apporte son savoir-faire. Cela facilite le partage des informations au sein du projet.
- **L'inspection** : Scrum prévoit d'organiser régulièrement des inspections sur les artefacts produits afin de détecter des comportements imprévus au sein du projet.
- **L'adaptation** : Scrum fonctionne par des cycles de vie. À chaque début de cycle, sont réalisés une revue du cycle précédent et une préparation du cycle suivant [4].

1.5.4 Concepts de base de la méthode de gestion de projet Scrum

La méthode Scrum est caractérisée par le contrôle de la performance et la mise en œuvre d'un plan d'amélioration pour la gestion des projets. L'équipe Scrum (le groupe travaillant sous la méthode scrum) se penche vers le respect des rituels élémentaires de la méthode pour pouvoir améliorer la rentabilité et la productivité en minimisant les pertes de temps et en maximisant surtout l'affranchissement des obstacles.

Scrum est considéré comme un cadre ou « Framework » de gestion de projet. Ce cadre est constitué d'une définition des rôles, de réunions et d'artefacts. L'un des aspects qui distinguent Scrum des autres méthodes agiles réside dans le fait qu'il fait appel à des acteurs qui occupent, chacun, un rôle important dans le système.

1. Acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositifs matériels ou autres système) qui interagissent directement avec le système étudié.

Product Owner

Il porte la vision du produit à réaliser et il s'agit donc généralement d'un expert métier. Il travaille en collaboration directe avec l'équipe de développement et a notamment la charge de remplir le backlog produit et de déterminer la priorité des user stories à réaliser. Il peut être interne ou externe, même s'il s'agit généralement du client [2].

Scrum Master

Son travail s'apparente à celui d'un manager ou d'un coach qui s'assure que les éléments des cahiers de charges et les principes de la méthode agile soient correctement respectés. Le SM est, en quelque sorte, le lien entre l'équipe de développement et le product owner [5].

Équipe de développement

Généralement composée de 1 à 6 personnes, elle est chargée de transformer les besoins exprimés par le product owner sous la forme des user stories en fonctionnalités réelles, opérationnelles et utilisables. L'équipe est généralement composée de plusieurs profils, ne se limitant pas à des développeurs, et peut intégrer des architectes, un DBA, un graphiste, un ergonomiste ou encore un ingénieur-système ou réseau, en fonction des besoins [2].

2. Évènements

Le product owner rencontre le client et analyse son besoin. Il identifie toutes les fonctionnalités dont le produit devra être composé (en d'autres mots les user stories) dans ce qui s'appelle le product Backlog, qui lui confie l'autorité pour définir les fonctionnalités du produit final. Ce « cahier des charges » n'est pas fixé pour toujours, et pourra évoluer en fonction des besoins du client.

L'équipe décide de ce qu'elle peut faire et dans quel ordre le faire en répartissant le cahier des charges en sprint divisé en plusieurs tâches représentées dans des user stories distribuées dans un Scrum board, un des piliers de la transparence. Des réunions quotidiennes sont faites "Daily Scrum ou Daily Stand-Ups", où chacun parle aux autres de son état d'avancement.

Sprint planning

L'équipe se répartit les tâches en fonction des priorités (Business Value/Ressource poids), elles sont définies dans le sprint backlog. Dédié à la planification du sprint. Les user stories sont découpées en tâches unitaires et estimées en termes d'effort par l'équipe [4].

Sprint review (revue de sprint)

L'équipe présente ce qu'elle a accompli durant le sprint. Le sprint review prend la forme d'une démo des nouvelles fonctionnalités ou de la compréhension de l'architecture. Elle prend place à la fin du sprint. Elle permet à l'équipe d'adapter le Product backlog. Durant cette phase on va définir les tâches terminées (objets utilisables et potentiellement livrable) entre le Product owner et l'équipe. Elle va permettre la démonstration du produit réalisé à l'issue du sprint et inspection des users stories contenues dans ce dernier. Elle peut être ouverte à d'autres personnes externes (client, user etc.). Enfin sa durée est de quatre heures pour un mois de sprint [4].

Sprint rétrospective (rétrospective de sprint)

Permet de définir périodiquement ce qui marche et ce qui ne marche pas, elle a une durée de quinze à trente minutes. L'équipe complète discute des tâches à démarrer, s'il faut les arrêter ou continuer. Il s'agit d'un échange sur la manière de travailler ensemble et permet d'apporter des améliorations dans ce sens. Le sprint rétrospectif est utile pour apprécier la vélocité de l'équipe via le nombre de points d'effort que l'équipe peut absorber et qui augmente avec le temps. Ainsi ce dernier point pourra apporter un gain de vélocité/performance dans la maîtrise du projet et de la technologie abordée [4].

Daily scrum meeting

A lieu quotidiennement, pour une durée de quinze minutes, debout, tout le monde est invité. Il n'est pas fait pour résoudre les problèmes en détail. En général on posera trois Questions : Qu'est-ce qu'on a fait hier ? Qu'est-ce qu'on va faire aujourd'hui ? Rencontrez-vous des obstacles ?[4]

3. Artefacts

En archéologie, le terme « artefact » fait référence à un objet fabriqué par un humain. Les racines latines du mot artefact se traduisent approximativement par « œuvre d'art ». Ainsi, nous fabriquons un artefact, qu'il s'agisse d'un outil permettant de résoudre un problème ou une œuvre d'art qui nous inspire. Scrum décrit trois artefacts principaux : le backlog de produit, le backlog de sprint et l'incrément de produit [6].

Product backlog

Il s'agit d'une liste de tout ce qui est désiré sur le projet on parle alors d'users stories, sa gestion est réalisée par le product owner qui peut modifier la priorité (notamment à la fin de chaque sprint), chaque user story doit être classée en fonction de sa business value qui change dynamiquement en fonction du besoin et des attentes du client. Affinement des users stories afin de créer des Product backlog item (PBI) ou tâches qui seront positionnées dans les sprints à venir [4].

Sprint backlog

Il s'agit des objectifs du sprint basés sur la plus haute valeur de tâches (ou PBI), définit le travail à effectuer par l'équipe de développement, plan en temps réel du travail dans le sprint. Il doit être défini avec le product owner. (Choix des itérations a commencé en fonction des priorités) Les équipiers sont libres de choisir les users stories en fonction de leur capacité, mais doivent s'engager à les réaliser et à les livrer à la fin du sprint. La capacité de poids sera affectée par équipiers. Aussi elle sera redéfinie si nécessaire à la fin de chaque sprint [4].

4. Users stories

Les spécifications sont souvent une cause majeure de l'échec d'un projet. De mauvaises spécifications peuvent entraîner un manque de vision sur le produit attendu, des fonctionnalités redondantes/contradictoires.

Le but de l'utilisation des « user stories » est de permettre de répondre plus rapidement et avec moins de coût au changement rapide des exigences du monde réel [7].

1.5.5 Présentation de la méthode Scrum

Le principe de la méthode agile Scrum est de concentrer l'équipe de développement sur un ensemble de fonctionnalités à réaliser de façon itérative, dans des itérations d'une durée de deux à quatre semaines, appelées des sprints. Chaque sprint doit aboutir à la livraison d'un produit partiel [8].

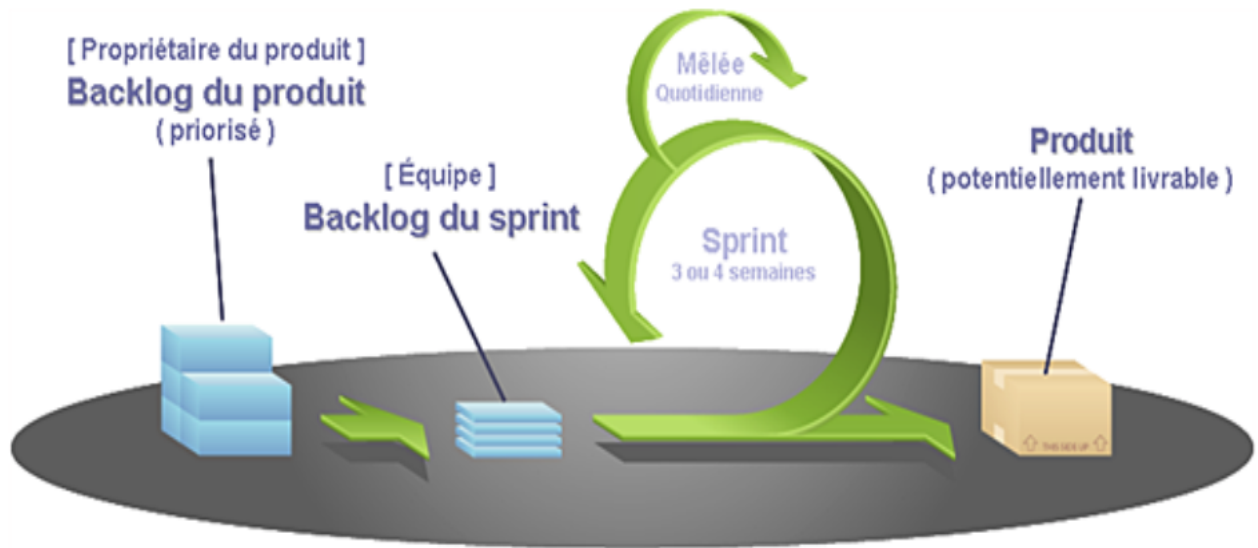


FIGURE 1.2 – Cycle de vie de Scrum [39].

Comme indiqué dans la figure ci-dessus, pour mettre en place la méthode Scrum, Il faut dégager dans un premier lieu le maximum de fonctionnalités à réaliser pour former le backlog du produit. En second lieu, définir les priorités des fonctionnalités et choisir lesquelles seront réalisés dans chaque itération.

Par la suite, focaliser l'équipe de façon itérative sur l'ensemble des fonctionnalités à réaliser dans des itérations appelées sprints. Un sprint aboutit toujours à la livraison d'un produit partiel fonctionnel appelé incrément. Ainsi, vers la fin de chaque sprint, une réunion aura lieu pour effectuer la revue de l'itération.

L'objectif de cette réunion consiste à valider l'incrément qui était produit pendant l'itération.

1.6 Langages de modélisation

Pour Faciliter notre tâche nous avons fait recours au langage de modélisation unifié UML

1.6.1 UML (Unified Modeling Language)

L'UML était développée en réponse à l'appel aux propositions lancées par l'OMG dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets. Elle est héritée de plusieurs autres méthodes telles qu'OMT et OOSE et Booch. Les principaux auteurs de la notation UML sont Grady Booch, Ivar Jacobson et Jim Rumbaugh [9].

1.6.2 Différents types de diagrammes UML

UML s'articule maintenant autour de 13 diagrammes différents, dont quatre nouveaux diagrammes introduits par UML 2.0. Chacun d'eux est dédié à la représentation d'un système logiciel suivant un point de vue particulier. Par ailleurs, UML modélise le système suivant deux modes de représentation : l'un concerne la structure du système pris " au repos", l'autre concerne sa dynamique de fonctionnement. Les deux représentations sont nécessaires et complémentaires pour schématiser la façon dont est composé le système et comment ses composants fonctionnent entre eux [10].

La figure suivant présente les différents types de diagrammes UML.

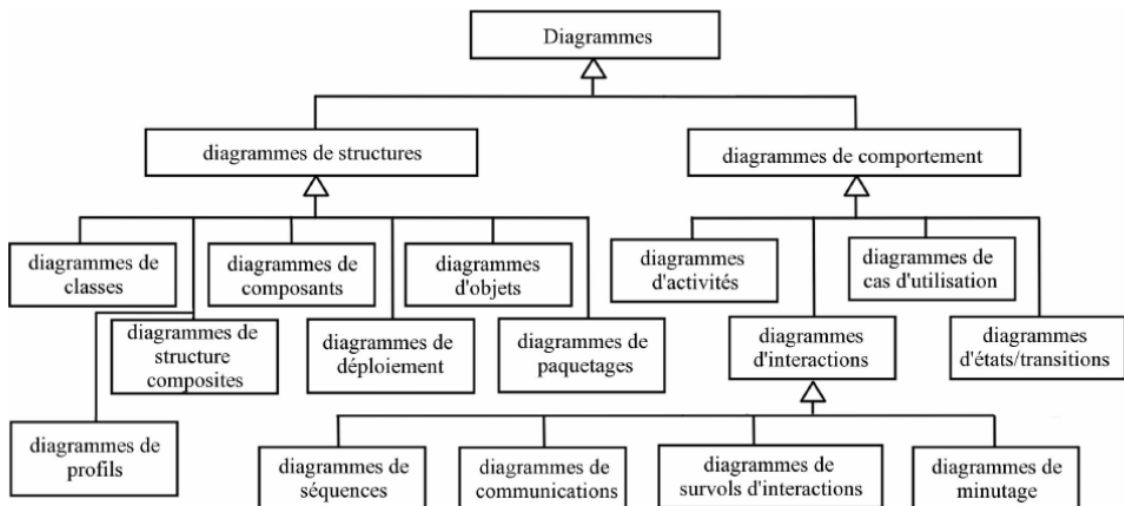


FIGURE 1.3 – Types de diagrammes UML [40]

Dans notre projet nous allons utiliser seulement les diagrammes suivants :

- **Diagramme de cas d'utilisation** : représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est utilisé dans les deux étapes de capture des besoins fonctionnels et qualités.
- **Diagramme de séquence** : est un diagramme d'interaction, il représente les échanges de messages entre objets, dans le cadre d'un fonctionnement particulier du système. Il sert ensuite à développer en analyse les scénarios d'utilisation du système.
- **Diagramme de classe** : a toujours été le plus important dans toutes les méthodes agiles. C'est également celui qui contient la plus grande gamme de notations et de variantes centralise l'organisation des classes de conception, c'est lui qui se transforme le plus aisément en code [10].

1.7 Conclusion

Dans ce chapitre, nous avons commencé par un aperçu du projet en décrivant l'organisme d'accueil, LaMOS. Nous avons présenté quelques notions sur la modélisation avec le langage UML et la méthode Scrum.

Dans le chapitre suivant nous allons présenter l'univers d'Android.

Chapitre 2

Généralités sur les applications mobiles

2.1 Introduction

Rares sont les systèmes ayant connu une progression telle que celle que connaît actuellement le système d'exploitation de téléphone.

Dans ce deuxième chapitre nous allons définir c'est quoi l'informatique mobile, et quelques notions de base, puis nous allons présenter l'univers Android tout en définissant c'est quoi le système Android, son architecture et le cycle de vie d'une application Android.

2.2 Informatique mobile

2.2.1 Présentation

Aujourd'hui, avec l'explosion de l'internet mobile. Cette dernière prend de plus en plus de place sur le marché. Les smartphones et tablettes ont envahi nos vies. Ils offrent des applications variées qui nous permettent de nous divertir et nous simplifient la vie. Cet afflux technologique change radicalement notre rapport à l'informatique en affranchissant de l'utilisation d'un micro-ordinateur classique. Par ailleurs, sa capacité de plus en plus grande permet de stocker musique, photo, vidéo, contact...

2.2.2 Terminaux mobiles

Un terminal mobile est un petit appareil informatique permettant le traitement et l'échange de données ou de communication qu'on peut transporter avec soi dans ses déplacements et utiliser comme terminal donnant accès sans fil à un ou plusieurs réseaux [11].

Parmi les terminaux mobiles, on trouve des assistants numériques personnels (PDA), téléphones intelligents (Smartphone), des tablettes, etc.

Assistants numériques personnels (PDA)

Un assistant numérique personnel est une machine de poche assurant les fonctionnalités d'agenda, de carnet d'adresses, de bloc-notes et de gestion des tâches.

Dotés d'un système d'exploitation propre, les PDA peuvent accueillir les applications tierces les plus diverses (calcul d'itinéraires en métro, gestionnaire de listes de courses, jeux...). Les assistants personnels proposent des outils de bureautique allégés tels qu'un traitement de texte, un tableur, une calculatrice, des visualiseurs pour un grand nombre de formats de fichiers (fichiers PDF, images, etc.).

En plus de ces fonctions de base, de plus en plus de PDA propose des outils multimédias avancés permettant de lire des vidéos (dans les différents formats, y compris le format DivX), des musiques (notamment au format mp3) ou des animations Flash [12].



FIGURE 2.1 – Un assistant numérique personnel [41]

Smartphones

Un smartphone (téléphone intelligent) désigne un téléphone portable multifonctions qui a la capacité de naviguer sur internet, lire des musiques et des films, équipé d'une puce GPS, d'un écran tactile, qui peut évoluer avec le temps à l'aide de mises à jour, et qui a la capacité de télécharger et installer de nouvelles applications [12].

La plupart de ces appareils fonctionnent sur l'un de ces systèmes d'exploitation mobiles populaire : Android, Symbian, IOS, BlackBerry OS et Windows.



FIGURE 2.2 – Les smartphones [42]

Tablettes

Une tablette tactile est un appareil doté d'un écran qui permet d'interagir simplement en touchant l'écran ou à l'aide d'un stylet. Certaines tablettes permettent également de connecter un clavier et une souris.

Une tablette permet de se connecter à internet grâce au Wifi (ou au réseau 3G), mais peut également fonctionner sans connexion pour certaines applications.

Sur une tablette, on peut surfer sur internet, consulter ses mails, écouter de la musique, prendre des photos et en consulter, jouer, lire un livre, regarder des vidéos et 1000 autres choses grâce à des « applications », c'est-à-dire des petits programmes destinés aux appareils mobiles [13].



FIGURE 2.3 – Les tablettes [43]

2.3 Applications mobiles

2.3.1 Définition

Une application mobile est un logiciel applicatif développé pour être installé sur un appareil électronique mobile, comme un smartphone, une tablette ou un baladeur numérique.

Une application mobile peut-être soit installée directement sur l'appareil dès sa fabrication en usine soit téléchargée depuis un magasin d'application dit « application store » tel que Google Play, l'App Store ou encore le Windows phone store. Une partie des applications disponibles sont gratuites tandis que d'autres sont payantes.

2.3.2 Avantages d'une application mobile

Une application mobile revêt de nombreux avantages comparés à un site web mobile, en voici 5 principaux :

- Un confort d'usage et une expérience utilisateur inégalée.
- L'accès direct aux contenus de l'application mobile via l'icône présent sur le Dashboard du téléphone ou de la tablette (mode d'accès sans URL).
- Un fonctionnement en mode déconnecté.
- Elle permet d'utiliser et d'intégrer toutes les fonctionnalités téléphone (accéléromètre, gyroscope, GPS, caméra...), ce qui n'est pas forcément le cas des Web Apps.
- L'implémentation de fonctionnalités natives comme par exemple les notifications « PUSH » [14].

2.3.3 Inconvénients d'une application mobile

- Le principal inconvénient d'une application mobile est qu'elle doit respecter les règles définies par les différentes sociétés des plateformes mobiles. Que ce soit l'approbation nécessaire des Apps Store pour diffuser l'application ou ses mises à jour
- Les conditions tarifaires imposées ou la non-compatibilité avec les autres systèmes d'exploitation mobiles
- Le coût lié au développement d'une application mobile est un frein car généralement plus élevé si elle est portée sur plusieurs plateformes (afin d'être disponible pour un maximum de mobinautes) que le coût d'un site mobile ou d'une Web App. Il faudrait potentiellement prévoir un développement sur chaque technologie, et donc un coût supplémentaire si l'on souhaite se positionner sur tous les modèles
- Pour que l'utilisateur ait accès à la dernière version, il faut qu'il la mette à jour depuis le store contrairement aux sites mobiles et Web Apps qui se mettent à jour directement [14].

2.3.4 Différents types d'application mobile

Il existe plusieurs types d'applications mobiles qui sont les applications natives, web et hybrides. Chaque type d'application à ses caractéristiques propres :

1. Application native

Une application native est installée sur un smartphone ou une tablette, elle est sur-mesure et doit être adaptée à chaque système d'exploitation (OS), en somme une version différente pour chaque type de téléphone [15].

2. Application web

En opposition à une application native, une application web est une application mobile développée avec les outils de développement web actuel : HTML5, CSS3 et JavaScript. C'est un genre de site internet mobile (dont le design ressemble à une application), créé pour être consulté à partir du navigateur internet d'un smartphone.

La connexion web est indispensable. À ne pas confondre avec le responsive design qui est une façon de coder une page web afin qu'elle soit consultable sur tous types d'écrans [15].

3. Application hybride

Les applications hybrides sont des applications qui combinent les éléments d'une Web App et les éléments d'une application native. Elles reposent essentiellement sur la solution Cordova/PhoneGap, cette solution sert de passerelle entre le langage web et le natif. Cette solution nous permet d'utiliser un seul et même outil pour le développement et les langages issus du développement Web pour tous les mobiles (IOS, Android et Windows Phone) [16].

2.4 Système d'exploitation Android

2.4.1 Présentation

Android est un OS (Operating System ou Système d'exploitation en français) Open Source pensé pour les téléphones mobiles et développé par l'Open Handset Alliance (OHA) sous l'autorité de Google. L'OHA est composé d'environ 80 sociétés telles que Samsung, HTC, SFR, Orange, Asus, Qualcomm. . .

Android est basé sur un kernel Linux et possède une licence Apache License 2.0. Il sépare la couche matérielle de la couche logicielle, c'est-à-dire que n'importe quel téléphone Android peut faire fonctionner la même application et ainsi créer un écosystème riche pour les constructeurs, les utilisateurs et les développeurs.

Android est aussi un Framework et en tant que développeur, vous avez accès au SDK (Software Development Kit - Kit de développement) et à tout le code source de la plateforme. Ce qui vous permet de comprendre son fonctionnement, de créer des versions personnalisées de l'OS et bien sûr, de développer des applications.

Par conséquent, plusieurs appareils exotiques tournants sous Android ont vu le jour : autoradio, montre, casque de ski, etc...[17]

2.4.2 Historique

L'histoire d'Android commence en octobre 2003, lorsque Andy Rubin, Rich Miner, Nick Sears et Chris White créent la société Android à Palto Alto (Californie). Google a racheté la société en août 2005. Deux ans plus tard, l'Open Handset Alliance est annoncé et Android devient officiellement Open Source.

La première version du SDK Android 1.0 sort en 2008 avec le premier téléphone sous Android (HTC Dream). En avril 2009, la version 1.5 (API 3) d'Android sorti. Cette version baptisée Cupcake (Petit gâteau) inaugure les nouveaux noms des versions d'Android. Ce qui donnera pour les futures versions les noms montrés dans le tableau suivant :

Nom	Version	Date	Niveau API
N/A	1.0	Septembre 2008	1
	1.1	Février 2009	2
Cupcake	1.5	Avril 2009	3
Donut	1.6	Septembre 2009	4
Eclair	2.0-2.1	Octobre 2009	5-7
Froyo	2.2-2.3	Mai 2010	8
GingerBread	2.3-2.3.7	Décembre 2010	9-10
HoneyComb(a)	3.0-3.2.6	Février 2011	11-13
Ice Cream Sandwich	4.0-4.0.4	Octobre 2011	14-15
Jelly Bean	4.1-4.3.1	Juin 2012	16-18
KitKat	4.4.x	Octobre 2013	19-20
Lollipop	5.0-5.1.1	Octobre 2014	21-22
Marshmallow	6.0-6.0.1	Mai 2015	23
Nougat	7.0-7.1.1	Septembre 2016	24-25
Oreo	8.x.x	Octobre 2017	26-27
Pie	9.x	Aout 2018	28
Android 10	10	Septembre 2019	29

TABLE 2.1 – Historique de versions du système Android [18]

2.4.3 Architecture

Le diagramme suivant illustre les composants principaux du système d'exploitation Android. Chaque section sera décrite dans ce qui suit :

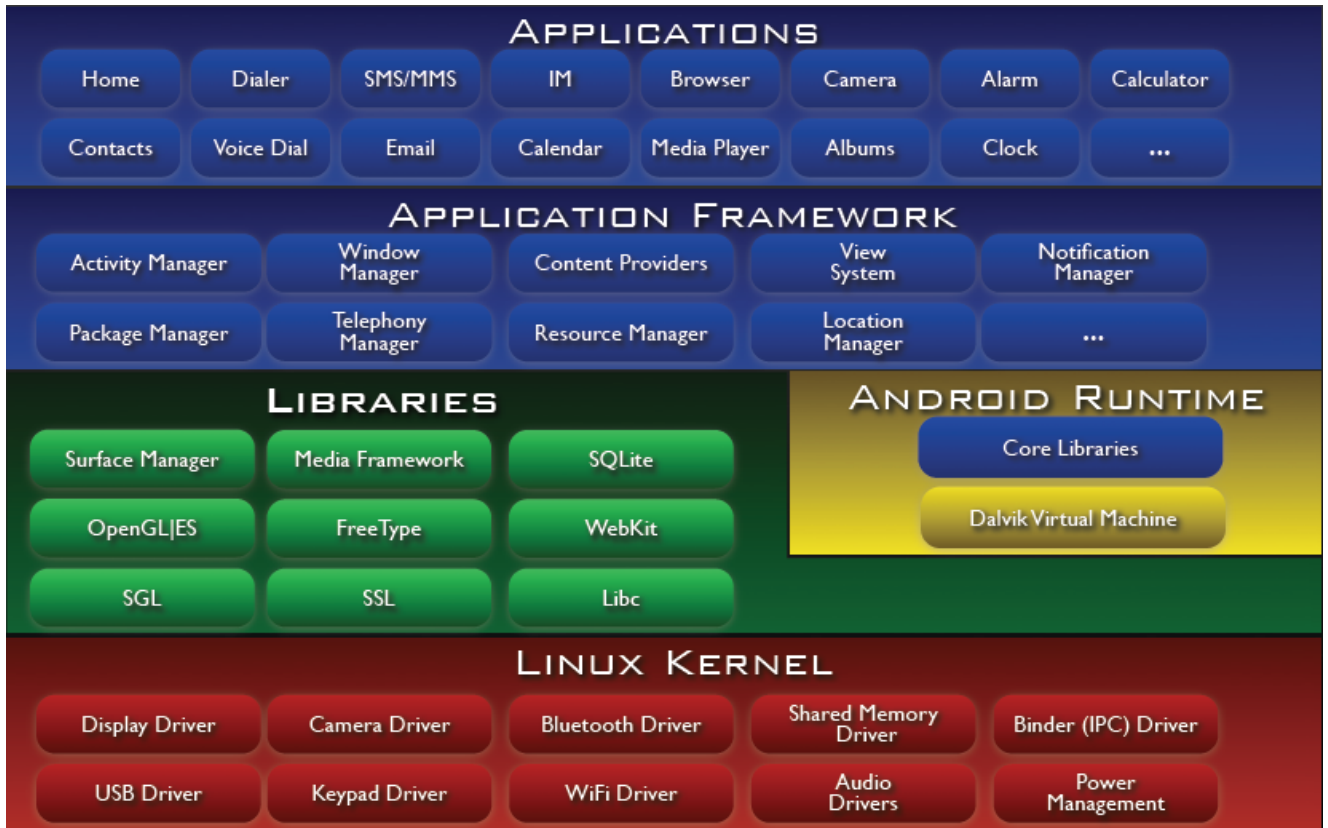


FIGURE 2.4 – Architecture de la plateforme Android [18]

Comme le montre la figure ci-dessus, le système d'exploitation Android est basé sur une architecture à quatre niveaux qui sont : linux kernel, Android runtime et librairies, Framework application et application.

— Linux Kernel

Android s'appuie sur un noyau (Kernel en anglais) dont la version est Linux 2.6. Pour être précis, le noyau est l'élément du système d'exploitation qui permet de faire le pont entre la partie matérielle et la partie logicielle. D'ailleurs, si vous regardez attentivement le schéma, vous remarquerez que cette couche est la seule qui gère le matériel. La version du noyau utilisée avec Android est une version conçue spécialement pour l'environnement mobile, avec une gestion avancée de la batterie et une gestion particulière de la mémoire [19].

— Les librairies

À ce niveau, figure la couche des bibliothèques principales du système fournies par des tiers. Celles-ci, de bas niveau, sont écrites en C et/ou C++. Elles fournissent des services essentiels tels que la bibliothèque d’affichage en 2D (SGL), la bibliothèque d’affichage graphique 3D (OpenGL), la bibliothèque de base de données (SQLite), la lecture et l’enregistrement audio et vidéo (Media Framework), un moteur de navigation Web (WebKit) [16].

— Android Runtime (Le moteur d’exécution linux)

Le runtime Android est basé sur le concept de machine virtuelle, utilisée en JAVA. Étant donné les limitations des dispositifs (peu de mémoire et la vitesse du processeur), il n’a pas été possible d’utiliser une machine virtuelle JAVA standard. Google a pris la décision de créer une nouvelle machine virtuelle Dalvik, afin de mieux répondre à ces limitations. Dans cette partie, qui est l’environnement d’exécution, on retrouve essentiellement deux parties :

Core Librairies : Les Cores librairies fournissent le langage JAVA, disponible pour les applications. Le langage JAVA fournit avec Android reprend en grande partie l’API JSE 1.5. Il y a des choses qui ont été mises de côté, car cela n’avait pas de sens pour Android (comme les imprimantes, swing, etc.) et d’autres APIs spécifiques requises pour Android ont été rajoutées.

Dalvik : Les applications JAVA, développées pour Android, doivent être compilées au format Dalvik exécutable (.dex) avec l’outil (dx). Cet outil compile les (.class) en (.dex) et ensuite il convertit ces (.class) en (.dex).[19]

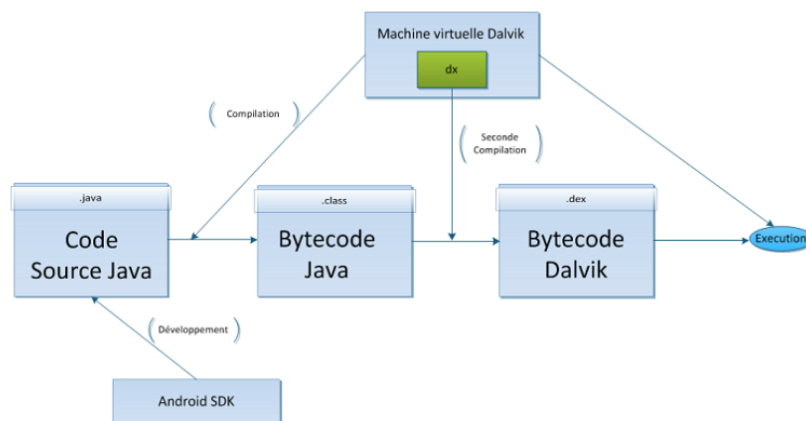


FIGURE 2.5 – La machine virtuelle Dalvik

— Framework Application

La couche Application Framework fournit de nombreux services de niveau supérieur aux applications sous forme de classes JAVA. Les développeurs d'applications sont autorisés à utiliser ces services dans leurs applications.

Le cadre Android comprend les services clés suivants :

- **Gestionnaire d'activités** : Contrôle tous les aspects du cycle de vie des applications et de la pile d'activités.
- **Fournisseurs de contenu** : Permet aux applications de publier et de partager des données avec d'autres applications.
- **Gestionnaire de ressources** : Donne accès à des ressources intégrées non codées telles que des chaînes, des paramètres de couleur et des dispositions d'interface utilisateur.
- **Gestionnaire de notifications** : Permet aux applications d'afficher des alertes et des notifications à l'utilisateur.
- **Système d'affichage** : Un ensemble extensible de vues utilisé pour créer des interfaces utilisateur d'application [20].

— Application

Les applications Android se trouvent dans la couche supérieure. Au niveau de la couche application, nous écrivons notre application à installer uniquement sur cette couche. Ces exemples d'applications sont les jeux, les messages, les contacts, etc [20].

2.5 Applications Android

2.5.1 Présentation

Une application Android est une application mobile spécifiquement développée pour les smartphones utilisant le système d'application Android acheté et développé par Google [21].

Sous Android, une application est composée d'une ou plusieurs activités. Une activité est la base d'un composant pour la création d'interfaces utilisateur. Afin de faciliter la cinématique de l'application, il est préconisé de n'avoir qu'une interface visuelle par activité [22].

2.5.2 État d'une activité (application)

Une application Android peut se trouver dans différents états qui sont : en cours d'exécution, en pause, stopper et tuer.

- **En cours d'exécution** : l'activité est lancée par l'utilisateur, elle s'exécute au premier plan
- **Pause** : l'activité est lancée par l'utilisateur, elle s'exécute et elle est visible, mais elle n'est plus au premier plan. Une notification ou une autre activité lui a volé le focus et une partie du premier plan
- **Stopper** : l'activité a été lancée par l'utilisateur, mais n'est plus au premier plan et elle est invisible. L'activité ne peut interagir avec l'utilisateur qu'avec une notification
- **Tuer** : l'activité n'est pas lancée [23].

2.5.3 Cycle de vie d'une activité Android

Par défaut, chaque application Android s'exécute dans un processus séparé.

Android gère les ressources disponibles dans un appareil et peut, si besoin, fermer des applications pour libérer des ressources (hors application en cours d'exécution). Le choix de l'application à fermer dépend fortement de l'état du processus dans lequel elle se trouve.

Si Android doit choisir entre deux applications se trouvant dans le même état, il choisira celle qui se trouve dans cet état depuis plus longtemps.

Le cycle de vie d'une activité est assez complexe et sa compréhension est indispensable dans le développement Android. Le schéma ci-dessous résume ce cycle de vie.

Lors du lancement d'une activité, la méthode **onCreate** est appelée. Dans cette méthode, vous devez initialiser votre vue et lier les données à une liste. Cette méthode prend en paramètre un Bundle (pile) contenant l'état précédent de l'activité

Cet appel est suivi par la méthode **onStart** afin de signifier le lancement effectif de l'application (elle est maintenant visible). Cette méthode peut aussi être appelée par la méthode **onRestart**.

Puis la méthode **onResume** est appelée afin d'exécuter tous les traitements nécessaires au fonctionnement de l'activité (thread, processus, traitement), initialiser des variables et les listeners. Ces traitements devront être arrêtés lors de l'appel à la méthode **onPause** et relancés si besoin lors d'un futur appel à la méthode **onRésumé**.

Après ces trois appels, l'activité est utilisable et peut recevoir les interactions utilisateurs. Si une autre activité passe au premier plan, l'activité en cours d'exécution passera en pause. Juste avant l'appel à la méthode **onPause**, la méthode **onSaveInstanceState** est appelée afin de vous permettre de sauvegarder les informations importantes portées par l'activité. Ces informations pourront être appliquées aux futurs lancements de l'activité lors de l'appel à la méthode **onRestoreInstanceState** ou de l'appel à **onCreate**.

La méthode **onPause** permet de stopper tous les traitements effectués (traitement non nécessaire si l'activité n'est pas visible) par l'activité (traitement, thread, processus). Si votre activité est de nouveau visible, cela correspondra à un appel à la méthode **onResume**. Le passage de l'activité à l'état "stopper" correspond à un appel à la méthode **onStop**. Dans cette méthode, il faut arrêter tous les traitements restants. Une fois stoppée, votre activité peut :

- **Soit être relancée** : cela s'effectue par un appel à la méthode **onRestart** suivi du cycle de vie normal de l'activité.
- **Soit être tuée** : cela s'effectue par un appel à la méthode **onDestroy**, dans laquelle vous devez arrêter tous les traitements restants, fermer toutes les connexions à la base de données, tous les threads, tous les fichiers ouverts, etc. Vous pouvez provoquer l'appel à la méthode **onDestroy** en utilisant la méthode **finish** [24].

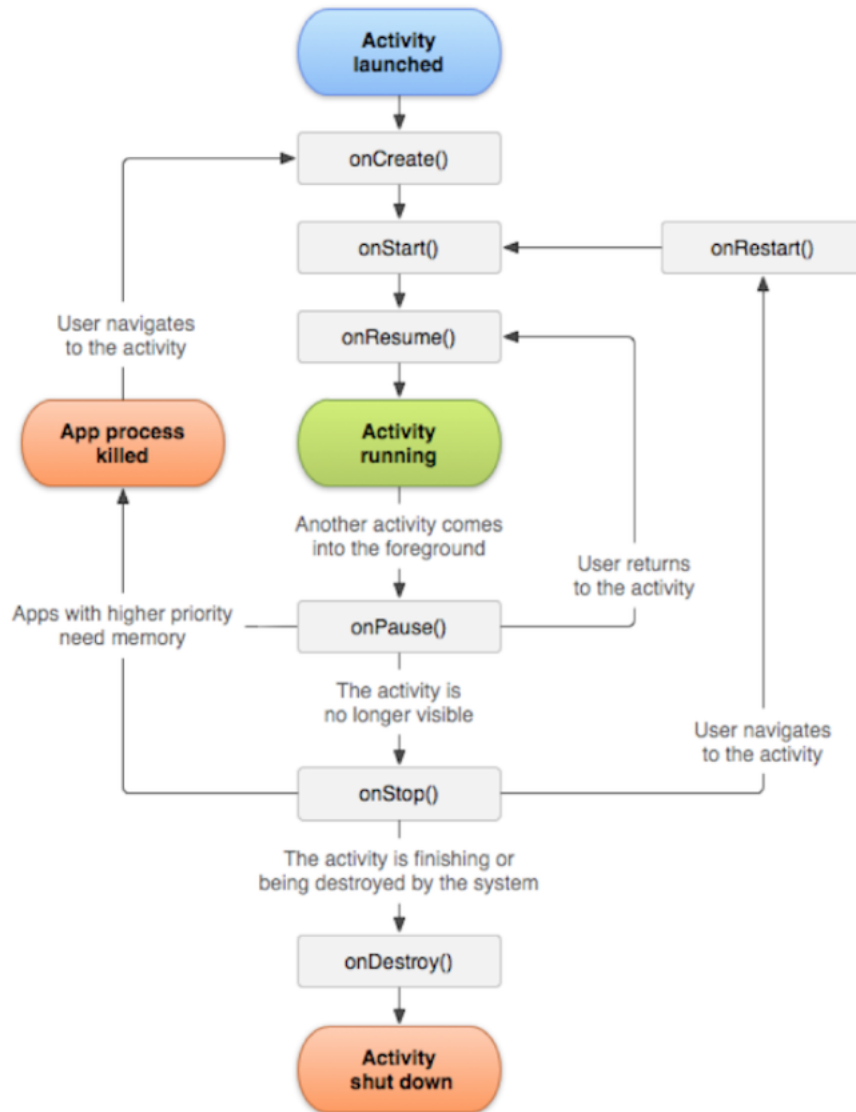


FIGURE 2.6 – Cycle de vie d’une activité Android [16]

2.6 Conclusion

Dans ce chapitre nous avons donné une brève description du système d’exploitation Android et quelques notions qui nous ont permis de mieux comprendre ce système.

Le chapitre suivant sera consacré à la phase de préparation de projet.

Chapitre 3

Préparation de projet

3.1 Introduction

Ce chapitre est consacré à la capture des besoins fonctionnels et non fonctionnels de notre système, au pilotage de projet avec scrum, par l'élaboration du backlog du produit avec une planification des sprints ainsi qu'une description de l'environnement de travail.

3.2 Spécification des besoins

La spécification des besoins consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités, les acteurs les plus pertinents et de préciser les risques les plus critiques.

3.2.1 Besoins fonctionnels

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée / sortie du système.

Notre application offre une interface mobile pour accéder aux espaces suivants :

Mode non connecté :

- Consulter les activités faites par les membres de l'unité LaMOS
- Fournir une présentation : les travaux effectués par l'unité, ces équipes et groupes.
- Consulter les informations d'un chercheur
- Assurer la sécurité du site par l'authentification (Email/Mot de passe)

Mode connecté :

- Assurer la gestion des activités par les membres d'équipe (Ajouter /Modifier /Supprimer /Rechercher)
- Assurer la gestion des membres par le chef d'équipe (Ajouter /Modifier /Supprimer /Rechercher)
- Générer les bilans d'un membre , d'une équipe et de toute l'unité par l'utilisateur
- Assurer la gestion des sorties pédagogiques par le directeur de l'unité (Ajouter /Modifier /Supprimer /Rechercher)
- Assurer la gestion des équipes par le directeur de l'unité (Ajouter /Modifier /Supprimer /Rechercher)

3.2.2 Besoins non fonctionnels

Une fois les besoins fonctionnels sont bien définis, les besoins non fonctionnels doivent être pris en compte tout au long du processus de développement de l'application à savoir :

- **Performance** : Un minimum de délai d'attente.
- **Simplicité d'utilisation** : un utilisateur avec un minimum de connaissances en application mobile doit être capable de manipuler les différentes tâches de l'application avec un minimum de clicks (touche écran) possibles.
- **Extensibilité** : l'application doit permettre à tout moment l'ajout d'autres modules.
- **Portabilité** : l'application doit être portable et fonctionnelle sur n'importe quel système d'exploitation.

3.3 Modélisation du besoin

3.3.1 Identification des acteurs

Un acteur représente une personne, un périphérique ou un autre système qui joue un rôle (interagit) avec le système [25].

Rôle	Fonctions
Visiteur	C'est un utilisateur qui n'est pas connecté et qui peut consulter les activités effectuées par les membres de l'équipe.
Membre de l'équipe	Est connecté individuellement il ajoute ses nouvelles activités
Chef d'équipe	Après sa connexion il peut faire ce que le membre de l'équipe fait de plus il peut gérer les nouveaux membres
Directeur	Après sa connexion il peut faire ce que le chef de l'équipe fait de plus il peut gérer les nouvelles équipes et organiser des sorties pédagogiques. Il dispose de toutes les autorisations sur l'application

TABLE 3.1 – Description des acteurs

3.3.2 Diagramme de cas d'utilisation global

Un diagramme de cas d'utilisation est une manière spécifique d'utiliser un système, il est utilisé pour exprimer les besoins du maître d'ouvrage [26].



FIGURE 3.1 – Diagramme de cas d'utilisation global

3.4 Pilotage du projet avec Scrum

Scrum est la méthodologie la plus utilisée parmi les méthodes agiles existantes.

3.4.1 Équipe et rôle

Pour notre projet les rôles sont repartis comme suit :

Rôles Scrum	Personnes Affectées
Product Owner	Unité de recherche LaMOS
Scrum Master	Mde K. ADEL
Team	Sonia KEBBI

TABLE 3.2 – Présentation de l'équipe Scrum

3.4.2 Backlog du produit

Le product backlog est l'ensemble des besoins recueillis pour créer le produit désiré. Si nous pensons tout de suite aux user-stories, cela peut concerner également les items techniques, les spike voir les bugs [27].

Chaque histoire utilisateur est caractérisée par une priorité définie par le product owner.

CHAPITRE 3. PRÉPARATION DE PROJET

Le backlog produit de notre futur système est le suivant :

Sprint	User Story	Entant que	Je veux...	Priorité
Authentification , Gestion des activités et la génération du bilan d'un membre d'équipe	Authentification	Membre de l'équipe	M'authentifier	1
	Gérer les activités	Membre de l'équipe	Ajouter une activité	2
	Gérer les activités	Membre de l'équipe	Afficher une activité	2
	Gérer les activités	Membre de l'équipe	Modifier une activité	2
	Gérer les activités	Membre de l'équipe	Supprimer une activité	2
	Gérer les activités	Membres de l'équipe	Rechercher une activité	2
	Gérer les bilans	Membre de l'équipe	Générer le bilan de ses activités	3

CHAPITRE 3. PRÉPARATION DE PROJET

Gestion des membres de l'équipe et la génération du bilan de l'équipe	Gérer les membres de l'équipe	Chef d'équipe	Ajouter un membre d'équipe	4
	Gérer les membres de l'équipe	Chef d'équipe	Supprimer un membre d'équipe	4
	Gérer les membres de l'équipe	Chef d'équipe	Modifier les information d'un membre d'équipe	4
	Gérer les membres de l'équipe	Chef d'équipe	Rechercher un membre d'équipe	4
	Gérer les bilans	Chef d'équipe	Générer le bilan de l'équipe	5
Gestion des équipes et la génération du bilan de l'unité	Gérer les équipes	Directeur	Ajouter une équipe	6
	Gérer les équipes	Directeur	Supprimer une équipe	6
	Gérer les équipes	Directeur	Modifier une équipe	6
	Gérer les équipes	Directeur	Rechercher une équipe	6
	Gérer les bilans	Directeur	Générer le bilan de l'unité	7

Gestion des sorties pédagogiques et gestion des activités visiteur	Gérer les sorties pédagogiques	Directeur	Ajouter une sortie pédagogique	8
	Gérer les sorties pédagogiques	Directeur	Supprimer une sortie pédagogique	8
	Gérer les sorties pédagogiques	Directeur	Modifier une sortie pédagogique	8
	Gérer les sorties pédagogiques	Directeur	Rechercher une sortie pédagogique	8
	Activité visiteur	Visiteur	Visionner les activités des membres de l'unité	9
	Activité visiteur	Visiteur	Visionner les informations d'un membre de l'unité	9
	Activité visiteur	Visiteur	Afficher la présentation de l'unité	9
	Activité visiteur	Visiteur	Rechercher une activité	9

TABLE 3.3 – Backlog du produit

3.4.3 Planification de Release

Un plan de très haut niveau pour plusieurs sprints est créé pendant la planification de la diffusion. C'est une ligne directrice qui reflète les attentes quant aux fonctionnalités qui seront mises en œuvre et quand elles seront terminées. Il sert également de base pour suivre les progrès au sein du projet. Les sorties peuvent être des livraisons intermédiaires effectuées pendant le projet ou la livraison finale à la fin.

Pour créer un plan de diffusion, les éléments suivants doivent être disponibles :

- Un backlog de produit Scrum hiérarchisé et estimé
- La vitesse de l'équipe Scrum
- Conditions de satisfaction (objectifs pour l'emploi du temps, portée, ressources).

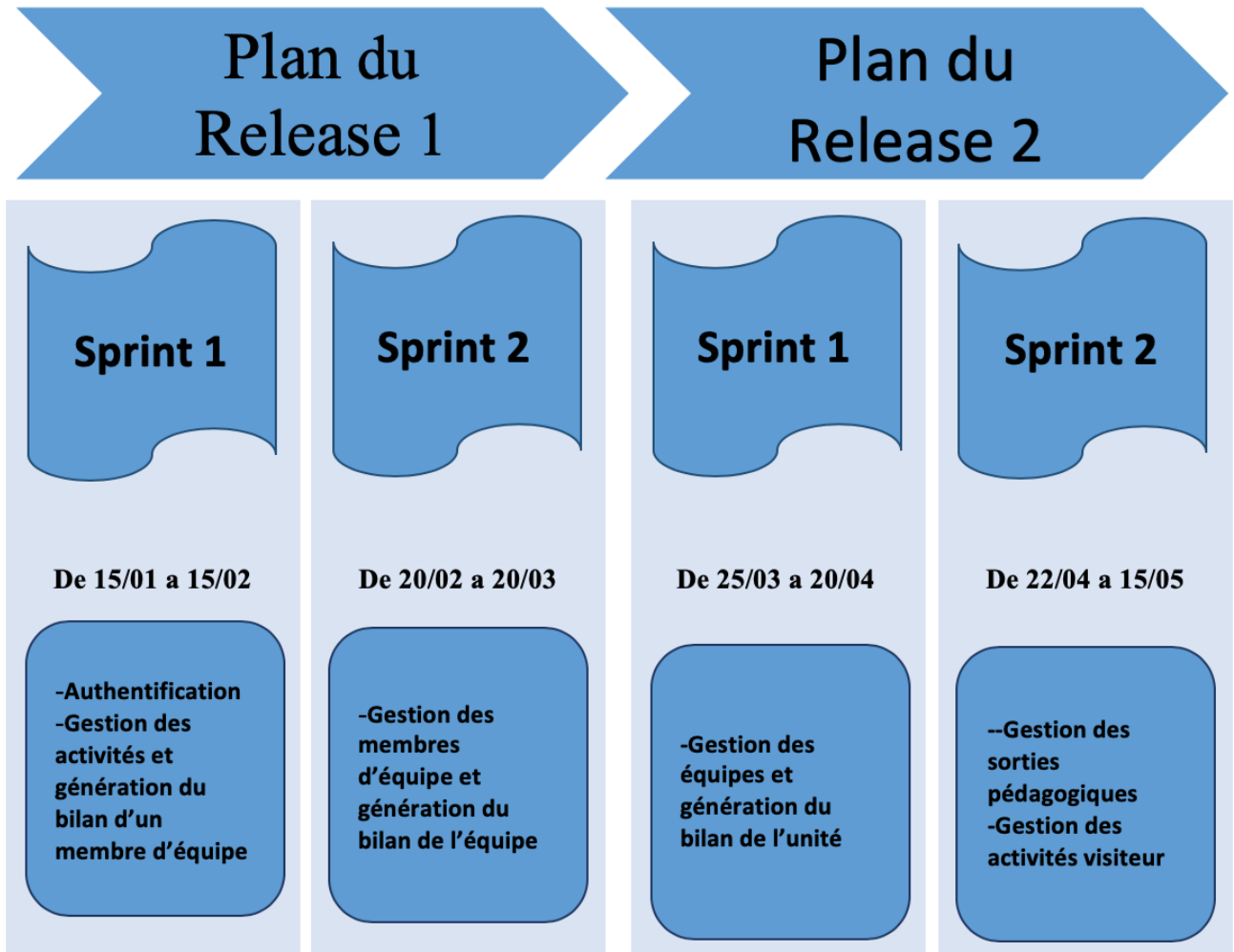


FIGURE 3.2 – Schéma de Releases

3.5 Environnement de travail

Dans cette partie, nous présenterons l'environnement matériel et technique relatif à la réalisation de l'application.

3.5.1 Environnement matériel

L'architecture matérielle utilisée pour la réalisation de l'application est la suivante :

Pc Portable	MacBook pro
RAM	8 Go
Processeur	Dual-Core Intel Core i5
Disque dur	500 Go
Système d'exploitation	MacOs Catalina (10.15.3)

TABLE 3.4 – Environnement matériel

3.5.2 Environnement Logiciel

Au cours de la réalisation de notre application nous avons eu recours à certains outils et langages de programmation ainsi que des bibliothèques que nous allons à présent citer et définir dans ce qui suit. Nous avons choisi ces outils pour leurs compatibilités avec ce que nous voulons réaliser et leur simplicité d'utilisation.

Outils de développement et modélisation

— Android studio

Android Studio est l'environnement de développement intégré (IDE) officiel pour le développement d'applications Android. Il est basé sur IntelliJ IDEA, un environnement de développement intègre JAVA pour les logiciels, et intègre ses outils d'édition et de développement du code. Pour prendre en charge le développement d'applications dans le système d'exploitation Android, Android Studio utilise un système de construction basé sur Gradle, un émulateur, des modèles de code et une intégration GitHub [28].

— XAMPP

XAMPP est un ensemble de logiciels servant à mettre en place aisément un serveur Web, un serveur FTP et un serveur de messagerie électronique. C'est une distribution de logiciels libres (X Apache MySQL Perl PHP) offrant une bonne souplesse d'utilisation, reconnue pour son installation simple et rapide. Ainsi, il est à la portée de la plupart des personnes dans la mesure où il ne requiert pas de connaissances spécifiques et fonctionnelles, qui plus est, sur les dispositifs d'exploitation les plus communs [29].

— **PhpMyAdmin**

PhpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée principalement en PHP et distribuée sous licence GNU GPL [28].

— **StarUML**

Est un logiciel Open Source de modélisation UML, développé en Delphi. Il est modulaire et propose plusieurs générateurs de code. Il propose également plusieurs nouveaux templates de projets [9].

— **Visual Paradigm**

Visuel Paradigm est un logiciel de création de diagrammes dans le cadre d'une programmation. Tout en un, il possède plusieurs options permettant une large possibilité de modélisation en UML [30].

Langages de développement

— **JAVA**

C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (JSP) [31].

— **PHP**

Le PHP(HyperText Preprocessor) désigne un langage informatique, ou un langage de script, utilisé principalement pour la conception de sites web dynamiques. Il s'agit d'un langage de programmation sous licence libre qui peut donc être utilisée par n'importe qui de façon totalement gratuite [32].

— SQL

Le langage SQL (Structured Query Language) est un langage informatique utilisé pour exploiter des bases de données. Il permet de façon générale la définition, la manipulation et le contrôle de sécurité de données [31].

— XML

Le XML, acronyme d'eXtensible Markup Language (qui signifie : langage de balisage extensible), est un langage informatique qui sert à enregistrer des données textuelles. Ce langage a été standardisé par le W3C en février 1998 et il est maintenant très populaire. Ce langage, grosso-modo similaire à l'HTML de par son système de balisage, permet de faciliter l'échange d'informations sur l'internet. Contrairement à l'HTML qui présente un nombre fini de balises, le XML donne la possibilité de créer de nouvelles balises à volonté [32].

— JSON

JSON (JavaScript Object Notation – Notation Objet issue de JavaScript) est un format léger d'échange de données. Il est facile à lire ou à écrire pour des humains. Il est aisément analysable ou générable par des machines. Il est basé sur un sous-ensemble du langage de programmation JavaScript. JSON est un format texte complètement indépendant de tout langage, mais les conventions qu'il utilise seront familières à tout programmeur habitué aux langages descendants du C, comme par exemple : C lui-même, C++, JAVA, JavaScript, Perl, Python et bien d'autres. Ces propriétés font de JSON un langage d'échange de données idéal. JSON se base sur deux structures :

- Une collection de couples nom/valeur. Divers langages la réifient par un objet, un enregistrement, une structure, un dictionnaire, une table de hachage, une liste typée ou un tableau associatif.
- Une liste de valeurs ordonnées. La plupart des langages la réifient par un tableau, un vecteur, une liste ou une suite [33].

3.6 Architecture de l'application

Dans cette section, nous commençons par la présentation de l'architecture MVC que nous avons choisie pour la réalisation de notre application. Bien évidemment, le choix de l'architecture adéquate dans la phase de conception de toute application est primordial, afin de garantir un fonctionnement correct, une meilleure performance et une maintenance facile.

Dans cette logique nous avons opté pour l'architecture MVC.

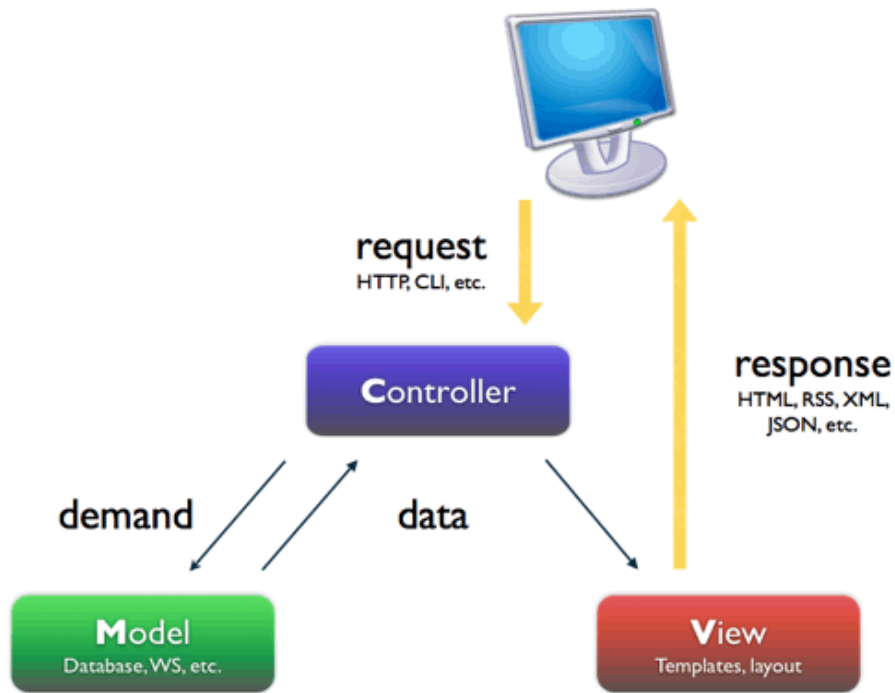


FIGURE 3.3 – Architecture MVC

L'architecture la plus utilisée est l'architecture MVC qui signifie Modèle, vue et contrôleur pour mieux organiser le code de notre système.

Cette architecture, comme dans l'illustration en haut, elle est séparée sur trois couches : modèle, vue et contrôleur.

Le modèle définit les données utilisées par l'application. En effet, c'est ici que le lien se fera entre notre application et la base de données. Par exemple, nous pourrions trouver les utilisateurs ou encore les différents articles pour un site de ventes en ligne. Ces données pourront être mises à jour dans le contrôleur et affichées au niveau de la vue [4].

La vue définit la façon dont les informations seront affichées à l'écran (via des composants par exemple). Il s'agit de l'interface utilisateur. C'est ici que nous utiliserons les données récupérées par le modèle afin de les présenter à l'utilisateur. Par exemple, pour un site de ventes en ligne, ce serait la page du produit qui s'affiche à l'écran [4].

Le contrôleur Dans le contrôleur, nous retrouvons toute la logique métier. En effet, lorsque l'utilisateur interagit avec la vue, la requête est traitée par le contrôleur. Il fonctionne comme un "listener", c'est-à-dire qu'il attend que l'utilisateur interagisse avec la vue pour récupérer la requête.

Ainsi, c'est le contrôleur qui définira la logique d'affichage, et affichera la vue suivante à l'écran [4].

3.7 Conclusion

Dans ce chapitre, nous avons planifié notre travail, identifié les besoins fonctionnels et non fonctionnels de notre application, les rôles des utilisateurs, par la suite nous avons présenté le backlog de notre système. Ainsi nous avons détaillé la phase de planification des sprints. Enfin nous avons choisi l'architecture de notre projet ainsi que notre environnement de travail.

Chapitre 4

Release 1

4.1 Introduction

Dans ce présent chapitre, nous allons détailler le travail réalisé durant la première release. En effet, chaque release, qui est l'ensemble de sprints, représente une vision distribuée de la période de la production du livrable. Cette première release comprend deux sprints :

- Sprint1 « Authentification, Gestion des activités et génération du bilan d'un membre d'équipe »
- Sprint 2 « Gestion des membres d'équipe et génération du bilan de l'équipe »

Le développement de chaque sprint passe par les étapes d'analyse, de conception et de réalisation.

4.2 Développement du Sprint 1 « Authentification, Gestion des activités et génération du bilan d'un membre d'équipe »

Ce premier sprint s'étale sur 30 jours et se décompose en trois items

- S'authentifier
- Gérer les activités
- Générer le bilan d'un membre d'équipe

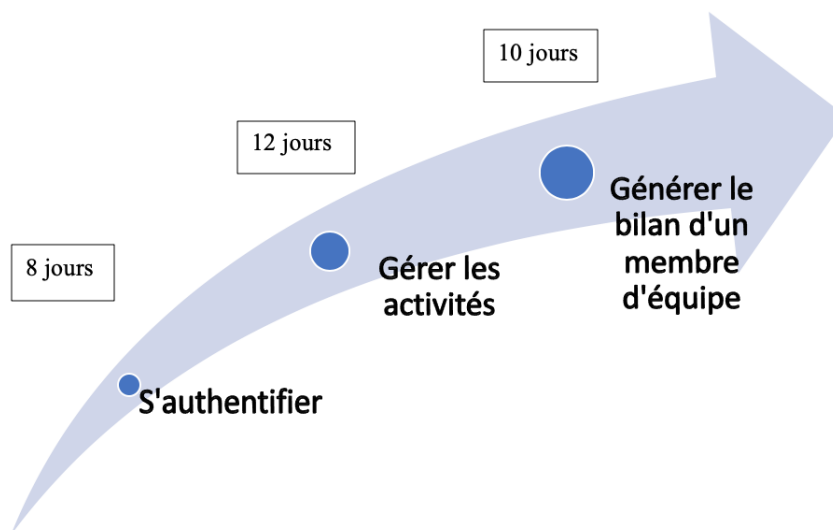


FIGURE 4.1 – Décomposition du sprint 1 en items

Une user story est la description simplifiée d’un besoin. En développement agile, vous pourrez toujours l’affiner selon la maturité et l’urgence de la demande.

Dans le tableau qui suit on présente les user stories du sprint 1 :

ID U. S	User Story
1	En tant qu'utilisateur, je dois m'authentifier pour accéder à mon espace
2.1	En tant que membre de l'équipe, je peux ajouter une activité
2.2	En tant que membre de l'équipe, je peux modifier une activité
2.3	En tant que membre de l'équipe, je peux supprimer une activité
2.4	En tant que membre de l'équipe, je peux chercher une activité
3.1	En tant que membre de l'équipe, Je peux générer le bilan de mes activités

TABLE 4.1 – Le sprint planning de sprint 1

Le backlog du sprint 1 est le suivant :

Item	User Story	Description	Priorité
S'authentifier	S'authentifier	En tant qu'utilisateur, je peux m'authentifier	1
Gérer les activités et générer le bilan d'un membre d'équipe	Ajouter une activité	En tant que membre de l'équipe, je peux ajouter une activité	2
	Modifier une activité	En tant que membre de l'équipe, je peux Modifier une activité	
	Supprimer une activité	En tant que membre de l'équipe, je peux supprimer une activité	
	Chercher une activité	En tant qu'utilisateur, je peux chercher une activité	
Générer le bilan d'un membre d'équipe	Générer le bilan d'un membre d'équipe	En tant que membre de l'équipe, Je peux générer le bilan de mes activités	3

TABLE 4.2 – Backlog du sprint 1

4.2.1 Analyse

Dans la phase d'analyse nous allons présenter les diagrammes de cas d'utilisation et leurs descriptions.

Diagramme de cas d'utilisation

La figure suivante décrit de façon globale les cas d'utilisation identifiés pour le sprint 1

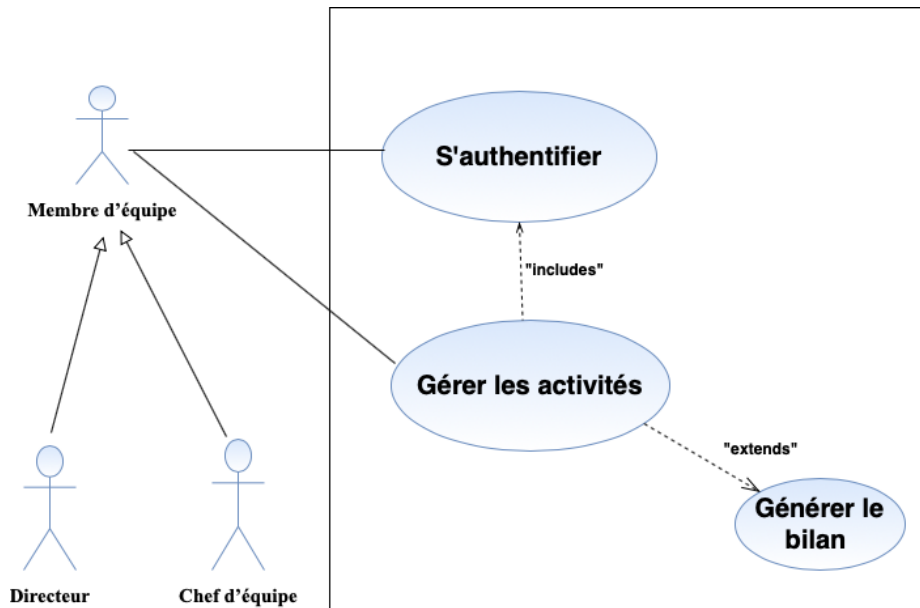


FIGURE 4.2 – Diagramme de cas d'utilisation du sprint 1

Description de l'item « S'authentifier »

Dans ce qui suit nous allons décrire le premier item du sprint 1 qui y'a duré 8 jours.

1. Diagramme de cas d'utilisation « S'authentifier »

Le figure ci-dessous représente le diagramme des cas d'utilisation "S'authentifier".

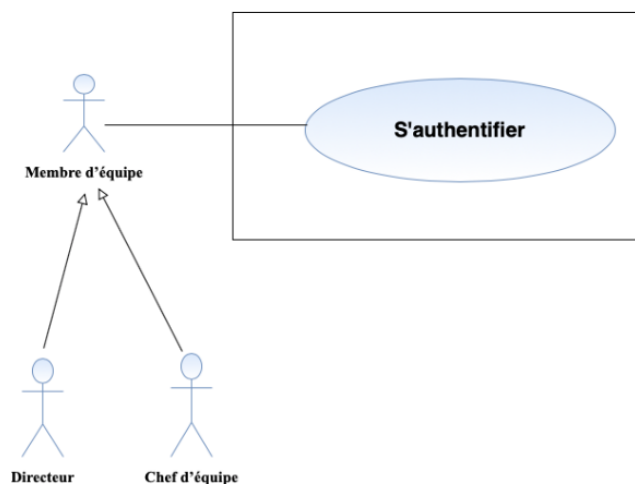


FIGURE 4.3 – Diagramme de cas d'utilisation « S'authentifier »

2. Description textuelle du cas d'utilisation « S'authentifier »

La description détaillée de l'user story 1 « s'authentifier » est donnée par le tableau suivant :

Cas utilisation	S'authentifier
Acteur	Membre d'équipe, Chef d'équipe, Directeur
Précondition	L'acteur doit être activé.
Objectif	Vérifier que l'utilisateur a bien le droit d'accès à l'application
Scenario Nominal	<ol style="list-style-type: none">1. L'acteur saisit son email et son mot de passe2. L'acteur confirme la saisie de ses données d'identification3. Le système vérifie les données d'identification4. Le système affiche l'interface d'accueil selon le type de l'acteur (Membre d'équipe, Chef d'équipe, Directeur).
Alternative [A1]	Si l'email et/ou le mot de passe sont incorrectes ou l'utilisateur n'existe pas, le système affiche un message d'erreur informant l'utilisateur que son email ou mot de passe sont incorrects et réaffiche la page d'authentification

TABLE 4.3 – Description textuelle du cas d'utilisation « S'authentifier »

Description de l'item « Gérer les activités »

Dans ce qui suit nous allons décrire le second item du sprint 1 qui y'a duré 12 jours.

1. Diagramme de cas d'utilisation « Gérer les activités »

Le diagramme ci-dessous représente le diagramme des cas d'utilisation "Gérer les activités".

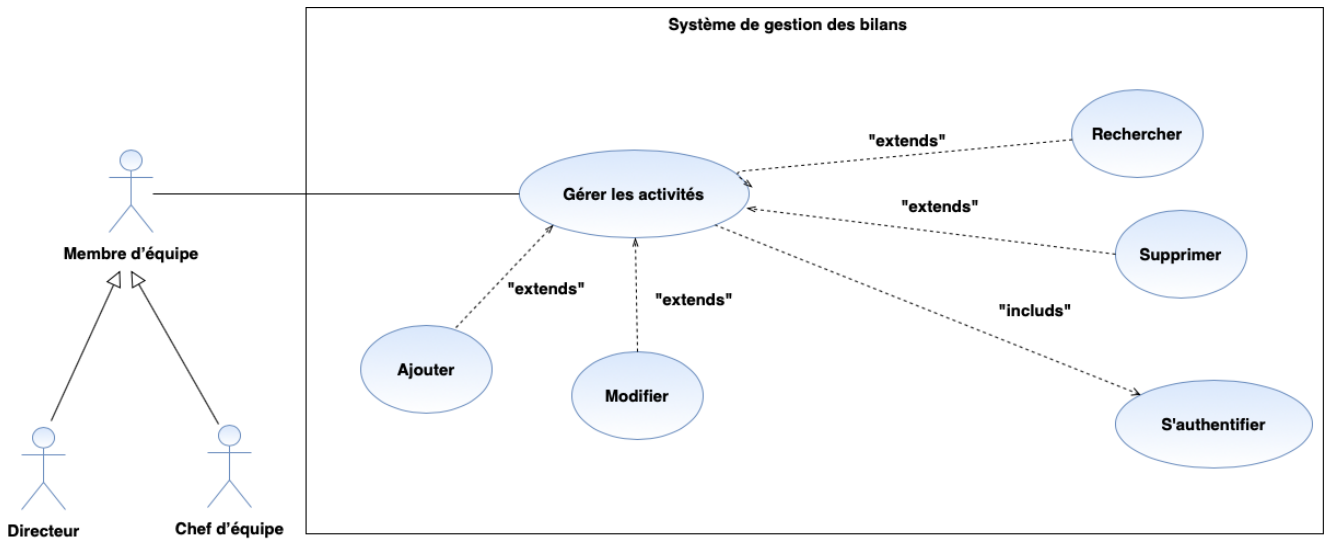


FIGURE 4.4 – Diagramme de cas d'utilisation « Gérer les activités »

2. Description textuelle du cas d'utilisation « Ajouter une activité »

La description détaillée de l’user story 2.1 « Ajouter une activité » est donnée par le tableau suivant :

Cas utilisation	Ajout une activité
Acteur	Membre d’équipe, Chef d’équipe, Directeur
Précondition	L’acteur doit être authentifié.
Objectif	Mise à jour de la liste des activités
Scenario Nominal	<ol style="list-style-type: none"> 1. L’acteur clique sur bouton « Ajouter activité » 2. Le système affiche l’interface d’ajout 3. L’acteur remplit les champs 4. Le système vérifie les données saisies par l’acteur 5. Le système enregistre les informations 6. Le système notifie l’acteur
Alternative [A1]	Si un champ d’information n’est pas valide ou l’activité existe déjà, le système affiche un message d’erreur

TABLE 4.4 – Description textuelle du cas d'utilisation « Ajouter une activité »

Description de l'item « Générer le bilan d'un membre d'équipe »

Dans ce qui suit nous allons décrire le troisième item du sprint 1 qui y'a duré 10 jours.

1. Diagramme de cas d'utilisation « Générer le bilan d'un membre d'équipe »

La figure ci-dessous représente le diagramme des cas d'utilisation "Générer le bilan d'un membre d'équipe".

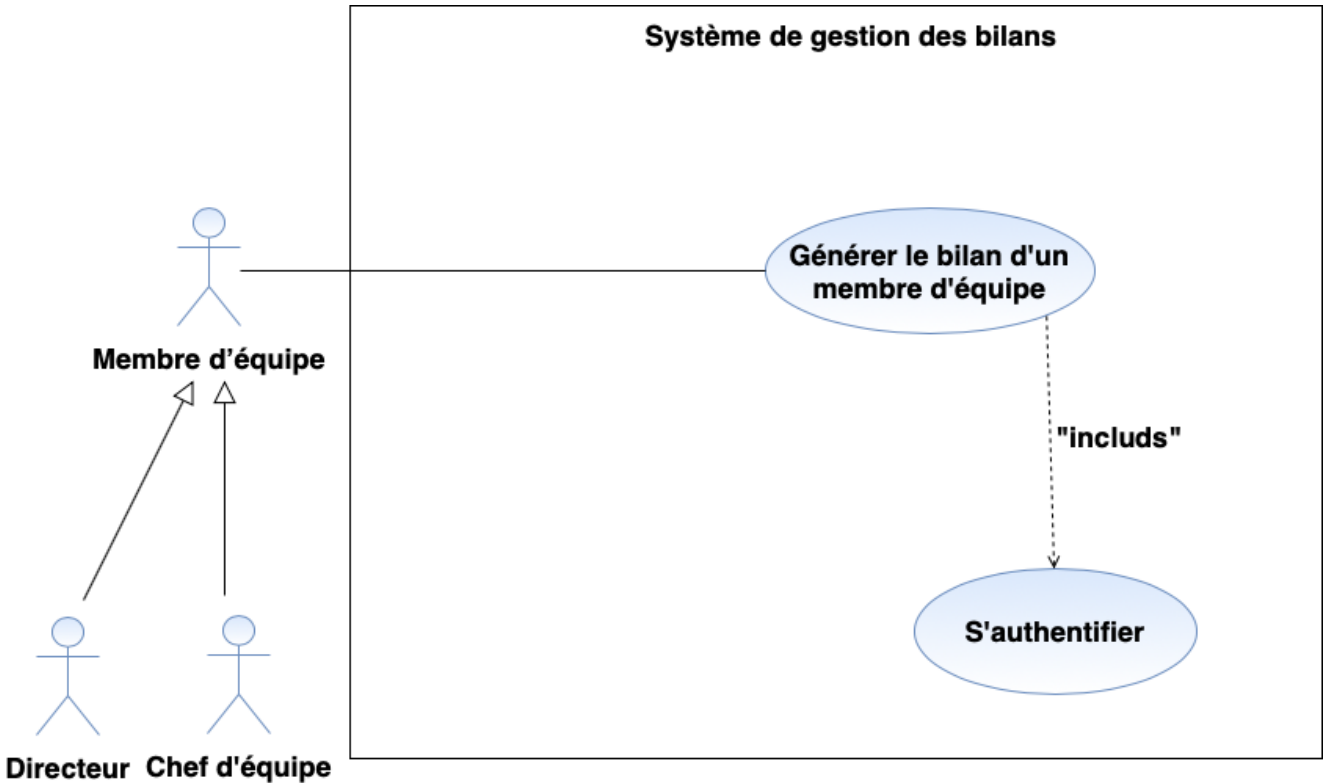


FIGURE 4.5 – Diagramme de cas d'utilisation « Générer le bilan d'un membre d'équipe »

2. Description textuelle du cas d'utilisation « Générer le bilan d'un membre d'équipe »

La description détaillée de l'user story 3.1 « générer le bilan d'un membre d'équipe » est donnée par le tableau suivant

Cas utilisation	Générer le bilan d'un membre d'équipe
Acteur	Membre d'équipe, Chef d'équipe, Directeur
Précondition	L'acteur doit être authentifié.
Objectif	Bilan d'un membre de l'unité
Scenario Nominal	<ol style="list-style-type: none"> 1. L'acteur clique sur le menu 2. L'acteur choisir le bouton « Générer Mon Bilan » 3. Le système vérifie l'existence des activités pour cette acteur 4. Le système notifie l'acteur 5. Le système remet à l'acteur un bilan
Alternative [A1]	Si aucune activité n'existe pour cette acteur, le système affiche un message d'erreur

TABLE 4.5 – Description textuelle du cas d'utilisation « Générer le bilan d'un membre d'équipe »

4.2.2 Conception

Dans la phase de conception nous allons décrire les interactions entre les acteurs de l'application et le système. Pour la réalisation des cas d'utilisation décrit dans la phase d'analyse.

Diagrammes de séquence

Les diagrammes de séquence permettent de présenter les interactions entre les objets selon un point de vue temporel. L'accent est mis sur la chronologie des envois de messages [34].

1. Diagramme de séquence « S'authentifier »

L'utilisateur doit s'authentifier en saisissant son email et mot de passe, puis le système vérifie les informations introduites, si l'une des coordonnées est non valide le système affiche un message d'erreur sinon il envoie une requête de vérification au serveur qui vérifie l'existence de l'utilisateur dans le SGBD, en cas d'erreur le système affiche un message d'erreur, sinon ce dernier affiche une interface correspondant au rôle de l'utilisateur.

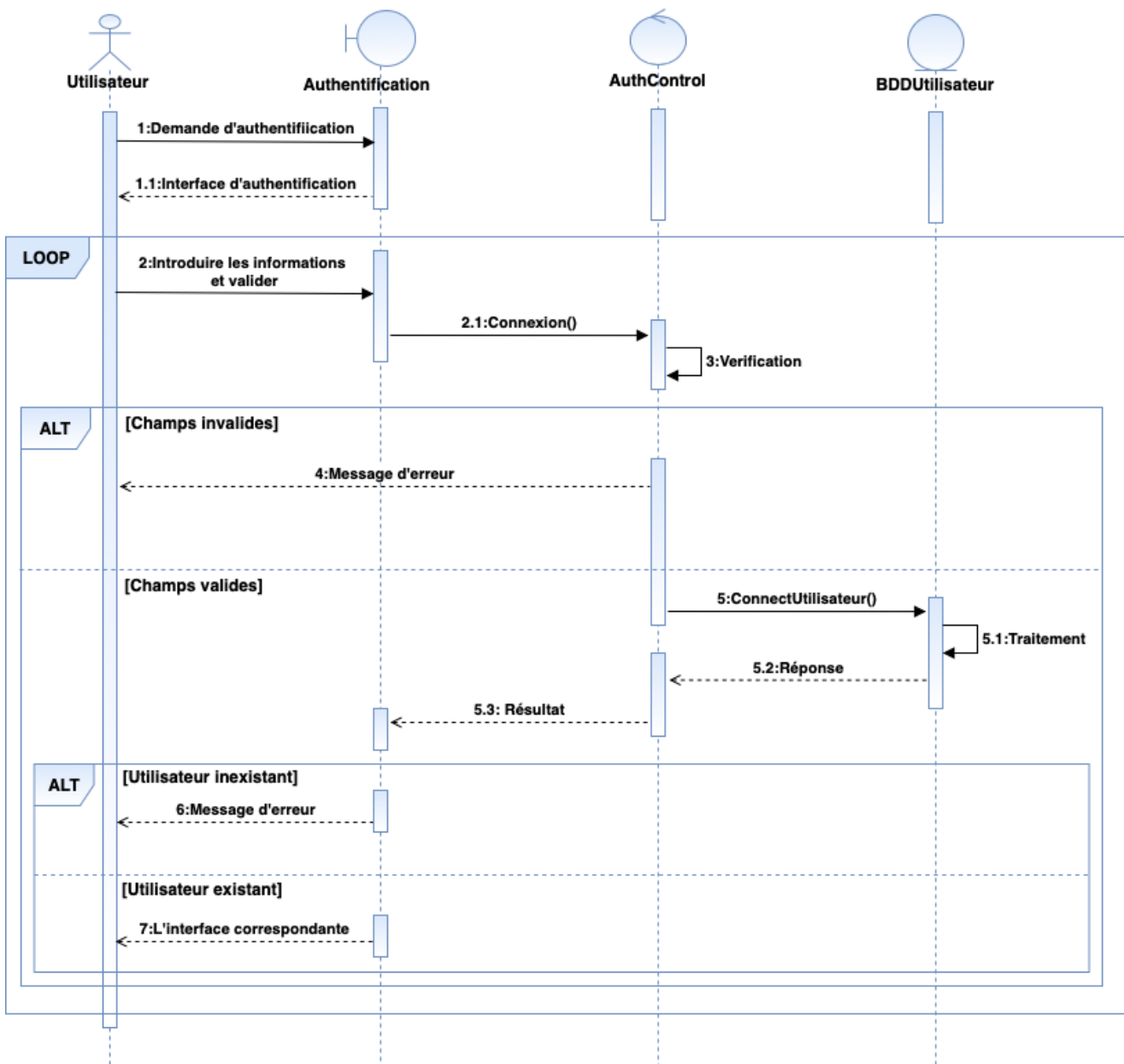


FIGURE 4.6 – Diagramme de séquence « S'authentifier »

2. Diagramme de séquence « Ajouter une activité »

Pour effectuer des mises à jour de la liste des activités, le membre de l'équipe doit s'authentifier, ensuite ajouter une activité en remplissant un formulaire d'ajout.

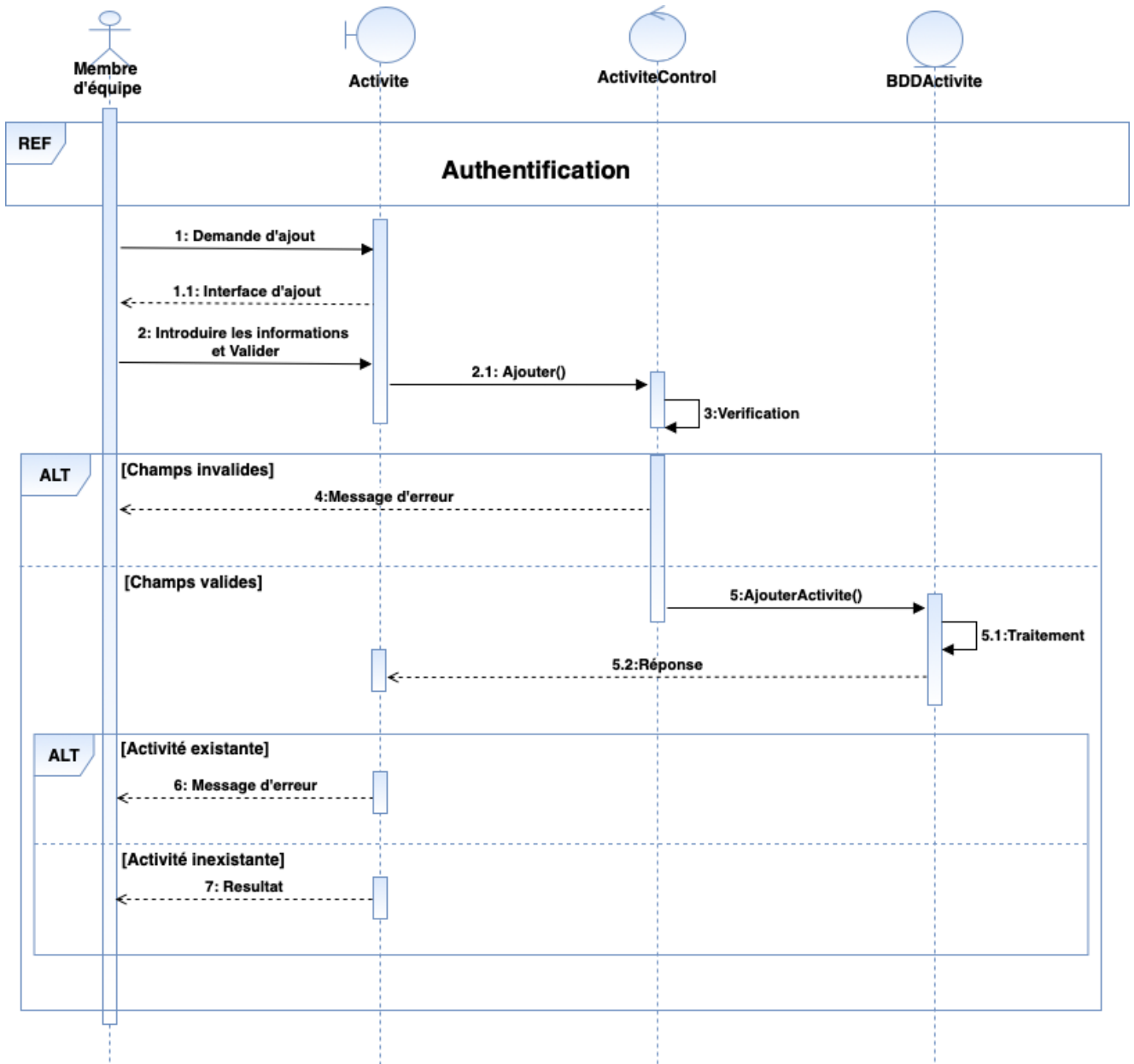


FIGURE 4.7 – Diagramme de séquence « Ajouter une activité »

3. Diagramme de séquence « Générer le Bilan d'un membre d'équipe »

Afin qu'un membre d'équipe puisse générer le bilan de ses activités il doit s'authentifier puis cliquer sur le bouton « générer mon Bilan ».

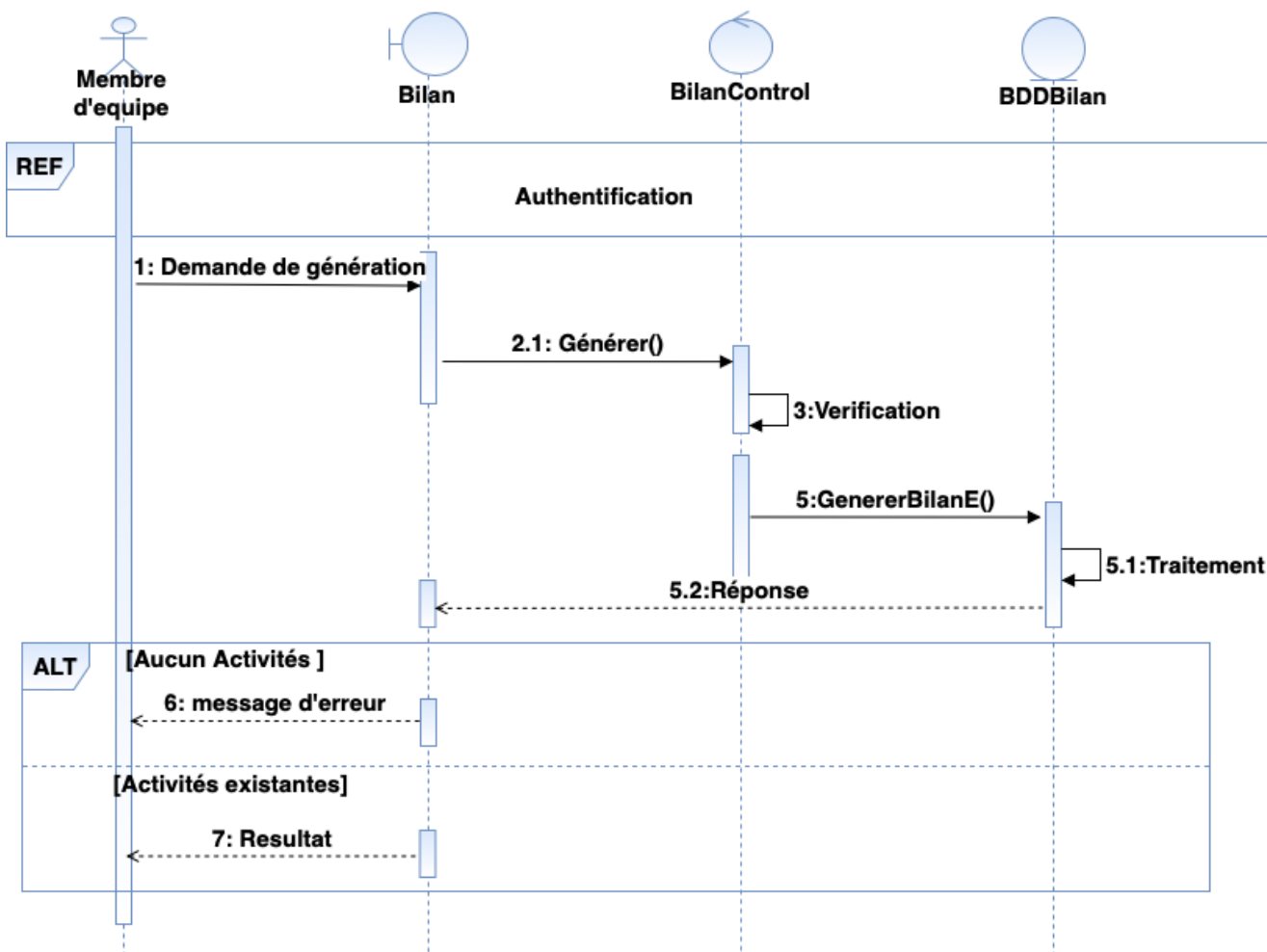


FIGURE 4.8 – Diagramme de séquence « Générer le bilan d'un membre d'équipe »

Dictionnaire de données

Le tableau suivant représente le dictionnaire de données :

Classe	Attributs	Signification	Type	Taille
Utilisateur	idU	Identifiant de l'utilisateur	INT	30
	nomU	Nom utilisateur	String	30
	prénom	Prénom utilisateur	String	30
	datedenais	Date de naissance de l'utilisateur	Date	30
	sexe	Sexe de l'utilisateur (masculin ou féminin)	String	15
	password	Mot de passe de l'utilisateur	String	30
	adresseU	Adresse de l'utilisateur	String	50
	pays	Pays de l'utilisateur	String	30
	telephone	Numéro de téléphone de l'utilisateur	String	30
	codepostal	Code postal de l'utilisateur	String	30
Activité	idA	Identifiant de l'activité	INT	30
	typeA	Type de l'activité (publication , conférence, ...)	String	30
	titre	Titre de l'activité	String	30
	Chapitre	Les chapitres de l'activité	String	30
	nbrepages	Nombre de page de l'activité	INT	30
	volume	Volume de l'activité	INT	30
	nombre	Nombre de l'activité	INT	30
	échelle	Échelle de l'activité	String	30
	datepub	Date de publication	Date	30
	AdresseA	Adresse de l'activité	String	30
	editionA	Edition de l'activité	String	30
	note	Note sur l'activité	String	50
	URL	URL pour le fichier pdf correspondant à l'activité	String	50
	motscles	Mots clés sur l'activité	String	50
	ville	Ville de l'activité	String	30
Bilan	idB	Identifiant du bilan	INT	30
	typeB	Type de bilan (bilan de l'équipe , bilan de l'unité...)	String	30
Membre d'équipe	idM	Identifiant du membre de l'équipe	INT	30
Chef d'équipe	idC	Identifiant du chef d'équipe	INT	30
Directeur	idB	Identifiant du directeur	INT	30

TABLE 4.6 – Le dictionnaire de données

Diagramme de classes

Dans la figure suivante nous présentons les classes associées à ce sprint.

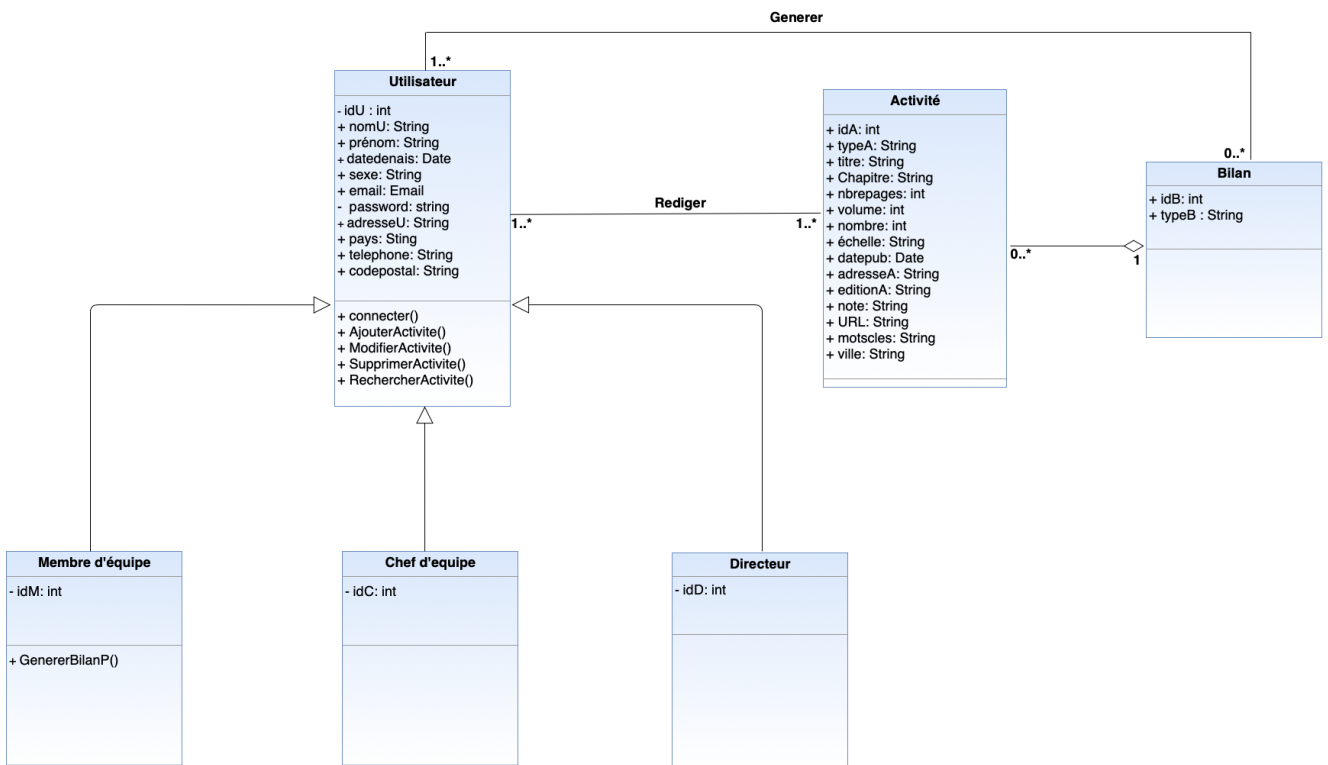


FIGURE 4.9 – Diagramme de classe du sprint 1

Le modèle relationnel

Le modèle relationnel est une manière de modéliser les relations existantes entre plusieurs informations et de les ordonner entre elles.

1. Passage au modèle relationnel

Les règles utilisées pour le passage du diagramme de classe de notre application au modèle relationnel

Règle 1 : transformation de classes

Chaque classe devient une relation, l’identifiant (respectivement les attributs) de la classe devient la clé primaire (respectivement des attributs) de la relation.

Règle 2 : association un a plusieurs

Il faut ajouter un attribut de type clé étrangère dans la relation fils de l’association l’attribut porte le nom de la clé primaire de la relation père de l’association.

Règle 3 : association plusieurs a plusieurs

L'association devient une relation dont la clé primaire est composée par la concaténation des identifiants des classes connectés à l'association, les attributs de l'association doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère.

Règle 4 : Association un-a-un

Il faut ajouter un attribut clé étrangère, dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association.

Règle 5 : Transformation de l'héritage

Trois décompositions sont possibles pour traduire une association d'héritage en fonction des contraintes à l'association.

- **Décomposition par distinction** : créer une relation par classe et les relier par des associations.
- **Décomposition descendante (push-down)** : S'il existe une contrainte de totalité ou de Partition sur l'association d'héritage, il est possible de ne pas traduire la relation issue de la sous-classe. Il faut alors faire migrer tous ces attributs dans la (les) relation(s) issue(s) de la (des) surclasse(s).
- **Décomposition ascendante (push-up)** : Il faut supprimer la (les) relation(s) issue(s) de la (des) surclasse(s) et faire migrer les attributs Dans la relation issue de la surclasse.

Règle 6 : Transformation d'une composition

La clé primaire des relations déduites des classes composantes doit contenir l'identifiant de la classe composite (quelles que soient les multiplicités) [35].

2. Schéma relationnel

La base de données relationnelle est une base de données comprenant des relations dynamiques entre les différents objets contenus dans les tables :

Utilisateur (idU, nomU, prenomU , datedenais , sexe , email, password , adresseU , pays. telephone) .

Membre d'équipe (idM, #idU, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays. telephone) .

Chef d'équipe (idC, #idU, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays. telephone) .

Directeur (idD, #idU, nomU, prenomU, datedenais, sexe, email, password , adresseU , pays. telephone) .

Activité (idA, typeA, titre, Chapitre, nbrepages, volume, nombre, échelle, datepub, adresseA, editionA, note, URL, ville, #idB)

Rediger (#idU, #idA)

Bilan (idB, typeB)

Generer (#idU, #idB)

4.2.3 Réalisation

Dans la phase de réalisation nous allons décrire les interfaces de notre application.

Description des interfaces d'un membre d'équipe

Nous présentons dans ce qui suit quelques interfaces représentant le travail élaboré dans ce sprint.

1. Interface d'authentification

Pour se connecter, l'utilisateur saisit son login et mot de passe .

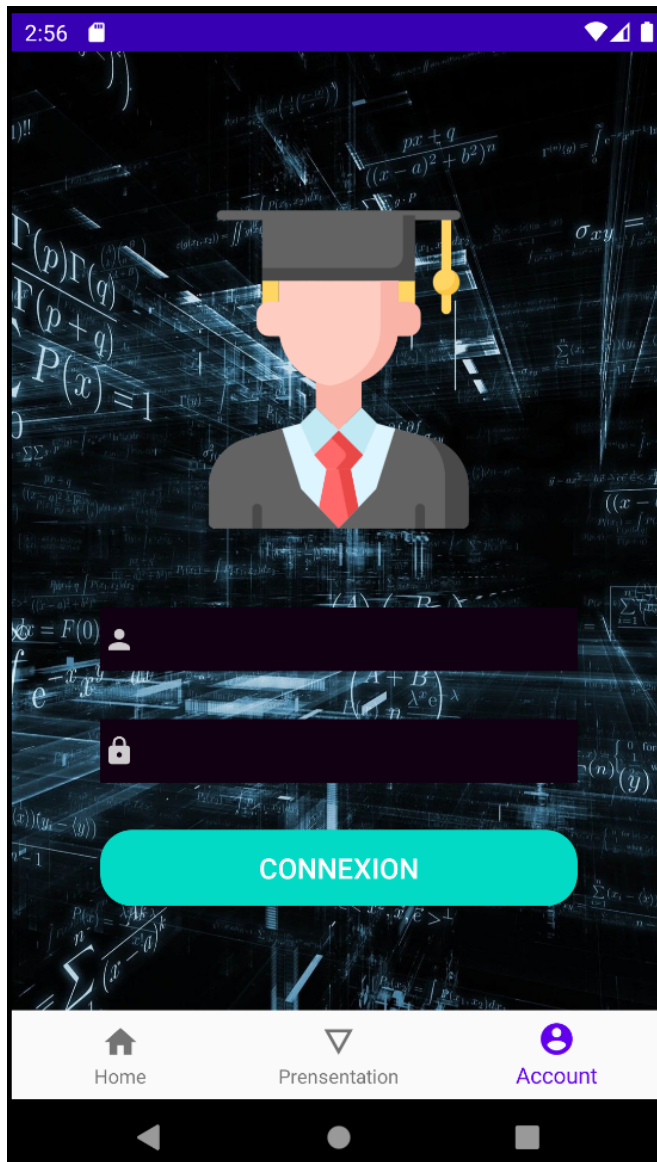


FIGURE 4.10 – Interface d'authentification

2. Interface d'ajout d'une activité

Après la connexion, l'utilisateur peut ajouter une activité en remplissant un formulaire d'ajout.

3:55

Ajouter Une Activite

Remplir les champs suivants

Type Activite

Titre Activite

Chapitre Nombre de pages

Volume Nombre

Adresse Ville

Date De Publication

Edition Note

Echelle National International

Auteur(s)

Joindre le fichier pdf

FIGURE 4.11 – interface d'ajout d'une activité

3. Interface du Bilan d'un Membre d'équipe

Cette interface illustre les différentes activités ajoutées par le même membre d'équipe qui forme son bilan.

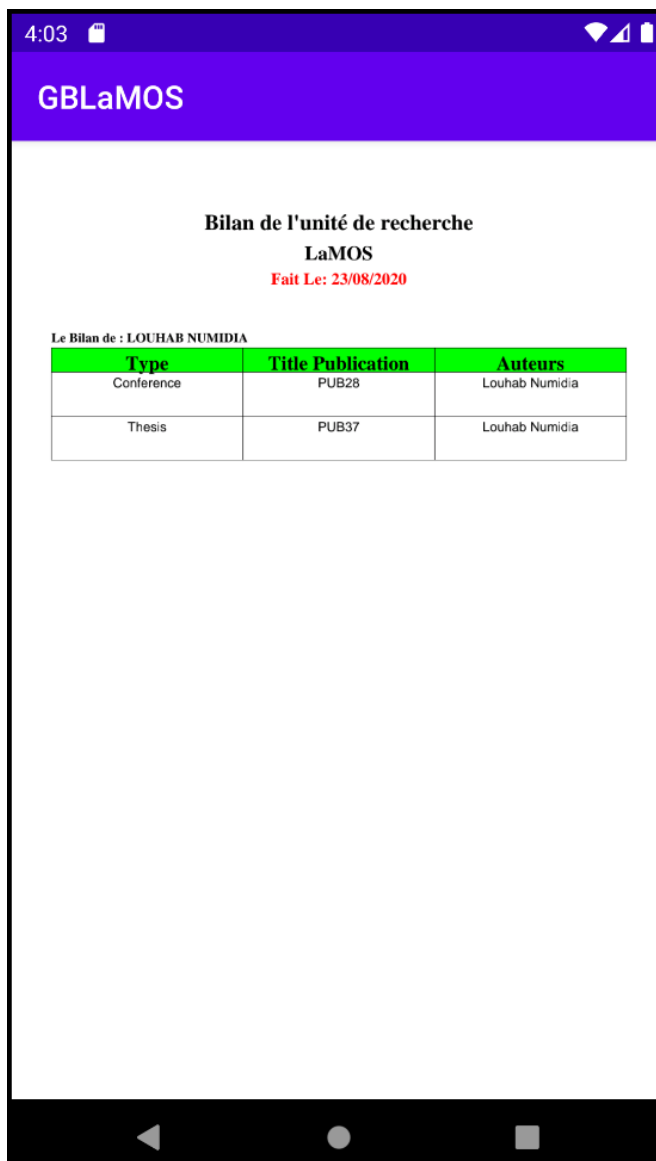


FIGURE 4.12 – interface du Bilan d'un membre d'équipe

4.3 Développement du sprint 2 « gestion des membres d'équipe et génération du bilan de l'équipe »

Ce deuxième sprint s'étale sur 30 jours et se décompose en deux items :

- Gérer les membres d'équipe
- Générer le bilan de l'équipe

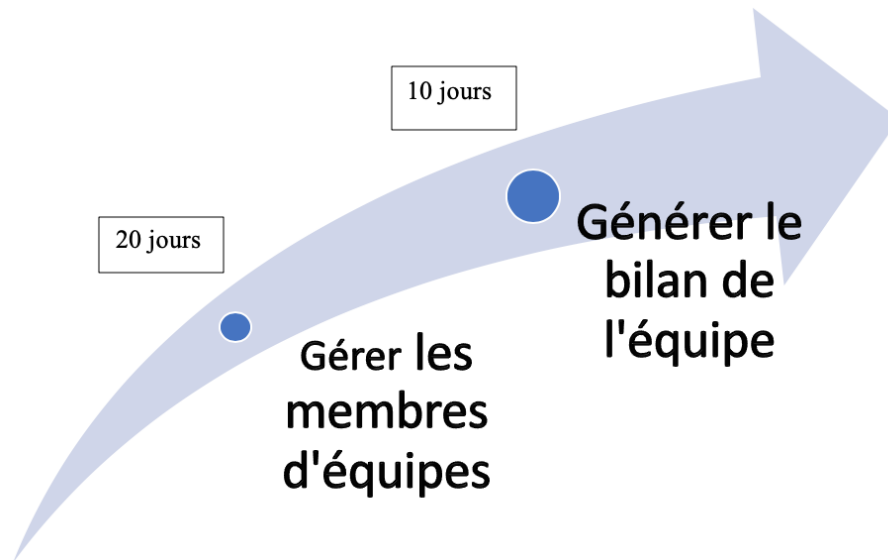


FIGURE 4.13 – Décomposition du sprint 2 en items

Une user story est la description fonctionnelle utilisée dans les méthodes agiles. L'objectif est de spécifier le développement d'une fonctionnalité. Elle doit alors exprimer à qui elle s'adresse et en quoi elle apporte de la valeur [36].

Dans le tableau qui suit on présente les user stories du sprint 2 :

ID U. S	User Story
4.1	En tant que chef d'équipe, je peux ajouter un membre d'équipe
4.2	En tant que chef d'équipe, je peux modifier un membre d'équipe
4.3	En tant que chef d'équipe, je peux supprimer un membre d'équipe
4.4	En tant que chef d'équipe, je peux chercher un membre d'équipe
5.1	En tant que chef d'équipe, Je peux générer le bilan de l'équipe

TABLE 4.7 – Le sprint planning du sprint 2

Le backlog du sprint 2 est le suivant :

Item	User Story	Description	Priorité
Gérer les membres d'équipe	Ajouter un membre d'équipe	En tant que chef d'équipe, je peux ajouter un membre d'équipe	4
	Modifier un membre d'équipe	En tant que chef d'équipe, je peux Modifier un membre d'équipe	
	Supprimer un membre d'équipe	En tant que chef d'équipe, je peux supprimer un membre d'équipe	
	Chercher un membre d'équipe	En tant que chef d'équipe, je peux chercher un membre d'équipe	
Générer le bilan d'équipe	Générer le bilan d'équipe	En tant membre de l'équipe, Je peux générer le bilan de mes activités	5

TABLE 4.8 – Backlog de sprint 2

4.3.1 Analyse

Dans la phase d'analyse nous allons présenter les diagrammes de cas d'utilisation et leurs descriptions.

Diagramme de cas d'utilisation

La figure suivante décrit de façon globale les cas d'utilisation identifiés pour le sprint 2.

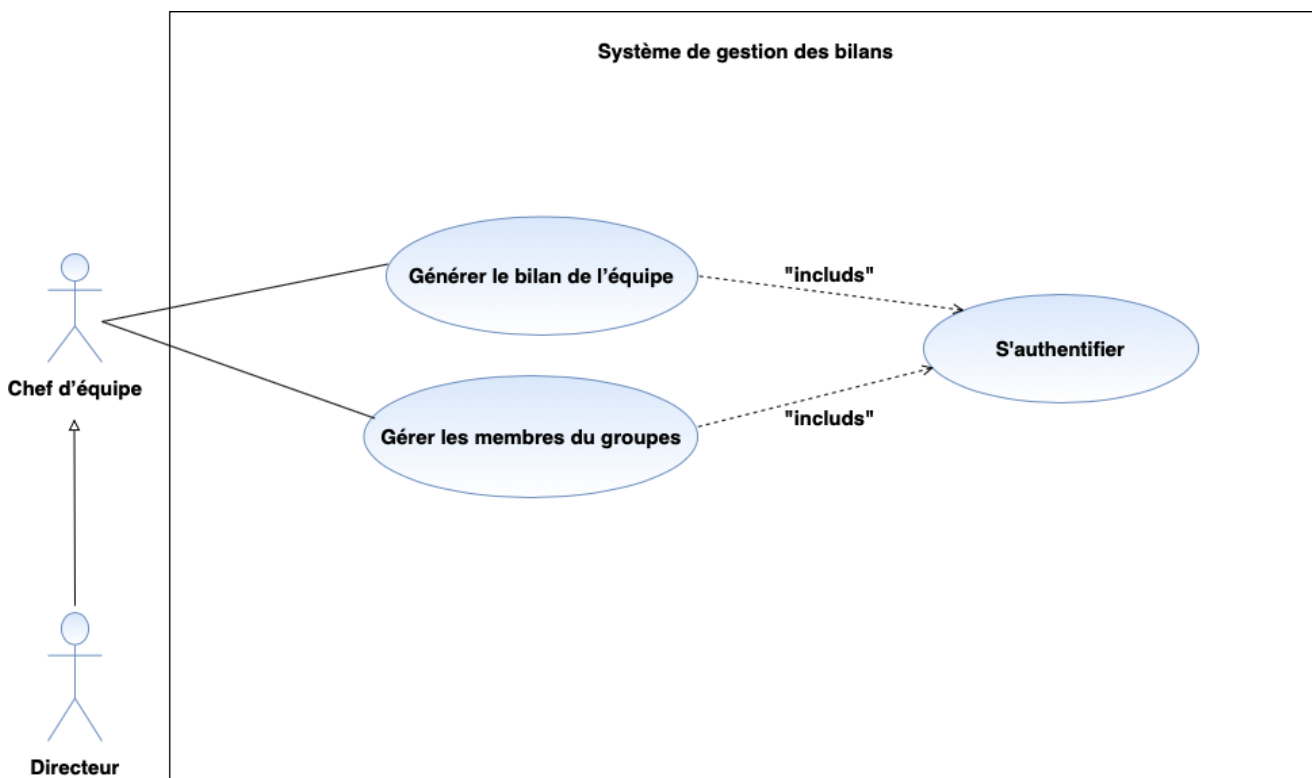


FIGURE 4.14 – Diagramme de cas d'utilisation du sprint 2

Description de l'item « Gérer les membres de l'équipe »

Dans ce qui suit nous allons décrire le premier item du sprint 2 qui y'a duré 20 jours.

1. Diagramme de cas d'utilisation « Gérer les membres de l'équipe »

La figure ci-dessous représente le diagramme des cas d'utilisation "Gérer les membres d'équipe".

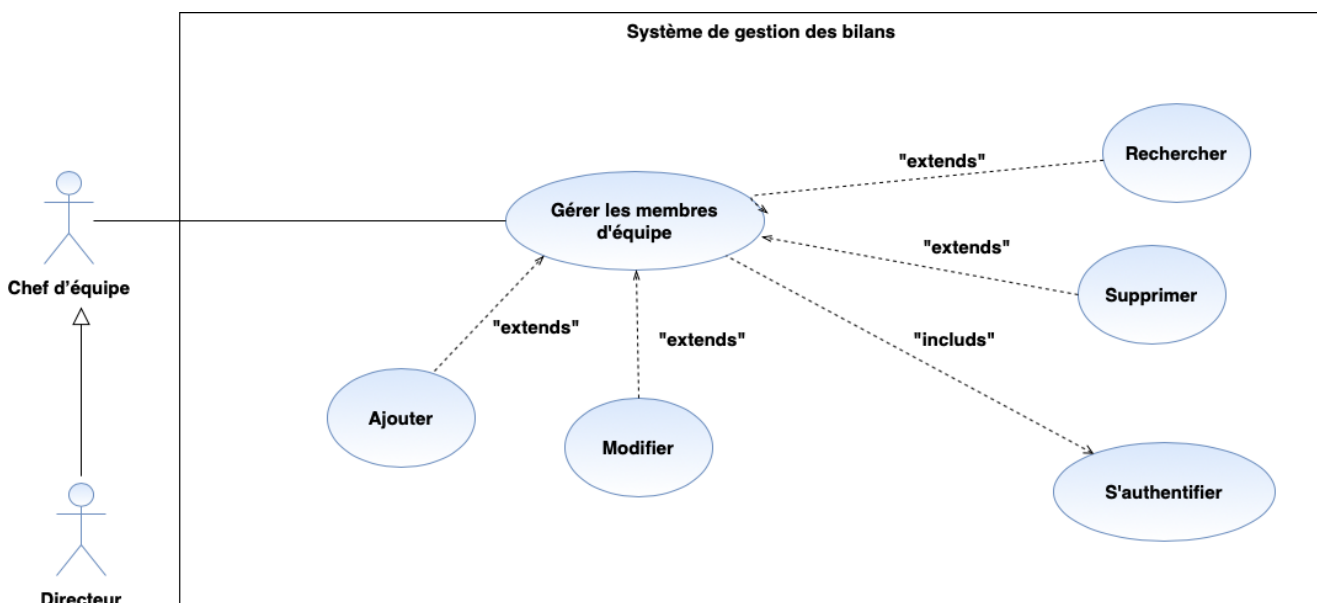


FIGURE 4.15 – Diagramme de cas « Gérer les membres d'équipe »

2. Description textuelle du cas d'utilisation « Ajouter un membre d'équipe »

La description détaillée de l'user story 4.1 « Ajouter une membre d'équipe » est donnée par le tableau suivant :

Cas utilisation	Ajout un membre d'équipe
Acteur	Chef d'équipe, Directeur
Précondition	L'acteur doit être authentifié.
Objectif	Mise à jour de la liste des membres d'équipe
Scénario Nominal	<ol style="list-style-type: none"> 1. L'acteur clique sur le menu 2. L'acteur choisir le bouton « Utilisateurs » 3. L'acteur clique sur bouton « Ajouter un membre d'équipe » 4. Le système affiche l'interface d'ajout 5. L'acteur rempli les champs 6. Le système vérifier les données saisies par l'acteur 7. Le système enregistre les informations 8. Le système notifie l'acteur
Alternative [A1]	Si un champ d'information n'est pas valide ou le membre existe déjà dans l'équipe, le système affiche un message d'erreur

TABLE 4.9 – Description textuelle du cas d'utilisation « Ajouter un membre d'équipe »

Description de l’item « Générer le bilan de l’équipe »

Dans ce qui suit nous allons décrire le deuxième item du sprint 2 qui y’a duré 10 jours.

1. Diagramme de cas d’utilisation « Générer le bilan d’équipe »

Le diagramme ci-dessous représente le diagramme des cas d’utilisation ”Générer le bilan d’équipe”.

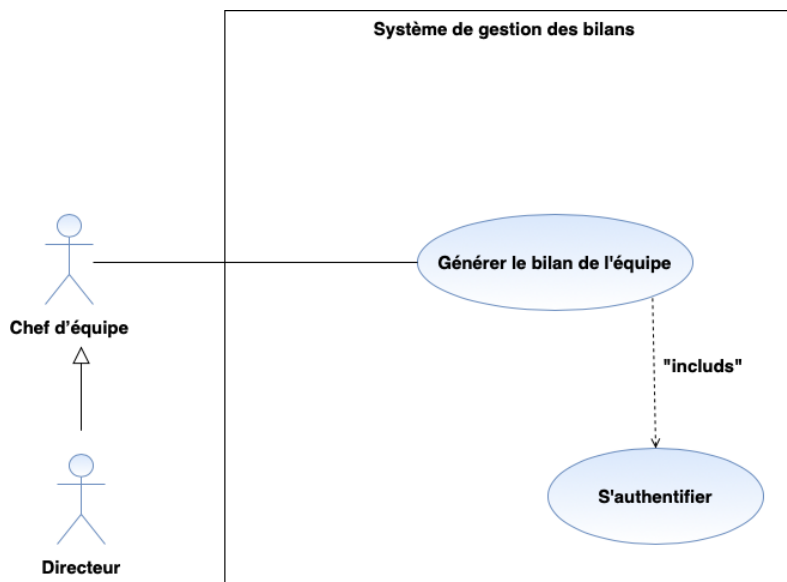


FIGURE 4.16 – Diagramme de cas « Générer le bilan de l’équipe »

2. Description textuelle du cas d’utilisation « Générer le bilan de l’équipe»

La description détaillée de l’user story 5.1 « générer le bilan de l’équipe » est donnée par le tableau suivant

Cas utilisation	Générer le bilan d’équipe
Acteur	Chef d’équipe, Directeur
Précondition	L’acteur doit être authentifié.
Objectif	Bilan d’une équipe de l’unité
Scenario Nominal	<ol style="list-style-type: none"> 1. L’acteur clique sur le menu 2. L’acteur choisir le bouton « Générer Le Bilan » 3. Le système vérifie l’existence des activités pour cette équipe 4. Le système notifie l’acteur 5. Le système remet à l’acteur un bilan
Alternative [A1]	Si aucune activité n’existe pour l’équipe de ce chef, le système affiche un message d’erreur

TABLE 4.10 – Description textuelle du cas d’utilisation « Générer le bilan de l’équipe »

4.3.2 Conception

Dans la phase de conception nous allons décrire les interactions entre les acteurs de l'application et le système. Pour la réalisation des cas d'utilisation décrit dans la phase d'analyse.

Diagrammes de séquence

Un diagramme de séquence est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont [37].

Diagramme de séquence « Ajouter un membre d'équipe »

Pour effectuer des mises à jour de la liste des membres d'équipe, le chef d'équipe doit s'authentifier, ensuite ajouter un membre en remplissant un formulaire d'ajout .

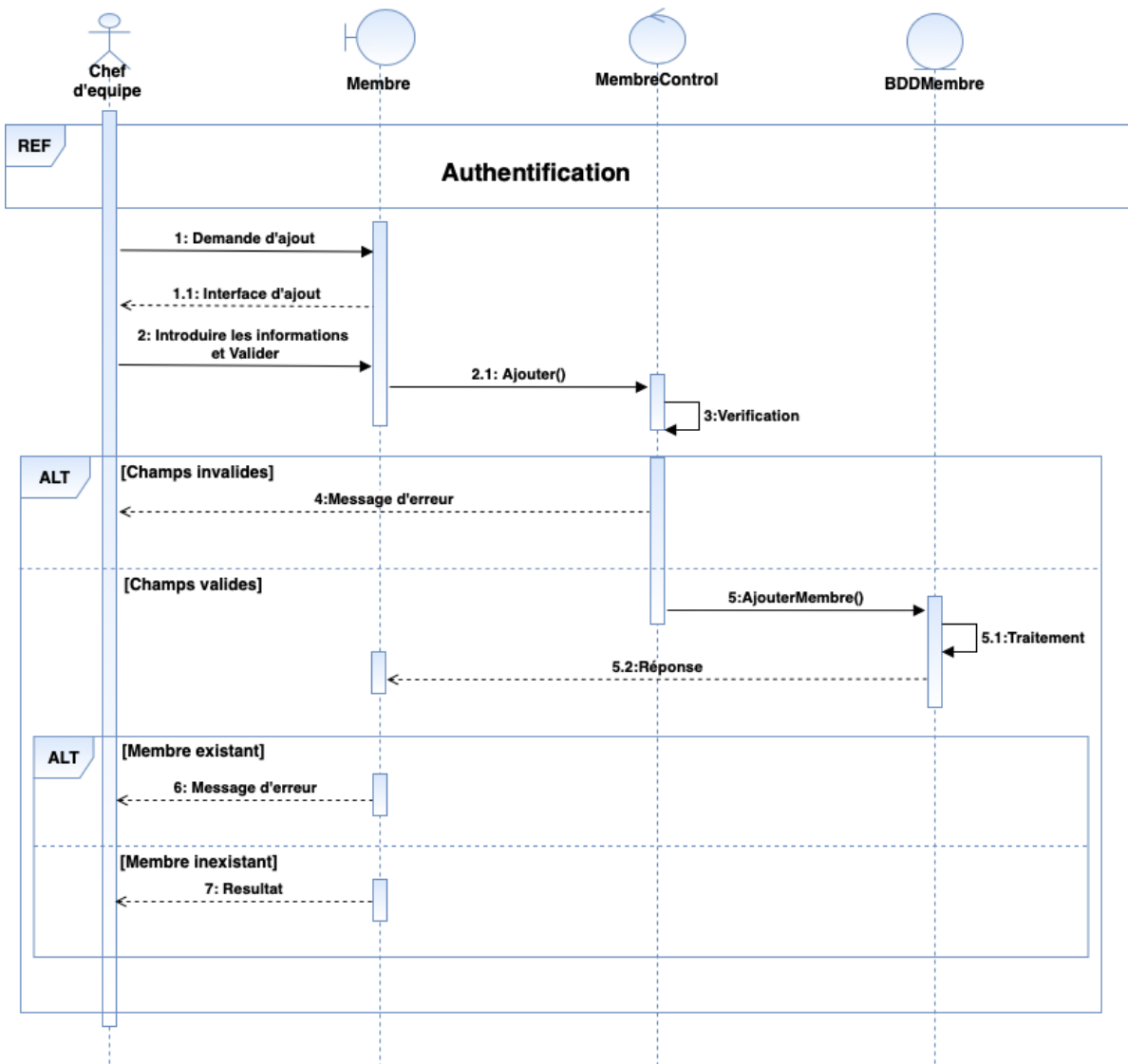


FIGURE 4.17 – Diagramme de séquence « Ajouter un membre d'équipe »

Diagramme de séquence « Générer le bilan de l'équipe »

Pour générer le bilan de tout les membres de la même équipe , le chef d'équipe doit s'authentifier , ensuite générer un bilan selon l'équipe du chef.

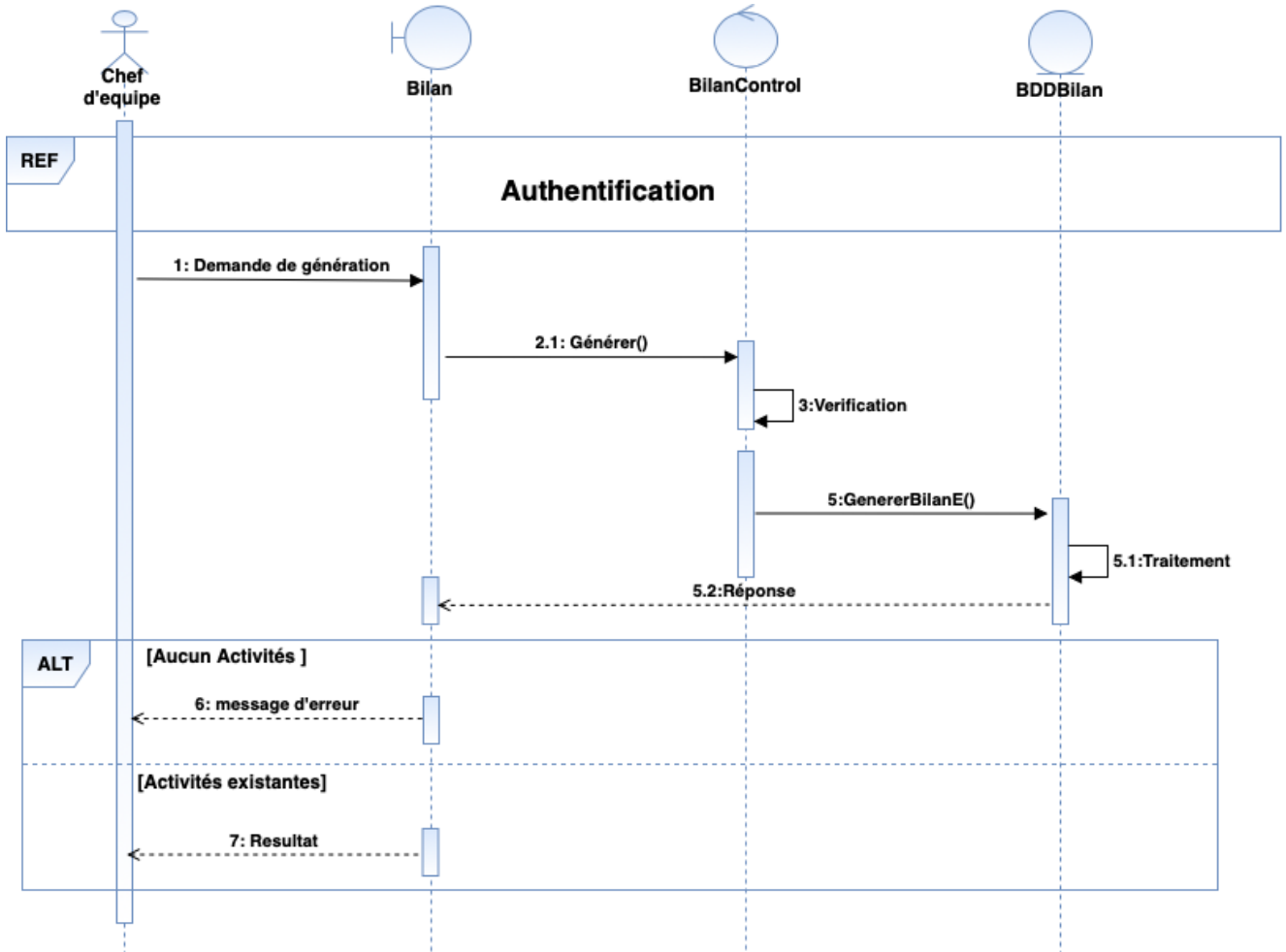


FIGURE 4.18 – Diagramme de séquence « Générer le bilan de l'équipe »

Diagramme de classes

Dans la figure suivante nous présentons les classes associées à ce sprint.

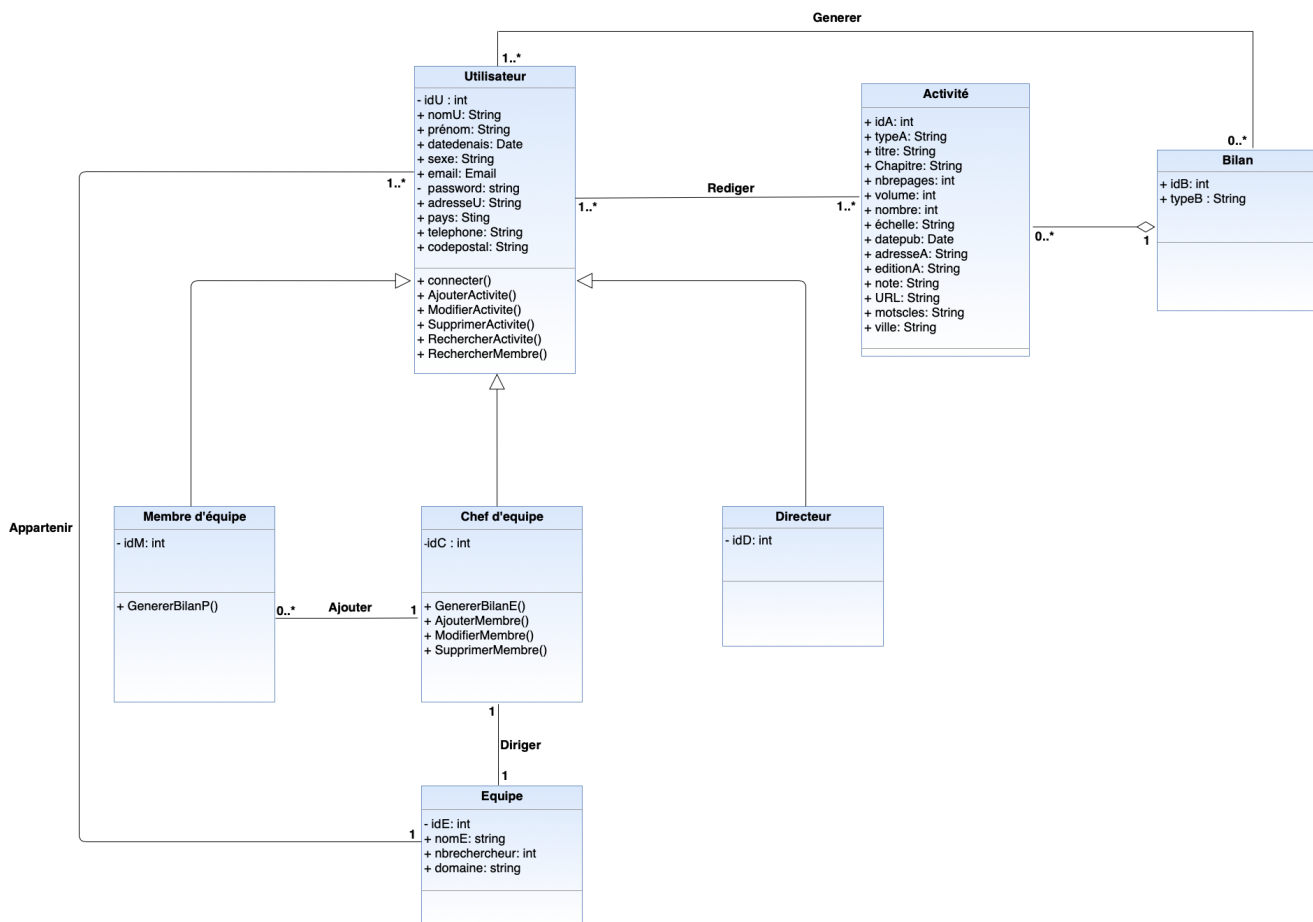


FIGURE 4.19 – Diagramme de classe du sprint 2

1. Schéma Relationnel

La base de données relationnelle est une base de données comprenant des relations dynamiques entre les différents objets contenus dans les tables :

Utilisateur (idU, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays. telephone, codepostal , #idE)

Membre d'équipe (idM, #idU, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays. telephone, #idC)

Chef d'équipe (idC, #dU, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays. telephone, #idE)

Directeur (idD, #idU, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays. telephone)

Activité (idA, typeA, titre, Chapitre, nbrepages, volume, nombre, échelle, datepub, adresseA, editionA, note, URL, ville, #idB)

Rediger #idU, #idA)

Bilan (idB, typeB)

Generer (#idU, #idB)

Equipe (idE, nomE, nbrechercheur, domaine)

4.3.3 Réalisation

Nous allons consacrer cette partie pour la présentation de quelques interfaces élaborées dans ce sprint.

Interface d'ajout d'un membre d'équipe

Sur cette interface le chef d'équipe doit remplir le formulaire d'ajout et cliquer sur valider ou annuler l'ajout.



FIGURE 4.20 – Interface d'ajout d'un utilisateur

Interface Bilan de l'équipe

Cette interface illustre les différentes activités ajoutées par les membres d'une même équipe qui forme le bilan de l'équipe.

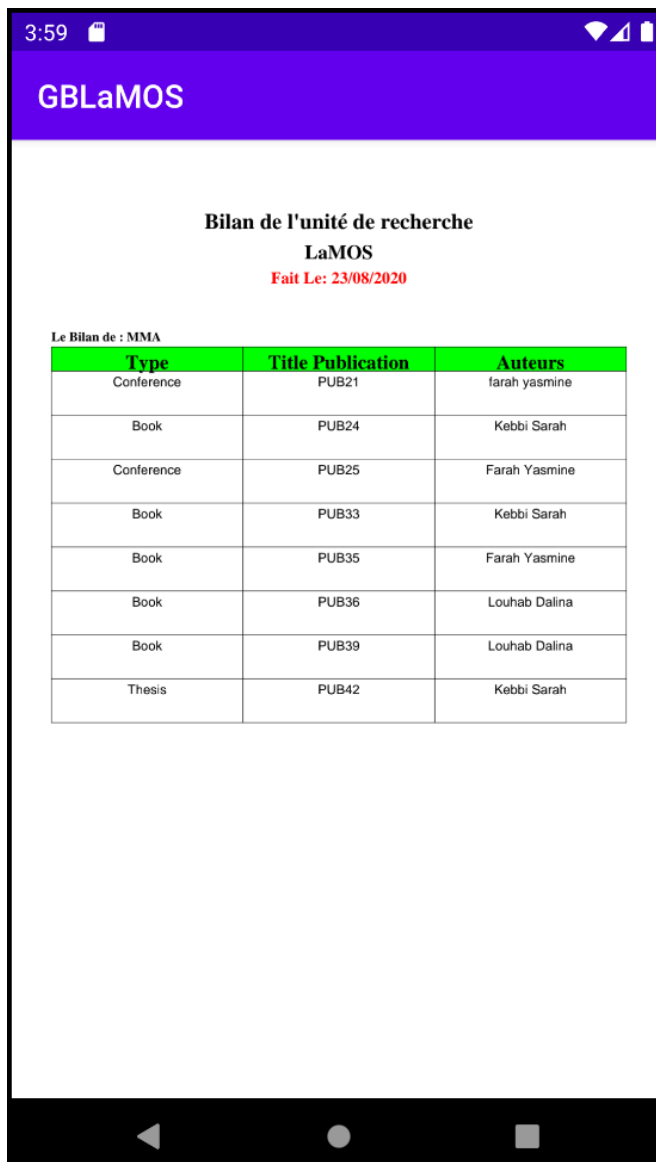


FIGURE 4.21 – Interface du bilan d'équipe

4.4 Conclusion

Dans ce chapitre, nous avons réussi à produire un incrément de valeur pour le client et utile dans un environnement de production. Et nous avons effectué la conception et la réalisation du module de sprint 1 et sprint 2.

La release a été présentée dans le cadre d'une réunion de fin de release et préparation au prochaine.

Dans le chapitre qui suit, nous allons produire une nouvelle release couvrant les fonctionnalités suivantes : Gestion des équipes , des activités visiteur , des sorties pédagogiques et génération du bilan général de l'unité.

Chapitre 5

Release 2

5.1 Introduction

Dans ce chapitre, nous allons détailler le travail réalisé durant la deuxième release qui est décomposée en 2 sprints qui sont :

- Sprint 1 « gestion des équipes et génération du bilan de l'unité »
- Sprint 2 « gestion des sorties pédagogiques, gestion des activités visiteur »

Le développement de chaque sprint passe par les étapes d'analyse, de conception et de réalisation.

5.2 Développement du sprint 1 « Gestion des équipes et génération du bilan de l'unité »

Ce premier sprint s'étale sur 26 jours et se décompose en deux items :

- Gérer les équipes
- Générer le bilan de l'unité

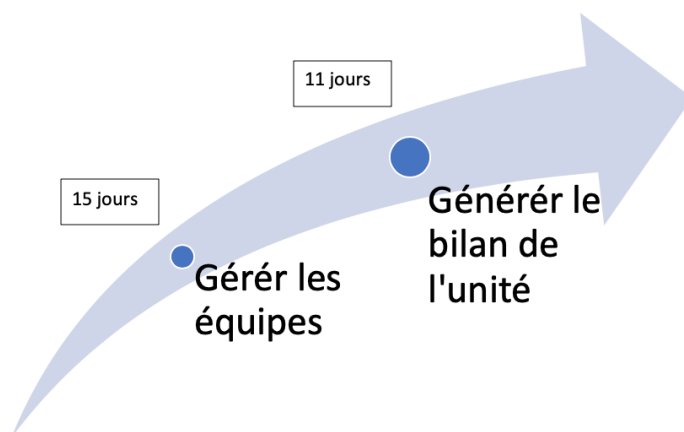


FIGURE 5.1 – Décomposition du sprint 1 en items

Une user story est une demande fonctionnelle basée sur l'un ou les utilisateurs clés du produit qui va rajouter de la valeur business au produit. Elle sera écrite dans un langage naturel compris par l'ensemble des acteurs du projet ou liés à celui-ci. Dans le tableau qui suit nous présentons les user stories du sprint 1 :

ID U. S	User Story
6.1	En tant que directeur, je peux ajouter une équipe
6.2	En tant que directeur, je peux modifier une équipe
6.3	En tant que directeur, je peux supprimer une équipe
6.4	En tant que directeur, je peux chercher une équipe
7.1	En tant que dicteur, Je génère le bilan de l'unité

TABLE 5.1 – Le sprint planning de sprint 1

Le backlog du sprint 1 est le suivant :

Item	User Story	Description	Priorité
Gérer les équipes et générer le bilan de l'unité	Ajouter une équipe	En tant que directeur, je peux ajouter une équipe	6
	Modifier une équipe	En tant que directeur, je peux Modifier une équipe	
	Supprimer une équipe	En tant que directeur, je peux supprimer une équipe	
	Chercher une équipe	En tant que directeur, je cherche une équipe	
Générer le bilan de l'unité	Générer le bilan de l'unité	En tant que directeur, Je génère le bilan de l'unité	7

TABLE 5.2 – Backlog du sprint 1

5.2.1 Analyse

Dans cette première phase nous allons présenter les diagrammes de cas d'utilisation ainsi que leurs descriptions.

Diagramme de cas d'utilisation

La figure suivante décrit de façon globale les cas d'utilisation identifiés pour le sprint 1

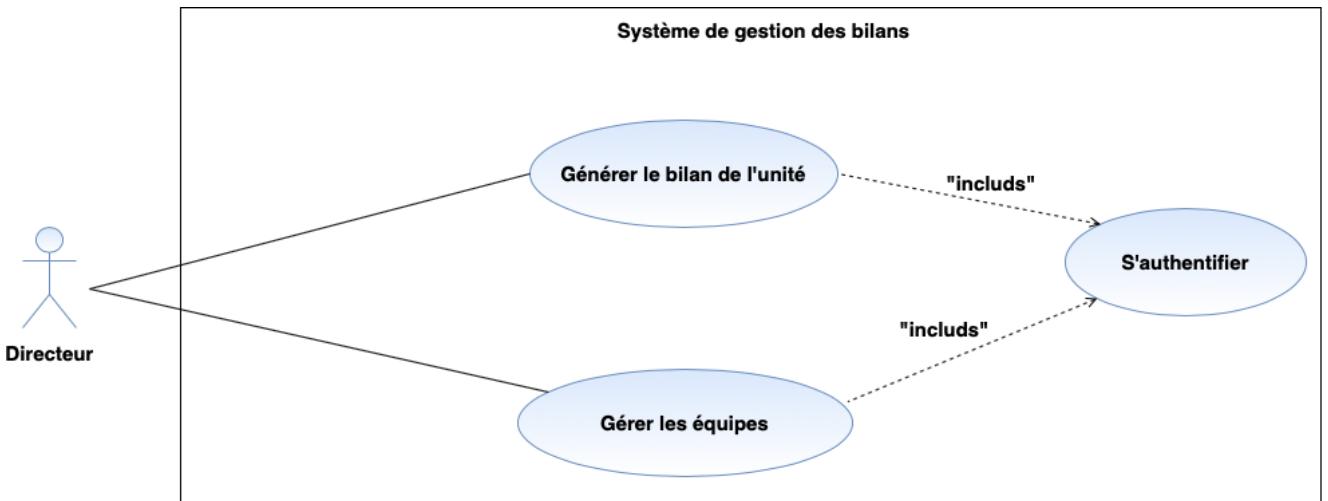


FIGURE 5.2 – Diagramme de cas d'utilisation du sprint 1

Description de l’item « Gérer les équipes »

Dans ce qui suit nous allons décrire le premier item du sprint 1 qui y’a duré 15 jours.

1. Diagramme de cas d'utilisation « Gérer les équipes »

La figure ci-dessous représente le diagramme des cas d'utilisation ” Gérer les équipes”.

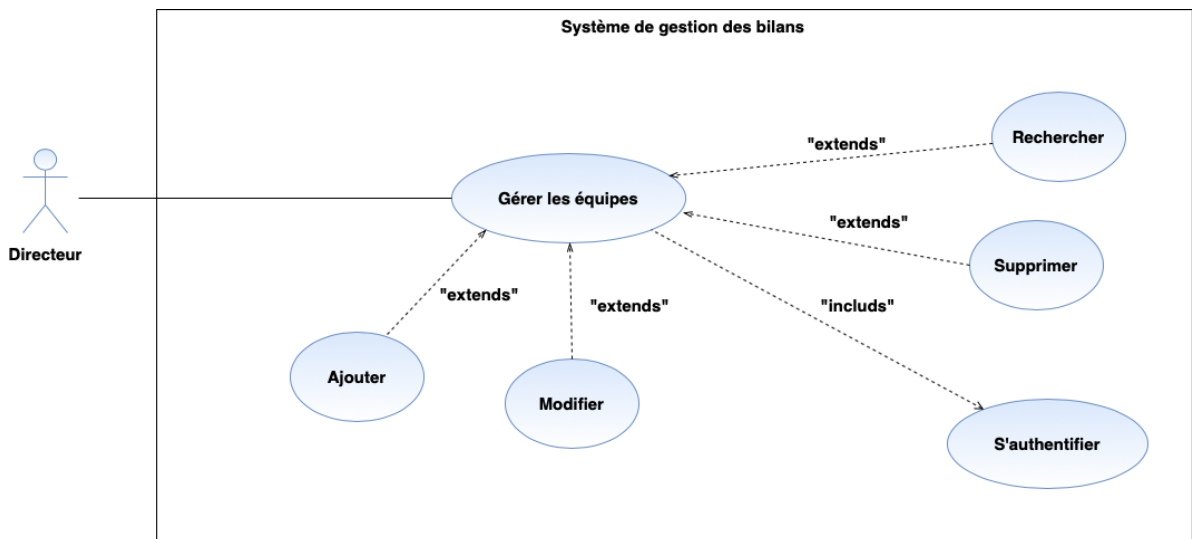


FIGURE 5.3 – Diagramme de cas d'utilisation « Gérer les équipes »

2. Description textuelle du cas d'utilisation « Ajouter une équipe »

La description détaillée de l'user story 6.1 « Ajouter une équipe » est donnée par le tableau suivant :

Cas utilisation	Ajout une activité
Acteur	Directeur
Précondition	L'acteur doit être authentifié.
Objectif	Mise à jour de la liste des équipes
Scenario Nominal	<ol style="list-style-type: none"> 1. L'acteur clique sur le menu 2. L'acteur clique sur « Equipe » 3. L'acteur choisir le bouton « Ajouter équipe » 4. Le système afficher l'interface d'ajout 5. L'acteur remplit les champs 6. Le système vérifier les données saisies par l'acteur 7. Le système enregistre les informations 8. Le système notifie l'acteur
Alternative [A1]	Si un champ d'information n'est pas valide ou l'équipe existe déjà, le système affiche un message d'erreur

TABLE 5.3 – Description textuelle du cas d'utilisation « Ajouter une équipe »

Description de l'item « Générer le bilan de l'unité »

Dans ce qui suit nous allons décrire le second item du sprint 1 qui y'a duré 11 jours.

1. Diagramme de cas d'utilisation « Générer le bilan de l'unité »

La figure ci-dessous représente le diagramme des cas d'utilisation "Générer le bilan de l'unité".

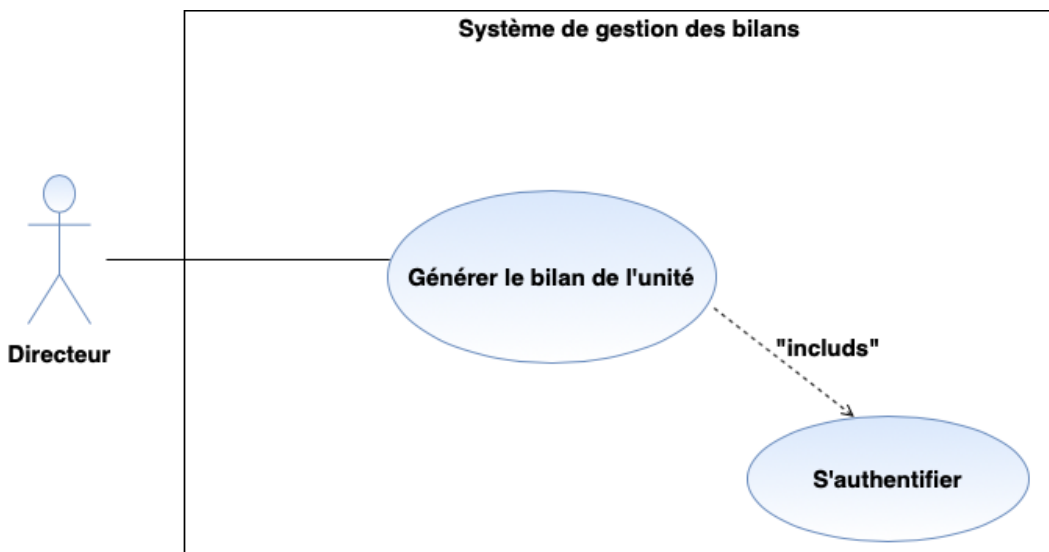


FIGURE 5.4 – Diagramme de cas d'utilisation « Générer le bilan de l'unité »

2. Description textuelle du cas d'utilisation « Générer le bilan de l'unité »

La description détaillée de l'user story 7.1 « Générer le bilan de l'unité » est donnée par le tableau suivant :

Cas utilisation	Générer le bilan de l'unité
Acteur	Directeur
Précondition	L'acteur doit être authentifié.
Objectif	Bilan de l'unité
Scenario Nominal	<ol style="list-style-type: none"> 1. L'acteur clique sur le menu 2. L'acteur choisit le bouton « Générer bilan » 3. L'acteur remplit la période de temps 4. Le système vérifie les données saisies par l'acteur 5. Le système notifie l'acteur 6. Le système remet à l'acteur un bilan selon la période saisie
Alternative [A1]	Si aucune activité n'existe dans cette période, le système affiche un message d'erreur

TABLE 5.4 – Description textuelle du cas d'utilisation « Générer le bilan de l'unité »

5.2.2 Conception

Dans cette deuxième phase nous allons décrire les interactions entre les acteurs de l'application et le système afin de réaliser les cas d'utilisation décrits dans la phase d'analyse.

Diagrammes de séquence

Un diagramme de séquence est un diagramme d'interaction dont le but est de décrire comment les objets collaborent au cours du temps et quelles responsabilités ils assument. Il décrit un scénario d'un cas d'utilisation [38].

Diagramme de séquence « Ajouter une équipe »

Pour effectuer des mises à jour de la liste des équipes, le directeur doit s'authentifier, ensuite ajouter une équipe en remplissant un formulaire d'ajout.

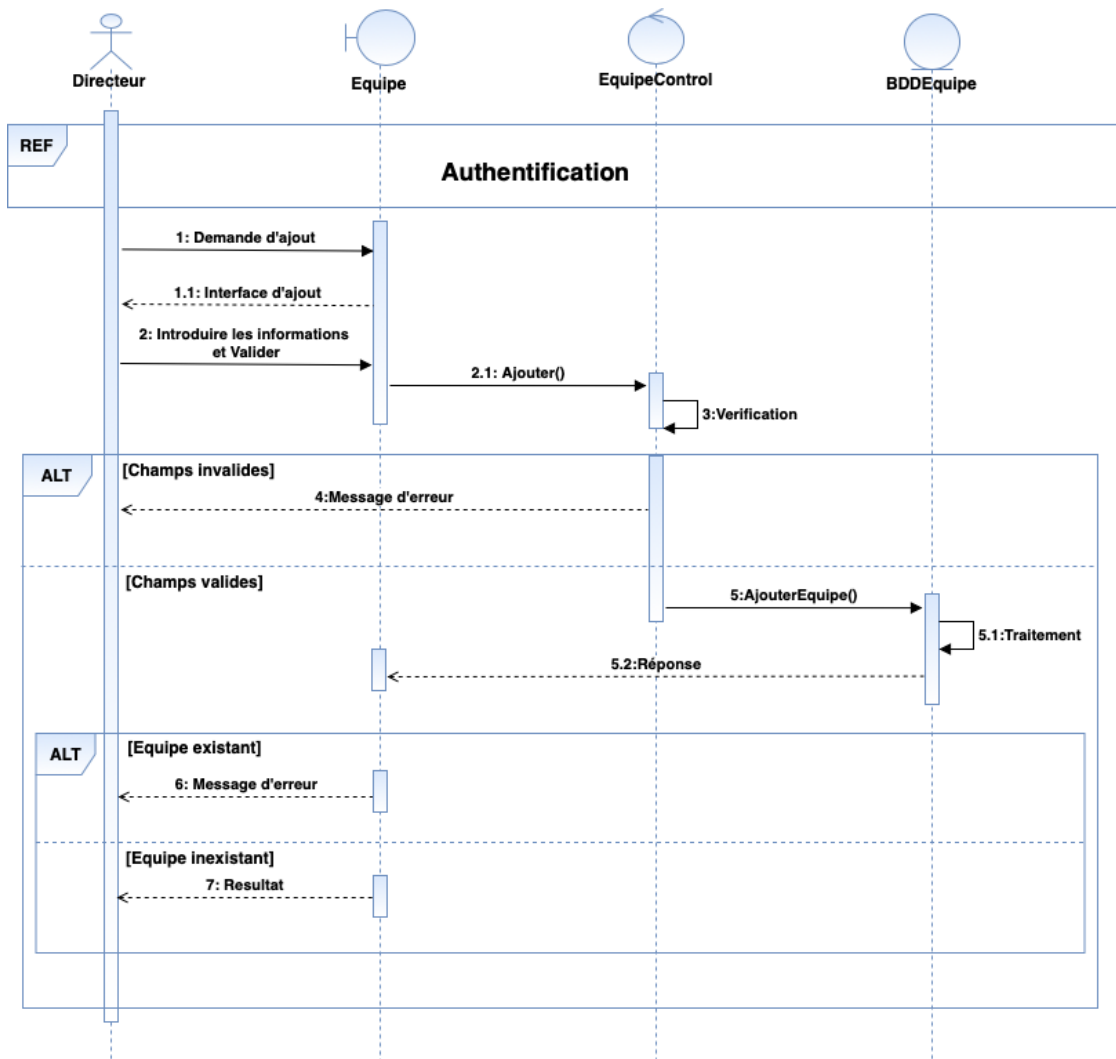


FIGURE 5.5 – Diagramme de séquence « Ajouter une équipe »

Diagramme de séquence « Générer le bilan de l'unité »

Pour générer le bilan de l'unité, le directeur doit s'authentifier, ensuite générer un bilan en remplissant quelques critères sur le bilan.

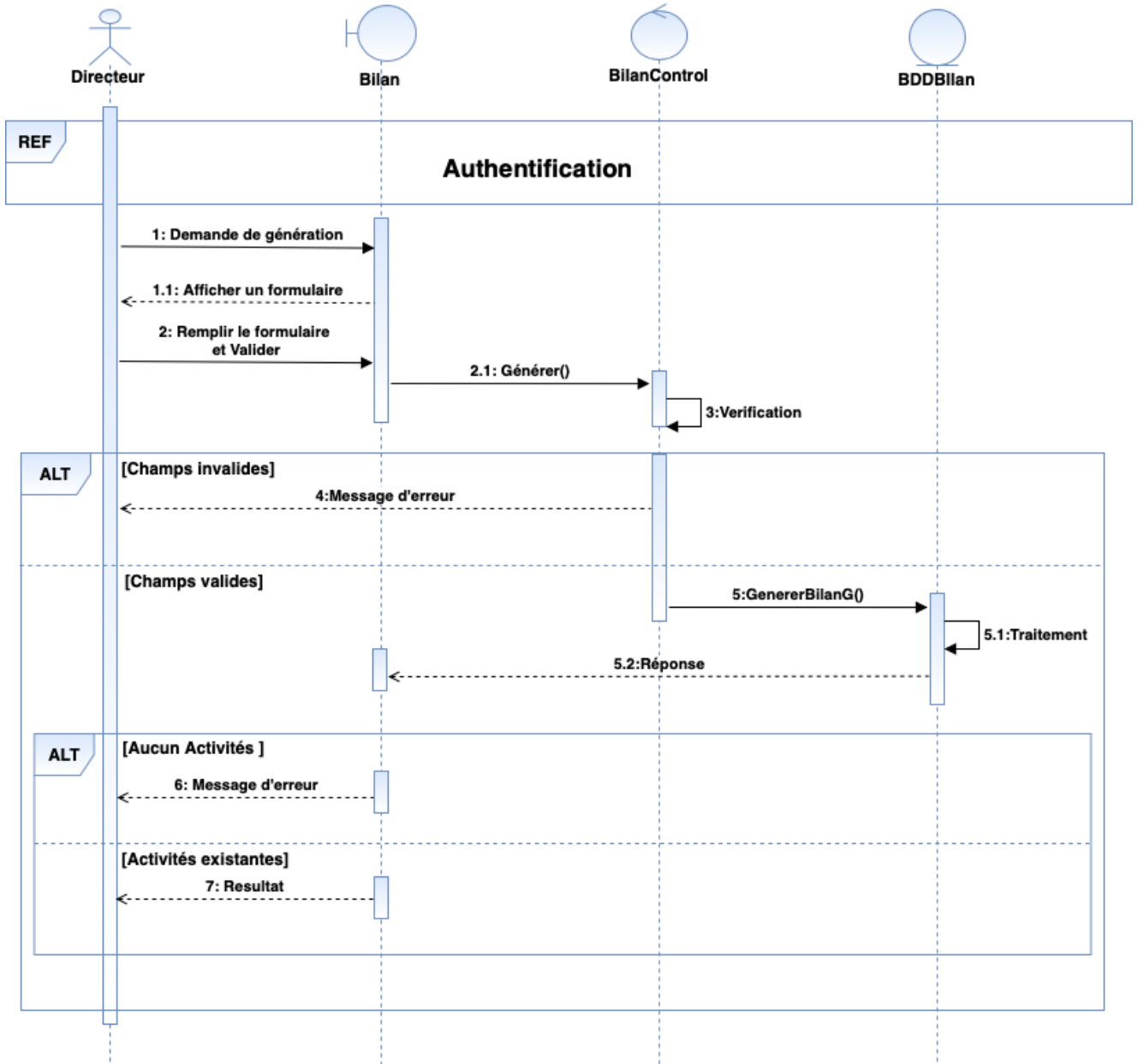


FIGURE 5.6 – Diagramme de séquence « Générer le bilan de l'unité »

Diagramme de classes

Dans la figure suivante nous présentons les classes associées à ce sprint.

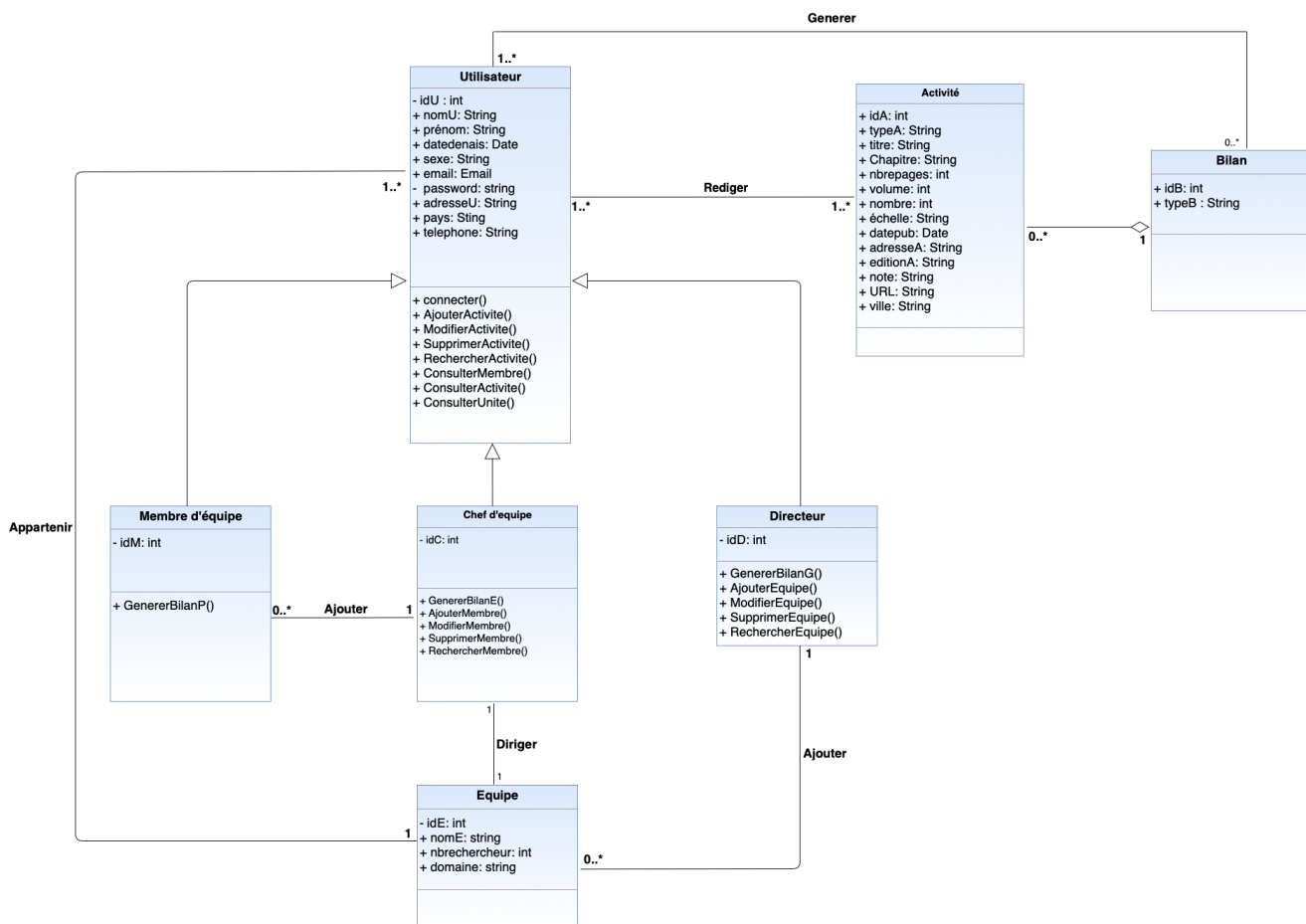


FIGURE 5.7 – Diagramme de classes du sprint 1

1. Schéma Relationnel

La base de données relationnelle est une base de données comprenant des relations dynamiques entre les différents objets contenus dans les tables :

Utilisateur (idU, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays, telephone, #idE)

Membre d'équipe (idM, #idU, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays, telephone, #idC)

Chef d'équipe (idC, #idU, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays, telephone, #idE)

Directeur (idD, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays, telephone)

Activité (idA, typeA, titre, Chapitre, nbrepages, volume, nombre, échelle, datepub, adresseA, editionA, note, URL, ville, #idB)

Rediger (#idU, #idA)

Bilan (idB, typeB)

Generer (#idU, #idB)

Equipe (idE, nomE, nbrechercheur, domaine, #idD)

5.2.3 Réalisation

Dans cette phase nous allons décrire les interfaces représentant le travail élaboré dans ce sprint.

Interface d'ajout d'une équipe

Dans cette interface le directeur peut ajouter une nouvelle équipe en remplissant un formulaire d'ajout.



FIGURE 5.8 – Interface d'ajout d'une équipe

Interface du bilan de l'unité

Cette interface montre le formulaire à remplir (date de début, date de fin) afin de générer le bilan général selon la période saisie.

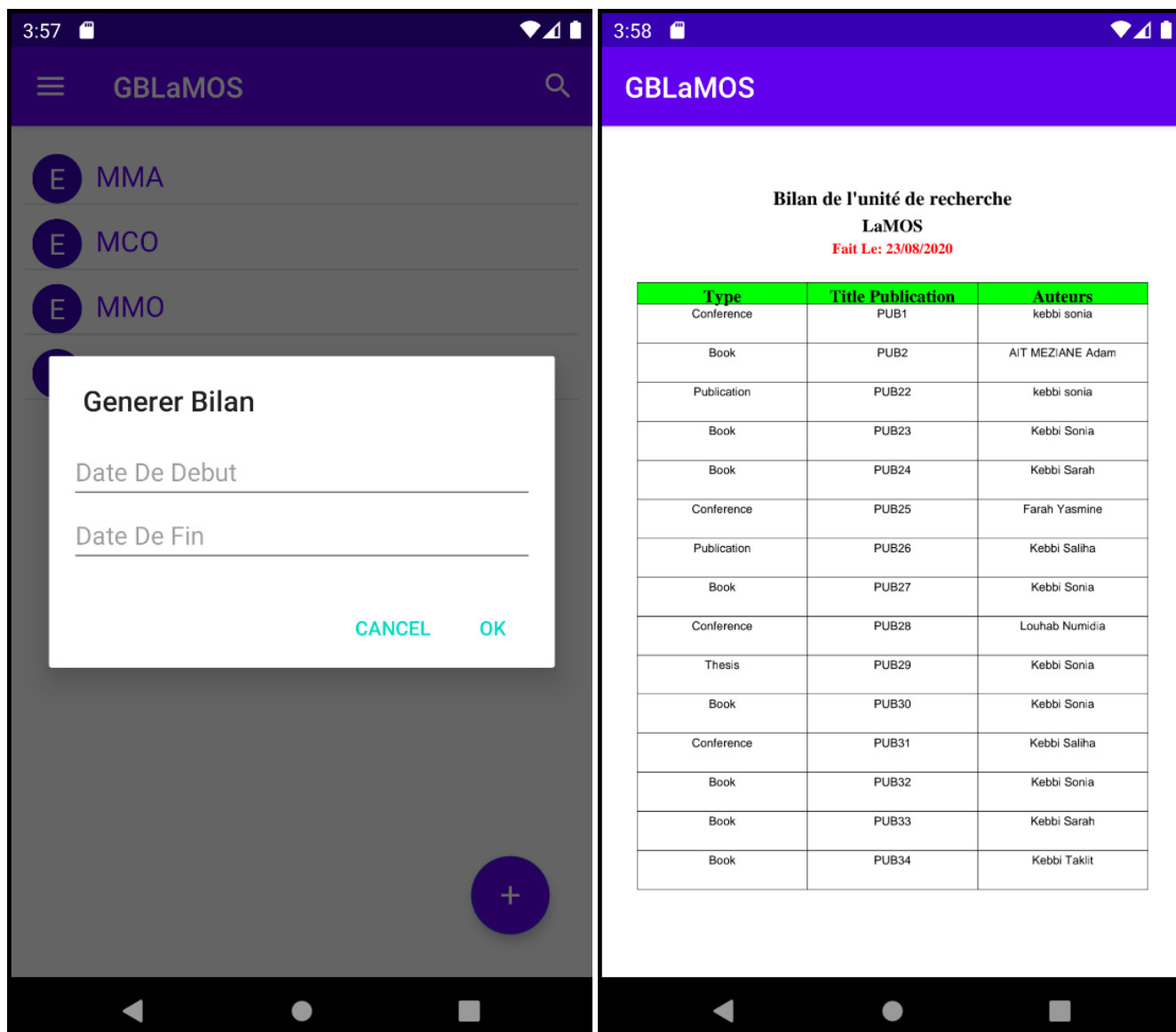


FIGURE 5.9 – Interfaces de génération du bilan de l'unité

5.3 Développement du sprint 2 « gestion des sorties pédagogiques et la gestion des activités visiteur »

Ce premier sprint s'étale sur 24 jours et se décompose en deux items

- Gérer les sorties pédagogiques
- Gérer les activités visiteur

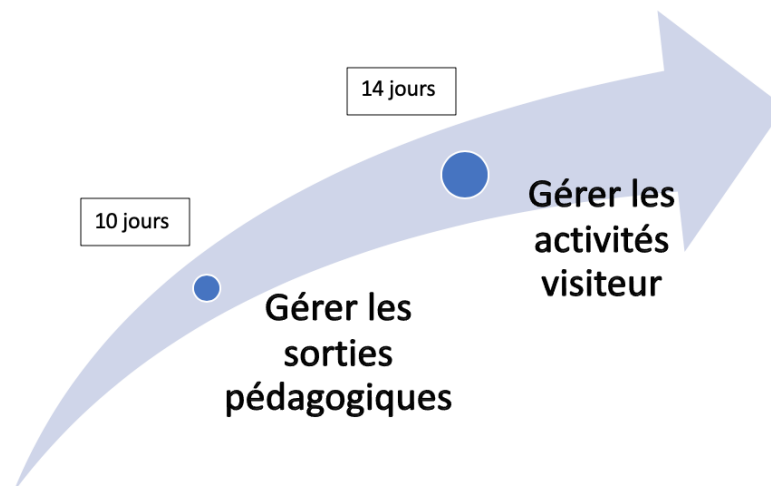


FIGURE 5.10 – Décomposition du sprint 2 en items

Dans le tableau qui suit on présente les user stories du sprint 2 :

ID U. S	User Story
8.1	En tant que directeur, je peux ajouter une sortie pédagogique
8.2	En tant que directeur, je peux modifier une sortie pédagogique
8.3	En tant que directeur, je peux supprimer une sortie pédagogique
8.4	En tant que directeur, je peux chercher une sortie pédagogique
9.1	En tant que visiteur, je peux consulter les activités
9.2	En tant que visiteur, je peux consulter le profil d'un membre de l'unité
9.4	En tant que visiteur, je peux consulter les équipes de l'unité
9.5	En tant que visiteur, je peux consulter les sorties pédagogiques

TABLE 5.5 – Le sprint planning du sprint 2

Le backlog du sprint 2 est le suivant :

Item	User Story	Description	Priorité
Gérer les sorties pédagogiques	Ajouter une sortie	En tant que directeur, je peux ajouter une sortie	8
	Modifier une sortie	En tant que directeur, je peux Modifier une sortie	
	Supprimer une sortie	En tant que directeur, je peux supprimer une sortie	
	Chercher une sortie	En tant que directeur, je peux chercher une sortie	
Gérer les activités visiteur	Consulter les activités	En tant qu'utilisateur, je peux consulter les activités	9
	Consulter le profil d'un membre de l'unité	En tant qu'utilisateur, je peux consulter le profil d'un membre de l'unité	
	Consulter les équipes de l'unité	En tant qu'utilisateur, je peux consulter les équipes de l'unité	

TABLE 5.6 – Backlog de sprint 2

5.3.1 Analyse

Dans cette première phase nous allons présenter les diagrammes de cas d'utilisation ainsi que leurs descriptions.

Diagramme de cas d'utilisation

La figure suivante décrit de façon globale les cas d'utilisation identifiés pour le sprint 2.

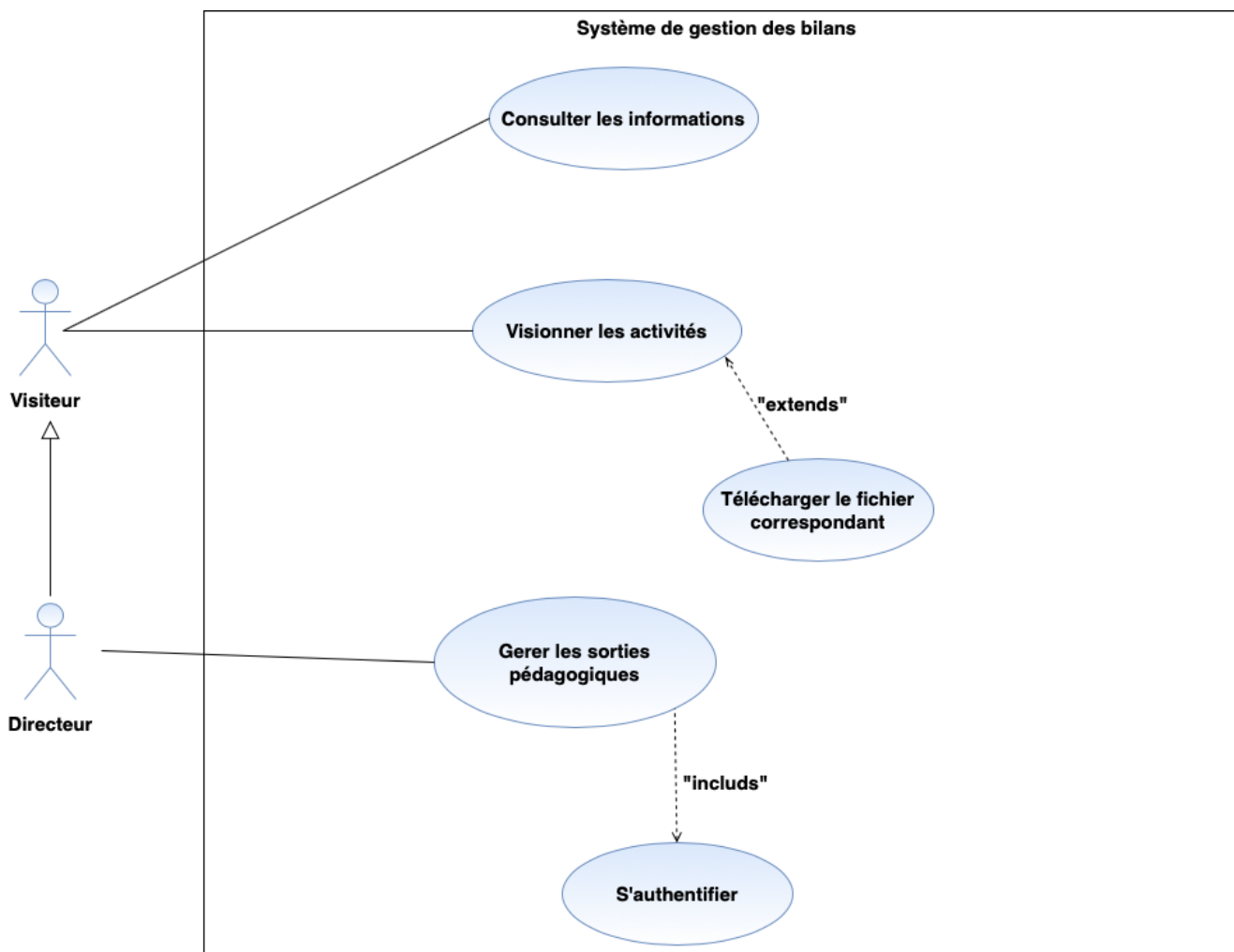


FIGURE 5.11 – Diagramme de cas d'utilisation du sprint 2

Description de l'item « Gérer les sorties pédagogiques »

Dans ce qui suit nous allons décrire le premier item du sprint 2 qui y'a duré 10 jours.

1. Diagramme de cas d'utilisation « Gérer les sorties pédagogiques »

La figure ci-dessous représente le diagramme des cas d'utilisation "Gérer les sorties pédagogiques".

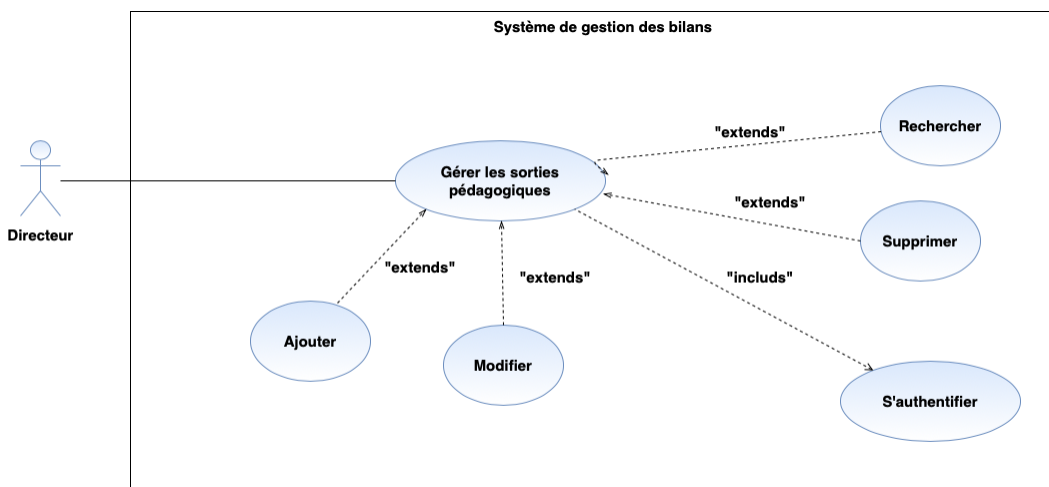


FIGURE 5.12 – Diagramme de cas « Gérer les sorties pédagogiques »

2. Description textuelle du cas d'utilisation « Supprimer une sortie pédagogique »

La description détaillée de l’user story 8.3 « Supprimer une sortie pédagogique » est donnée par le tableau suivant :

Cas utilisation	Supprimer la sortie pédagogique
Acteur	Directeur
Précondition	L’acteur doit être authentifié.
Objectif	Mise à jour de la liste des membres d’équipe
Scénario Nominal	<ol style="list-style-type: none"> 1. L’acteur clique sur le menu 2. L’acteur choisie le bouton « Sortie » 3. L’acteur clique sur la sortie un menu apparait 4. L’acteur choisie le bouton « Supprimer une sortie » 5. Le système supprime la sortie de la base de données 6. Le système notifie l’acteur

TABLE 5.7 – Description textuelle du cas d'utilisation « Supprimer une sortie pédagogique »

Description de l’item « Gérer les activités visiteur »

Dans ce qui suit nous allons décrire le deuxième item du sprint 2 qui y’a duré 14 jours.

1. Diagramme de cas d’utilisation « Gérer les activités visiteur »

La figure ci-dessous représente le diagramme des cas d’utilisation ”Gérer les activités visiteur”.

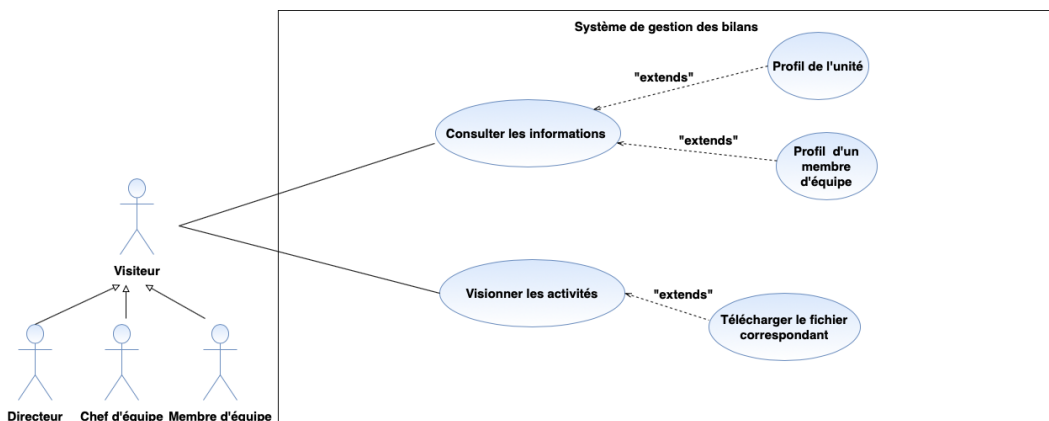


FIGURE 5.13 – Diagramme de cas « Gérer les activités visiteur »

2. Description textuelle du cas d’utilisation « Consulter le profil d’un membre d’équipe »

La description détaillée de l’user story 9.2 « Consulter profil d’un membre d’équipe » est donnée par le tableau suivant :

Cas utilisation	Consulter profil d’un membre d’équipe
Acteur	Visiteur
Objectif	Consultation des informations d’un membre d’équipe
Scenario Nominal	<ol style="list-style-type: none"> 1. L’acteur rentre dans l’application 2. L’acteur choisie une activité 3. L’acteur choisie un auteur 4. Le système affiche le profil de l’auteur choisi

TABLE 5.8 – Description textuelle du cas d’utilisation « Consulter le profil d’un membre d’équipe »

5.3.2 Conception

Dans cette deuxième phase nous allons décrire les interactions entre les acteurs de l'application et le système afin de réaliser les cas d'utilisation décrit dans la phase d'analyse.

Diagrammes de séquence

Un diagramme de séquence est un diagramme d'interaction dont le but est de décrire comment les objets collaborent au cours du temps et quelles responsabilités ils assument. Il décrit un scénario d'un cas d'utilisation [38].

Diagramme de séquence « Supprimer une sortie pédagogique »

Pour effectuer des mises à jour de la liste des sorties pédagogiques, le directeur doit s'authentifier, ensuite supprimer une sortie choisie.

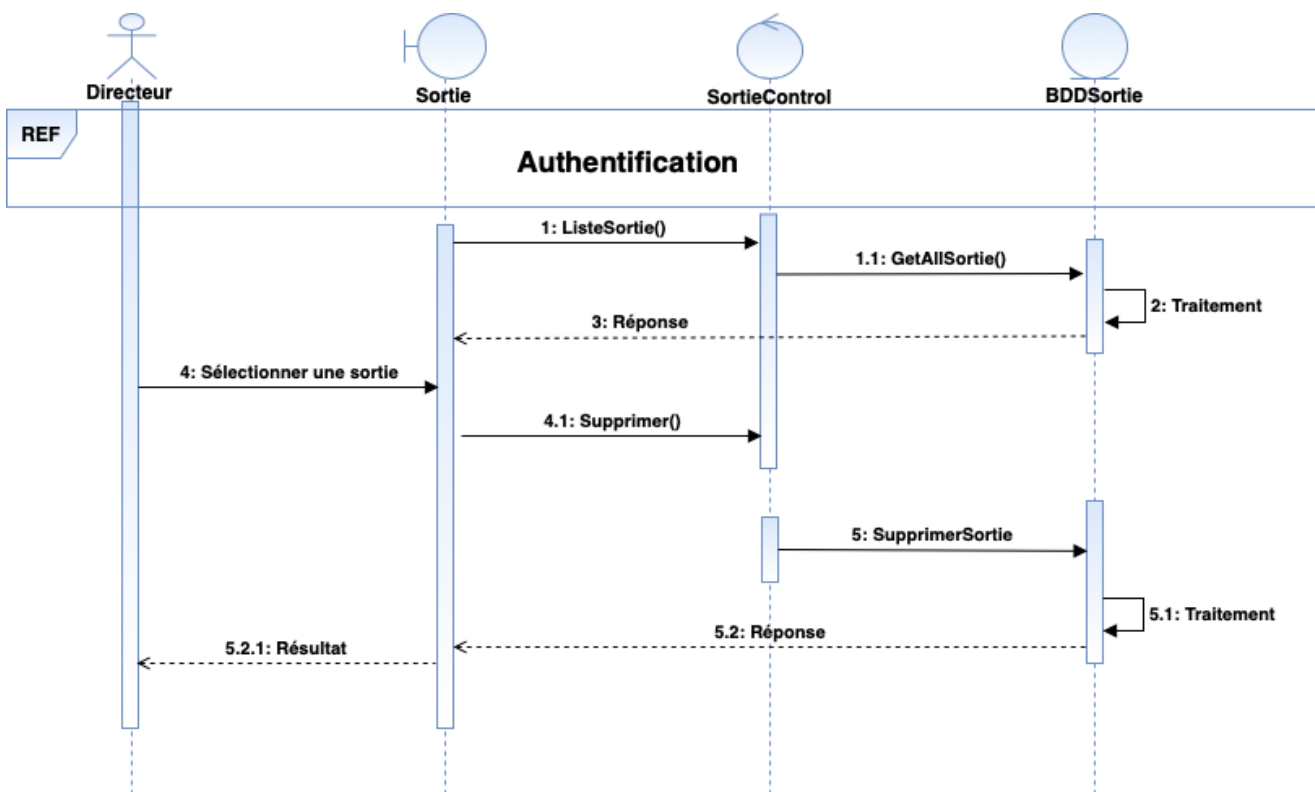


FIGURE 5.14 – Diagramme de séquence « Supprimer une sortie pédagogique »

Diagramme de séquence « Consulter profil d'un membre d'équipe »

Dans ce cas d'utilisation, le visiteur doit d'abord accéder à l'application , pour pouvoir consulter les activités ajoutées par les membres de cette unité.

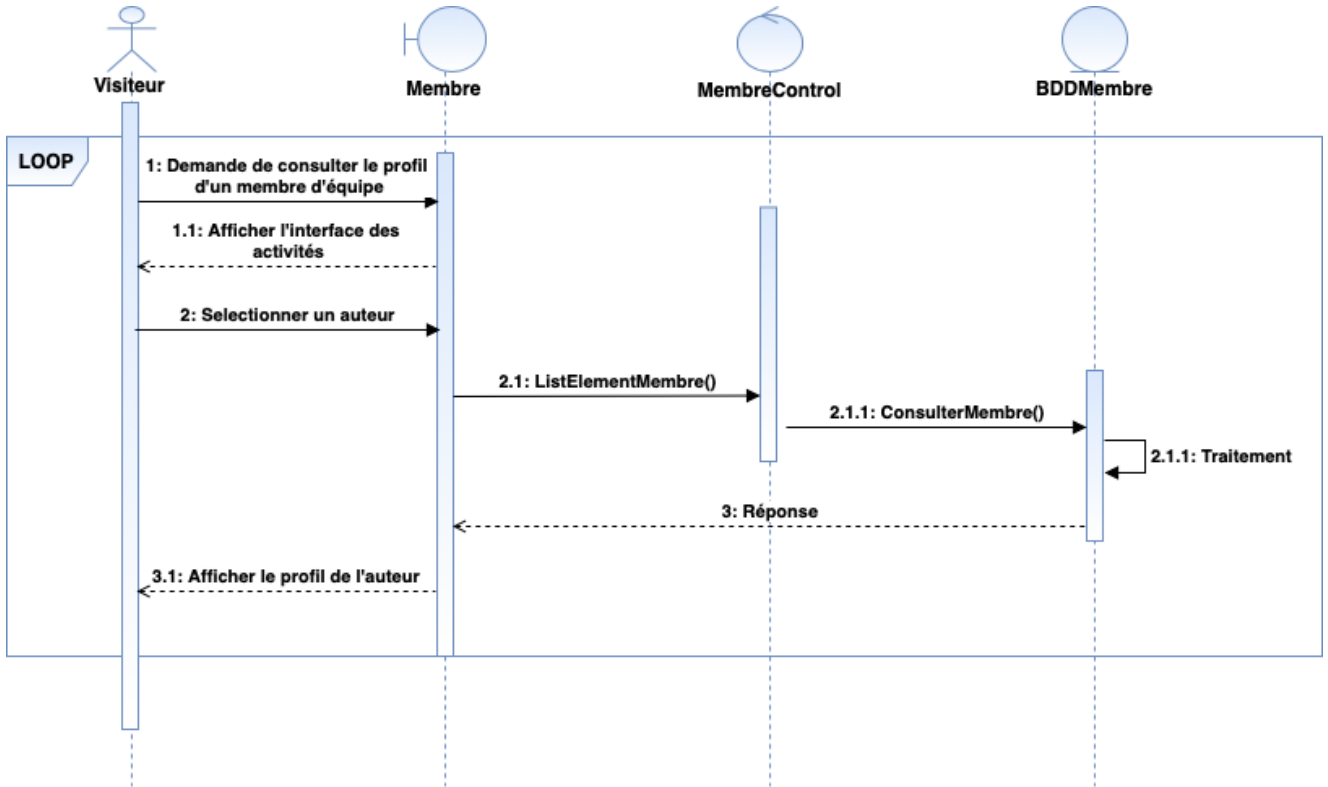


FIGURE 5.15 – Diagramme de séquence « Consulter le profil d'un membre d'équipe »

Diagramme de classes

Dans la figure suivante nous présentons les classes associées à ce sprint.

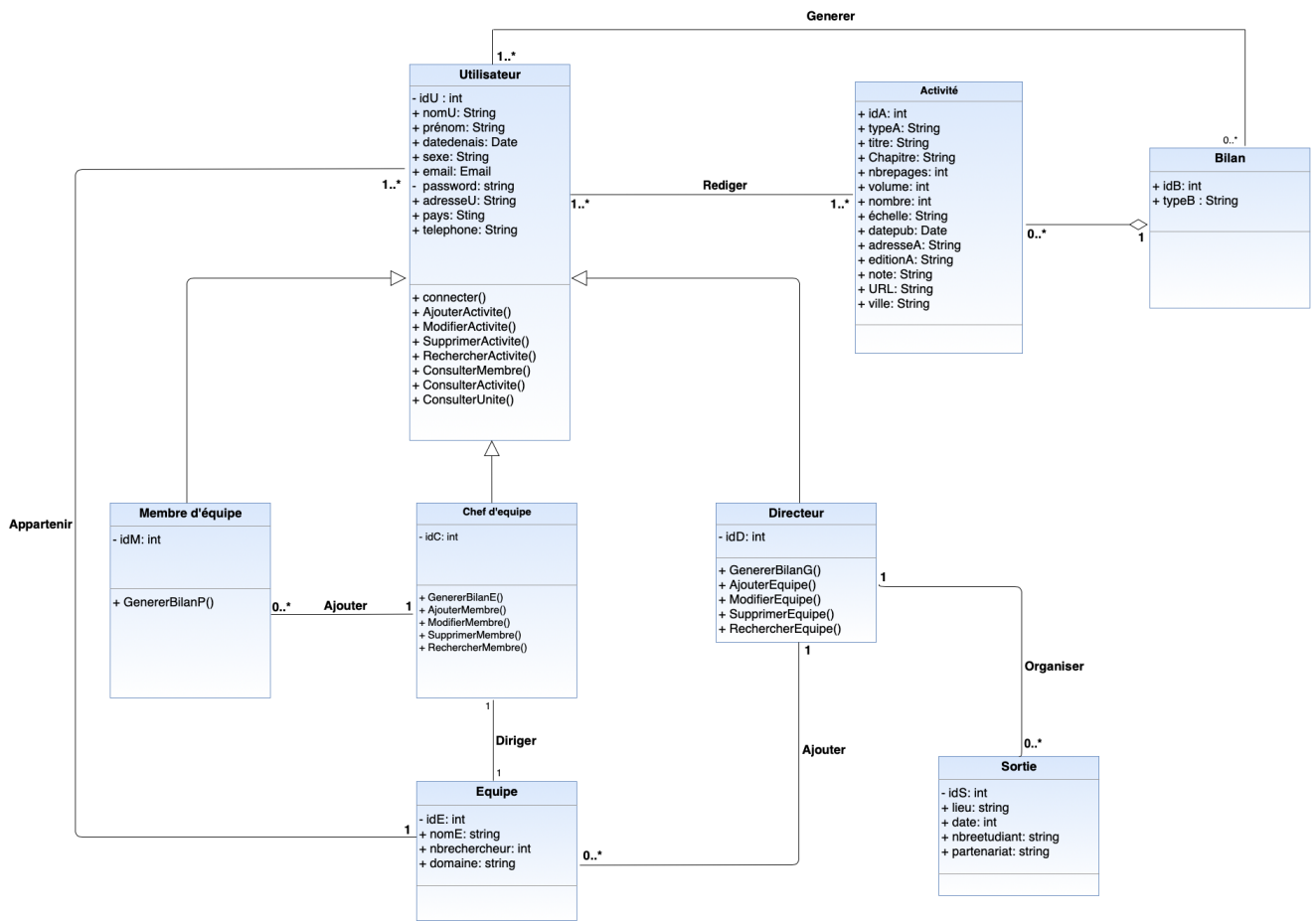


FIGURE 5.16 – Diagramme de classe du sprint 2

1. Schéma Relationnel

La base de données relationnelle est une base de données comprenant des relations dynamiques entre les différents objets contenus dans les tables :

Utilisateur (idU, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays, telephone, #idE)

Membre d'équipe (idM, #idU, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays, telephone, #idC)

Chef d'équipe (idC, #idU, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays, telephone, #idE)

Directeur (idD, nomU, prenomU, datedenais, sexe, email, password, adresseU, pays, telephone, #idU)

Activité (idA, typeA, titre, chapitre, nbrepages, volume, nombre, échelle, datepub, adresseA, editionA, note, URL, ville, #idB)

Rediger (#idU, #idA)

Bilan (idB, typeB)

Generer (#idU, #idB)

Equipe (idE, nomE, nbrechercheur, domaine, #idD)

Sortie (idS, lieu, date, nbreetudiant, partenariat, #idD)

5.3.3 Réalisation

Nous présentons dans ce qui suit quelques interfaces du travail réalisé dans ce sprint.

Interface de présentation

Cette interface illustre un aperçu sur la présentation de l'unité de recherche LaMOS.

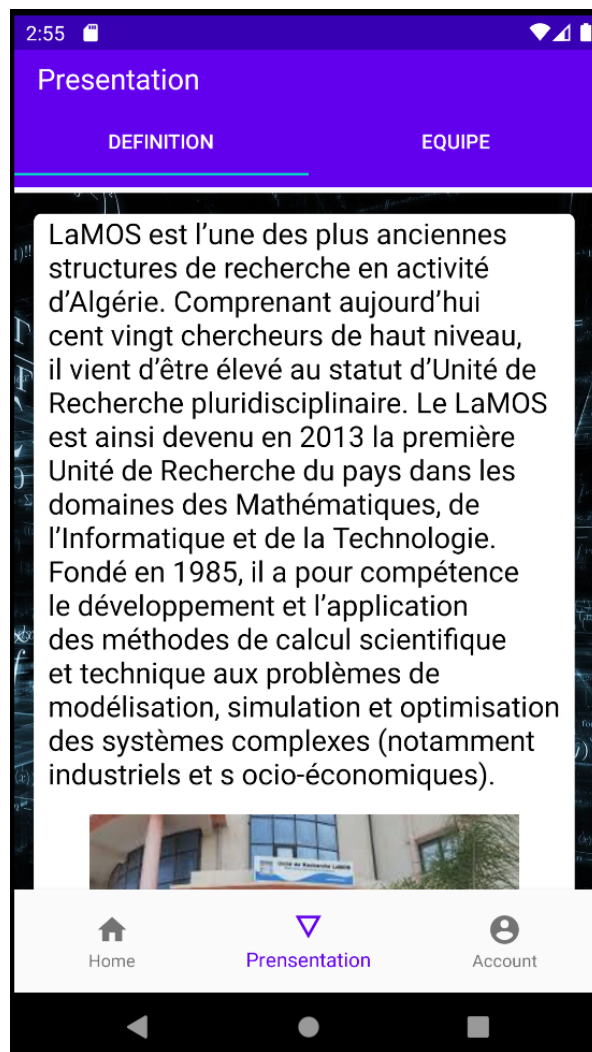


FIGURE 5.17 – Interface de présentation

Interface d'accueil

Cette interface illustre les activités ajoutées par les membres de l'unité.

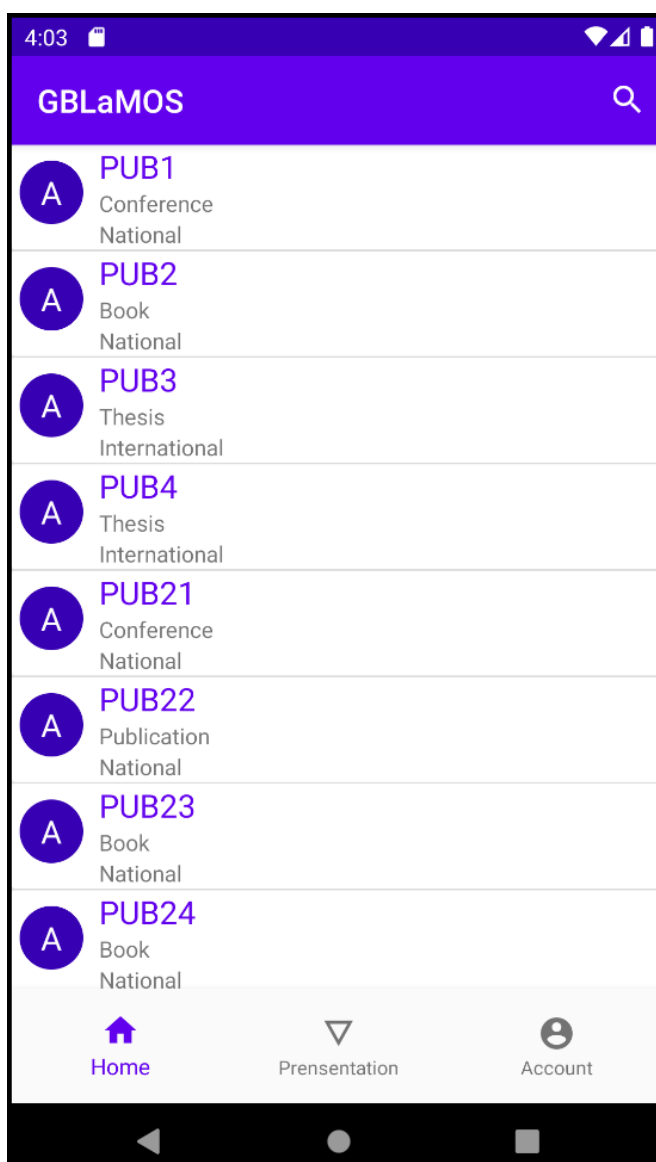


FIGURE 5.18 – Interface d'accueil

Interface d'ajout d'une sortie

Cette interface permet à un directeur d'ajouter une sortie en remplissant ce formulaire.



FIGURE 5.19 – Interface d'ajout d'une sortie

5.4 Conclusion

À la fin de ce chapitre, nous avons réussi à produire un incrément répondant aux besoins du client et étant utilisé dans un environnement de production. Nous clôturons ce rapport par une conclusion générale.

Conclusion Générale et Perspectives

Le développement de notre projet intitulé « Conception et réalisation d'une application mobile de gestion des Bilans » nous a permis de répondre aux besoins des chercheurs de l'unité de recherche LaMOS.

La mise en place de cette application nous a permis de mettre en œuvre nos connaissances théoriques acquises tout au long de notre formation à l'Université de Béjaïa.

Ce travail nous a fourni également un grand apport au niveau de plusieurs domaines. Sur le plan technique, nous avons appris à manipuler Android Studio, PHP , MySQL et pleins d'autres outils.

Nous nous sommes appuyés durant tout le processus de modélisation sur la méthode UML, qui est considéré parmi les langages de modélisation les plus utilisés pour illustrer la démarche de conception.

Le principe but de notre application est de permettre aux différents chercheurs et futur chercheurs de l'unité de recherche « LaMOS » d'accéder à un espace de gestion des Bilans à travers leurs téléphone portable a fin :

- Ajouter des Activités tel que les publications, les conférences, les books etc ;
- Générer les bilans de leurs travaux pour une période de temps précise ;
- Offrir un accès aux internautes de visualiser les travaux des chercheurs.

Bien que les principaux objectifs de notre projet ont été atteints, le système que nous avons développé pourrait être enrichi par d'autres fonctionnalités avancées tel que :

- Ajouter un forum de discussion pour les utilisateurs ;
- Améliorer l'esthétique des différentes interfaces et rajouter d'autres ;
- Système de notification à chaque nouvelle publication ;
- Sécurité et déploiement au niveau de Play store.

Bibliographie

- [1] <https://lamos.org>, consulté le 12/01/2020
- [2] <https://www.nutcache.com/fr/blog/les-methodes-agiles/>, consulté le 22/03/2020
- [3] <https://www.piloter.org/projet/methode/scrum.htm> , consulté le 24/03/2020.
- [4] <https://www.supinfo.com/articles/single/>, consulté le 28/03/2020
- [5] <https://www.gataka.fr/projet-scrum/> , consulté le 01/03/2020
- [6] <https://www.scrumalliance.org/learn-about-scrum/scrums-elearning-series/scrums-artifacts>, consulté le 07/03/2020
- [7] <https://lyontesting.fr/fr/du-bon-usage-des-user-story-et-epic/> , consulté le 08/03/2020
- [8] F.LOTHON, Guide de démarrage scrum. L'Agiliste, 2010.
- [9] P.GÉRARD , Cours UML. IUT de Villetaneuse DUT informatique, Université Paris,2013
- [10] R. P.FRANK VALLÉE , UML 2 en action, de l'analyse des besoins à la conception. 4 ème édition, EYROLLES, 2007
- [11] <http://gdt.oqlf.gouv.qc.ca/ficheOqlf.aspx?IdFiche=8360495>, consulté le 02/03/2020
- [12] <https://web.maths.unsw.edu.au/~lafaye/CCM/pc/pda.html>, consulté le 06/02/2020
- [13] https://www.pmtic.net/sites/default/files/filemanager/memos/pmticenvnumsystemexpl_mobil-tablettesgeneralites.pdf,© PMTIC - LabSET – ULiège 2017, consulté le 12/02/2020
- [14] http://www.contenus-en-ligne.com/avantages_inconvenients_dune_application-mobile, consulté le 07/01/2020
- [15] https://www.companeo.com/applications_mobiles/guide/creation-application-mobile, consulté le 08/03/2020
- [16] Z.MOUSSOUNI et al ,”Conception et réalisation d’une application mobile pour le service de tourisme, cas d’étude ”Wilaya de Bejaia” ”, Mémoire de l’université A.Mira Bejaia, 2016/2017.
- [17] <https://www.editionseni.fr/open/mediabook.aspx?idR=62ffcd2c2a048258d3c4d06fc2598168>, consulté le 16/02/2020
- [18] <http://openclassrooms.com/courses/creez-des-applications-pour-android/l-architecture-d-android>, consulté le 07/03/2020
- [19] <http://dspace.univ-tlemcen.dz/bitstream/112/11054/2/Microsoft20Word20-20chapitre-02-la-programmation-sous-android2028129.pdf>, consulté le 25/03/2020

BIBLIOGRAPHIE

- [20] <https://www.tutorialspoint.com/android/androidarchitecture.html> , consulté le 15/03/2020
- [21] <https://www.definitions-marketing.com/definition/application-android/>, consulté le 06/03/2020
- [22] R.MEIER, Professional Android Application Development. 2012
- [23] <https://mathias-seguy.developpez.com/tutoriels/android/comprendre-cyclevie-activite/>, consulté le 18/03/2020
- [24] B. NAZIM, Android 5 les fondamentaux du développement d'applications Java, 2015
- [25] [Http://www-inf.int-evry.fr/cours/CSC4002/EnLigne/Cours/CoursUML/4.12.6.html](http://www-inf.int-evry.fr/cours/CSC4002/EnLigne/Cours/CoursUML/4.12.6.html), consulté le 23/03/2020
- [26] B.CHERROUX et al, UML2, Pearson Education France, 3ème édition, 2010
- [27] <https://blog.myagilepartner.fr/index.php/2018/08/07/backlog/> , consulté le 24/04/2020
- [28] <https://fr.wikipedia.org/wiki/>, consulté le 17/02/2020
- [29] <http://www.standardduweb.com/xampp.php>, consulté le 20/02/2020
- [30] Wilfried-Erisco MVOU-OSSIALAS, Conception et développement d'une application de gestion d'une bibliotheque", Mémoire de l'université de tunis - Génie logiciel, 2003.
- [31] <http://glossaire.infowebmaster.fr/xml/>.", consulté le 18/04/2020
- [32] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203597-php-hypertext-preprocessor-definition/>, consulté le 08/04/2020
- [33] <http://slideplayer.fr/slide/173472/>.", consulté le 18/04/2020
- [34] K. ABDERRAHMANE et al., "Réalisation d'une Application web pour la gestion des stocks produits finis de Cevital", Mémoire de l'université A. Mira Bejaia, 2012/2013
- [35] L.AUDIBERT, "UML 2 - de l'apprentissage à la pratique", édition Ellipses, 2006
- [36] <https://blog.thiga.co/glossaire/definition-user-story/>, consulté le 05/04/2020
- [37] [http://docwiki.embarcadero.com/RADStudio/Rio/fr/Définition-des-diagrammes-de-séquence-UML-1.5](http://docwiki.embarcadero.com/RADStudio/Rio/fr/D%C3%A9finition-des-diagrammes-de-s%C3%A9quence-UML-1.5), consulté le 27/04/2020
- [38] <http://tvaira.free.fr/dev/uml/UML-DiagrammesDeSequence.pdf> , consulté le 03/04/2020
- [39] <https://www.editions-eni.fr/open/mediabook> , consulté le 13/05/2020
- [40] http://eleydet.free.fr/SN/exposes/SYSML_UML/index.html , consulté le 23/04/2020
- [41] <https://lckj2018.en.made-in-china.com/product/VsGxogFHrhWN/China-Quad-Core-1-3G-Bluetooth-Android-PDA-Device-with-1d-Laser-Scanner.html> , consulté le 20/04/2020
- [42] <https://www.quechoisir.org/guide-d-achat-smartphones> , consulté le 20/04/2020
- [43] <https://www.pinterest.fr/pin> , consulté le 20/04/2020

Annexe A

Diagramme de séquence système

A.1 Diagramme de séquence système ” Gérer les membres d’équipe ”

Les diagrammes de séquence système décrivent le comportement du système.

A.1.1 Diagramme de séquence système ” Ajouter un membre d’équipe ”

La figure suivante représente le diagramme de séquence système ”Ajouter un membre d’équipe”.

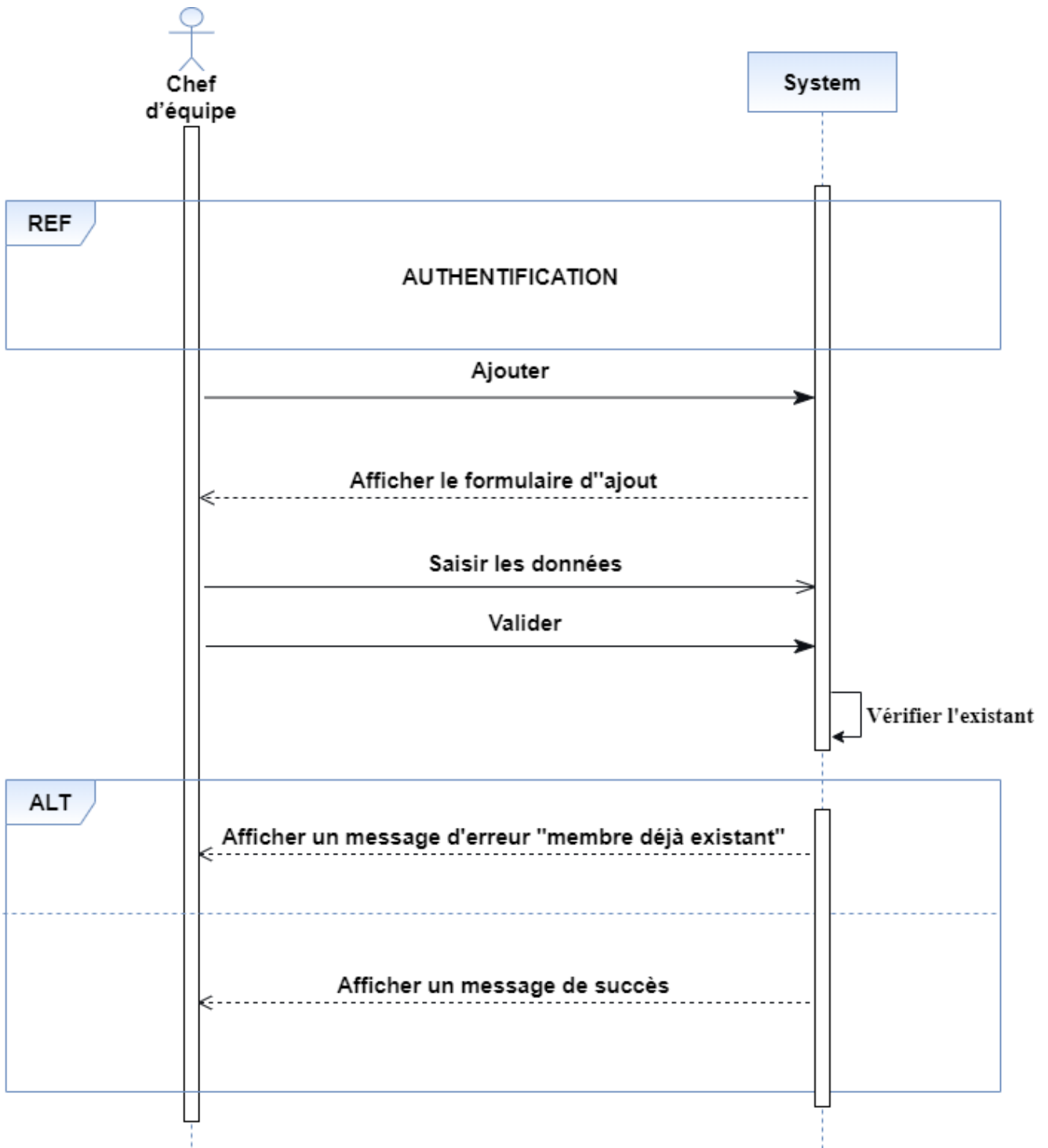


FIGURE A.1 – Diagramme de séquence système ” Ajouter un membre d’équipe ”

A.1.2 Diagramme de séquence système ” Modifier un membre d’équipe ”

La figure suivante représente le diagramme de séquence système ”modifier un membre d’équipe”.

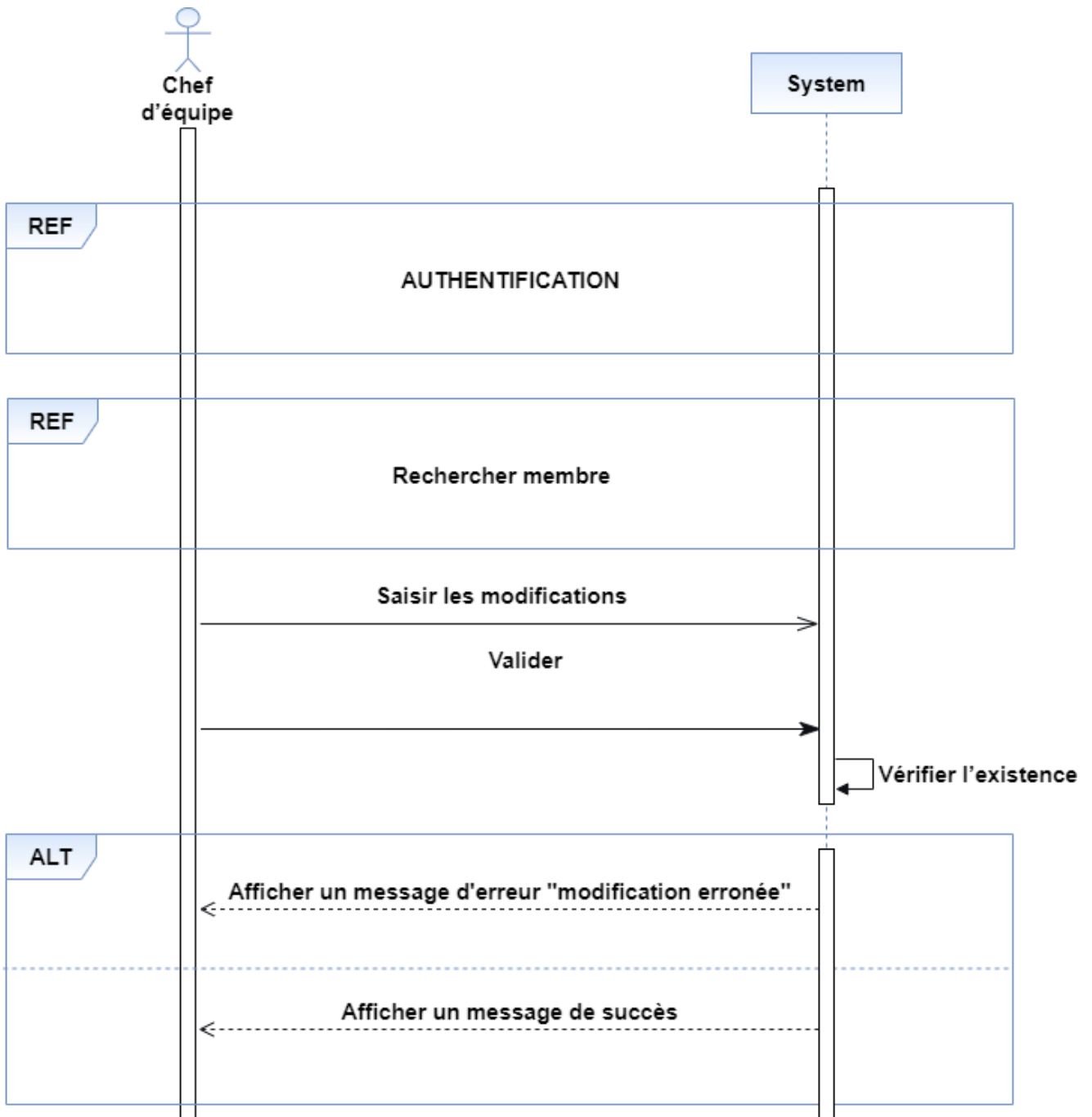


FIGURE A.2 – Diagramme de séquence système ” Modifier un membre d’équipe ”

A.1.3 Diagramme de séquence système ” Supprimer un membre d’équipe ”

La figure suivante représente le diagramme de séquence système ”supprimer un membre d’équipe”.

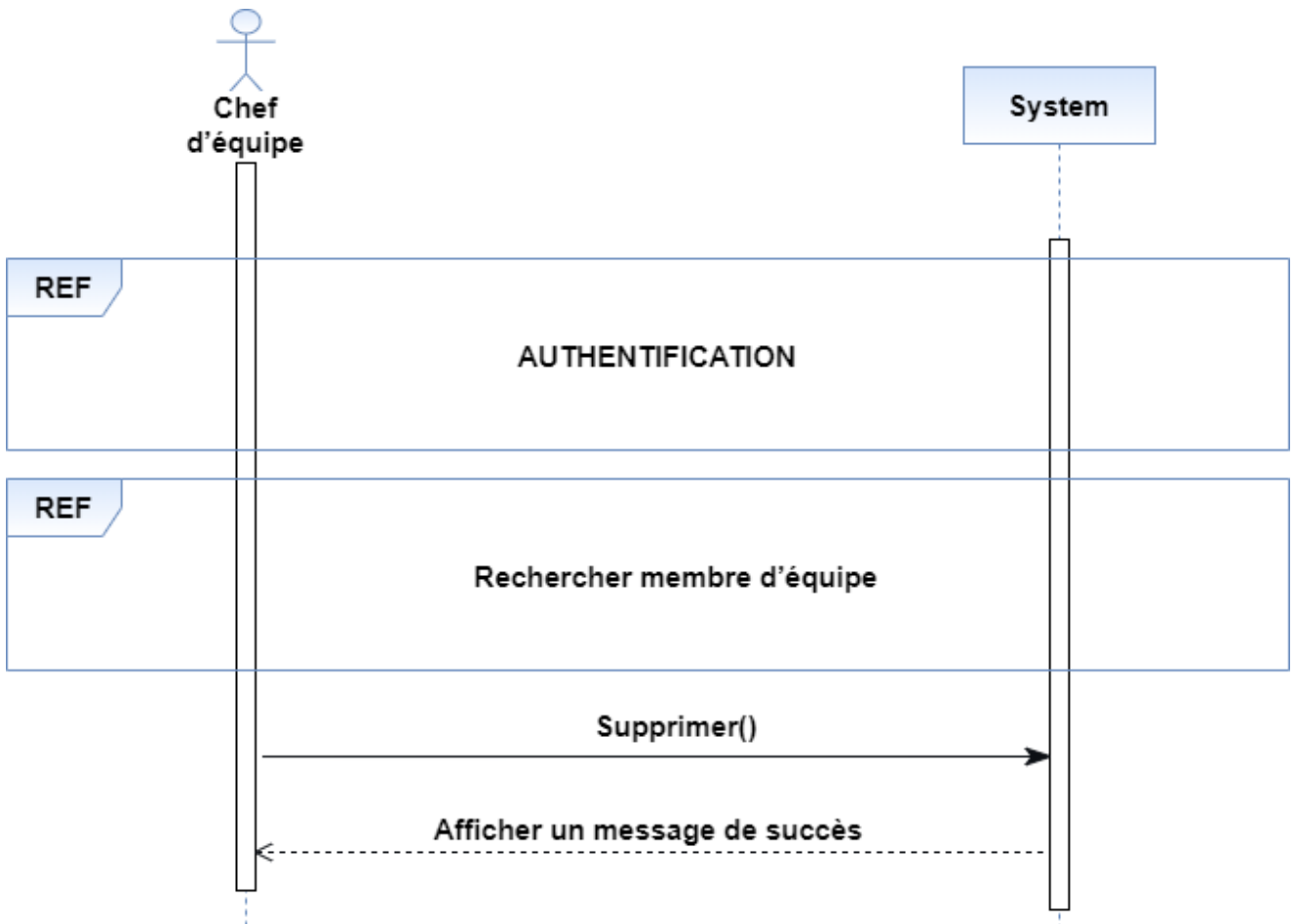


FIGURE A.3 – Diagramme de séquence système ” Supprimer un membre d’équipe ”

A.1.4 Diagramme de séquence système ” Rechercher un membre d’équipe ”

la figure suivante représente le diagramme de séquence système ”rechercher un membre d’équipe”.

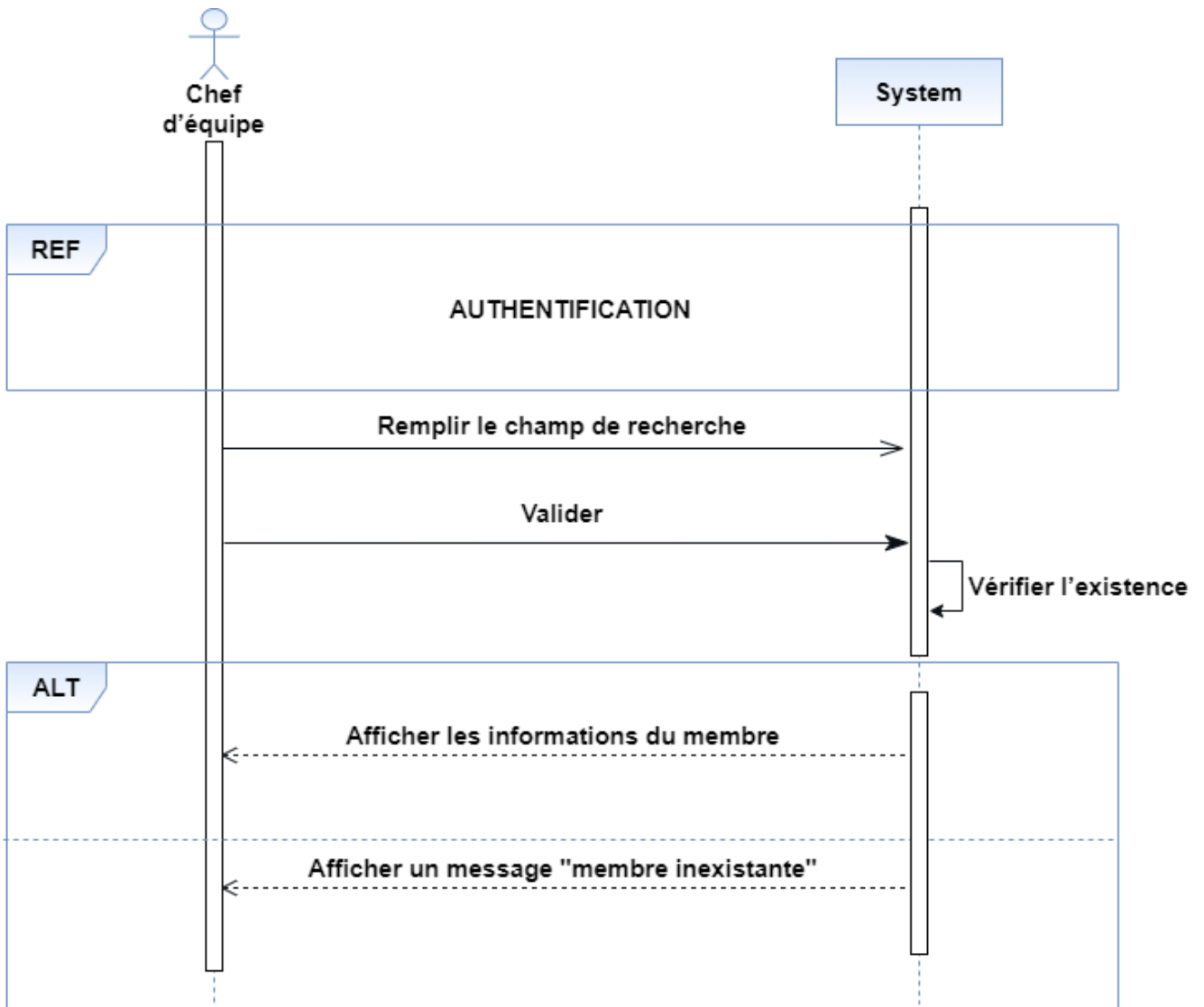


FIGURE A.4 – Diagramme de séquence système ” Rechercher un membre d’équipe ”

A.1.5 Diagramme de séquence système " Générer le bilan d'un membre d'équipe "

La figure suivante représente le diagramme de séquence système "Générer le bilan d'un membre d'équipe".

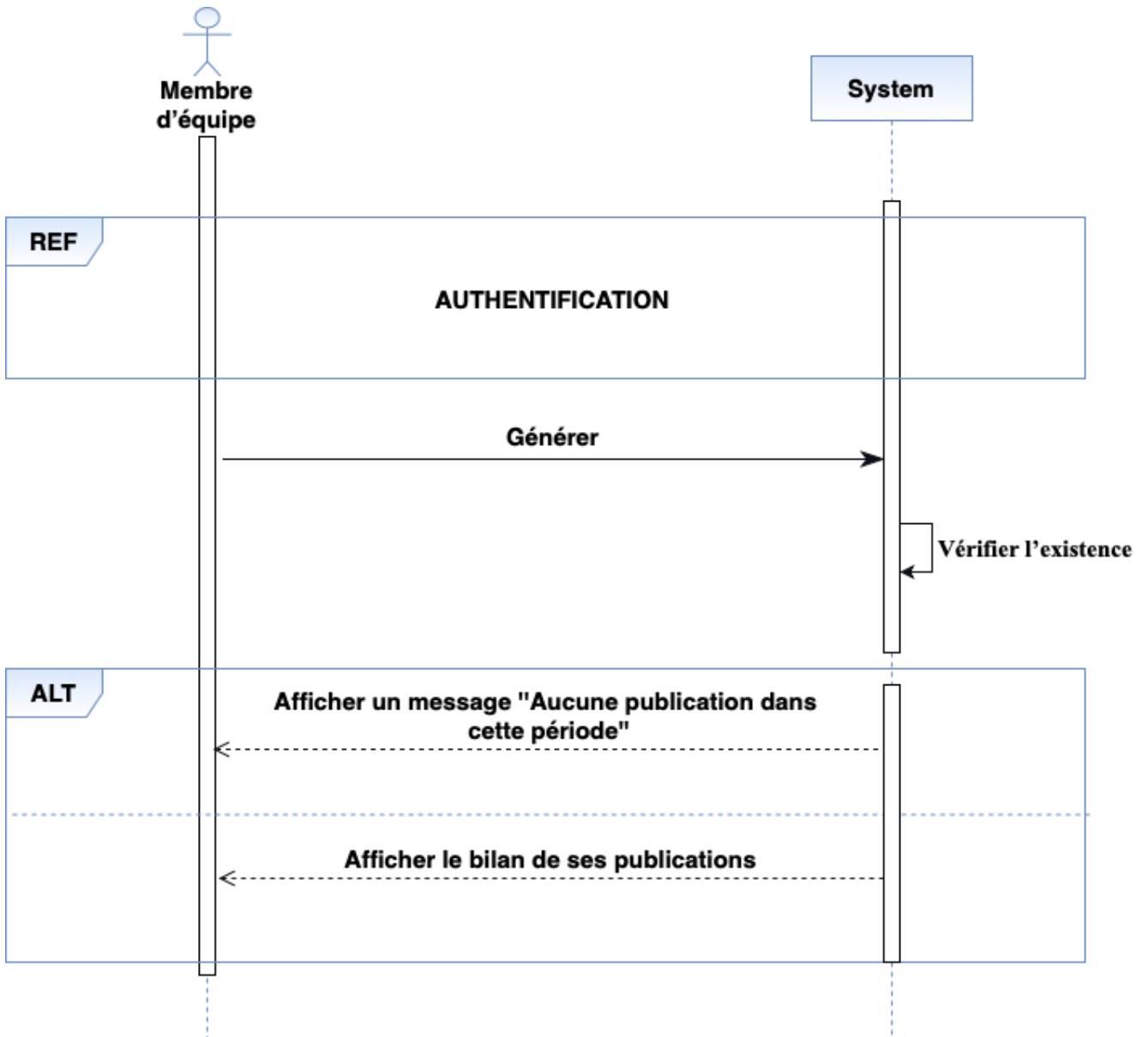


FIGURE A.5 – Diagramme de séquence système " Générer le bilan d'un membre d'équipe "

A.2 Diagramme de séquence système ” Gérer les activités ”

Nous allons présenter dans ce qui suit les diagrammes de séquence système pour les différents méthodes effectuées sur les activités (ajout , modification,suppression et recherche).

A.2.1 Diagramme de séquence système ” Ajouter une activité ”

La figure suivante représente le diagramme de séquence système ”Ajouter une activité”.

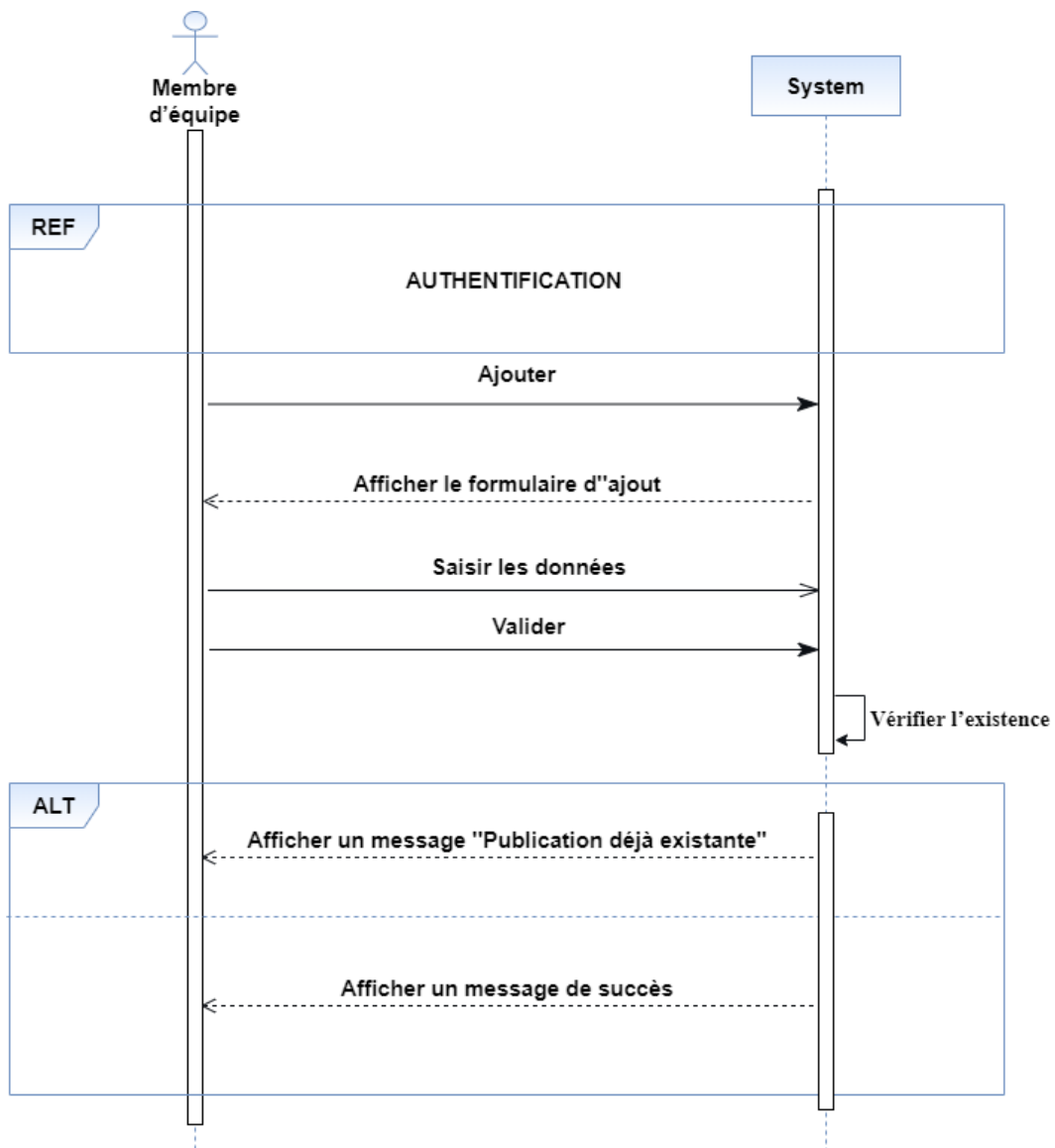


FIGURE A.6 – Diagramme de séquence système ” Ajouter une activité ”

A.2.2 Diagramme de séquence système ” Modifier une activité ”

La figure suivante représente le diagramme de séquence système ”Modifier une activité”.

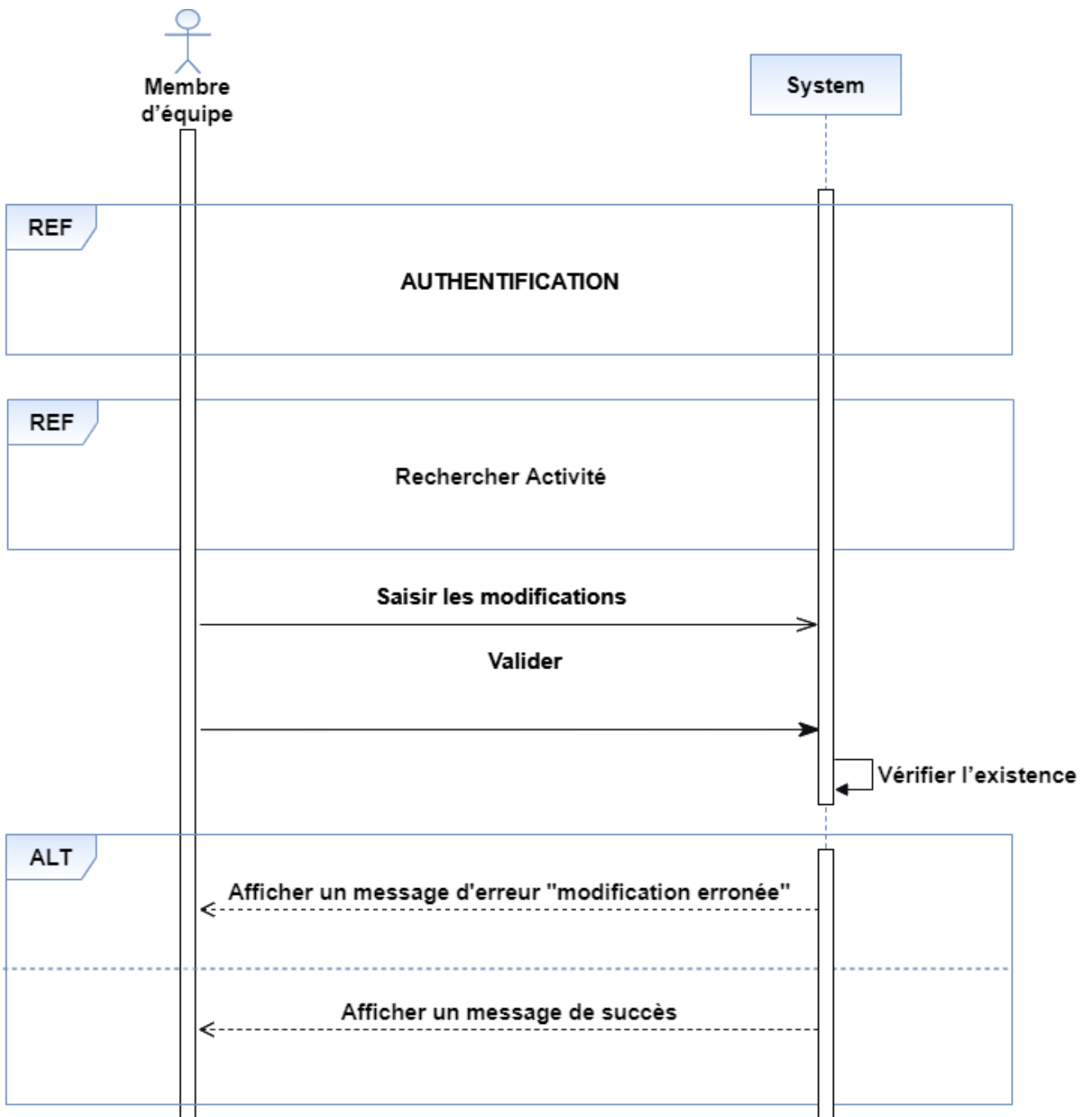


FIGURE A.7 – Diagramme de séquence système ” Modifier une activité ”

A.2.3 Diagramme de séquence système ” Supprimer une activité ”

La figure suivante représente le diagramme de séquence système ”Supprimer une activité”.

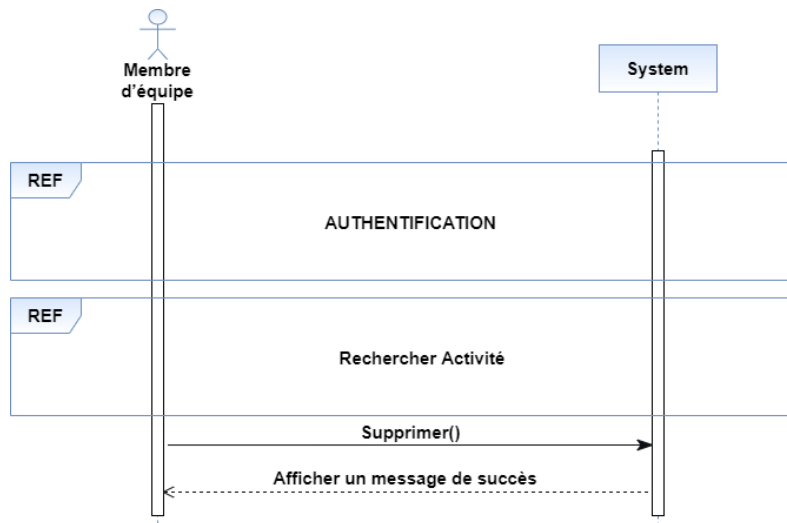


FIGURE A.8 – Diagramme de séquence système ” Supprimer une activité ”

A.2.4 Diagramme de séquence système ” Rechercher une activité ”

La figure suivante représente le diagramme de séquence système ”Rechercher une activité”.

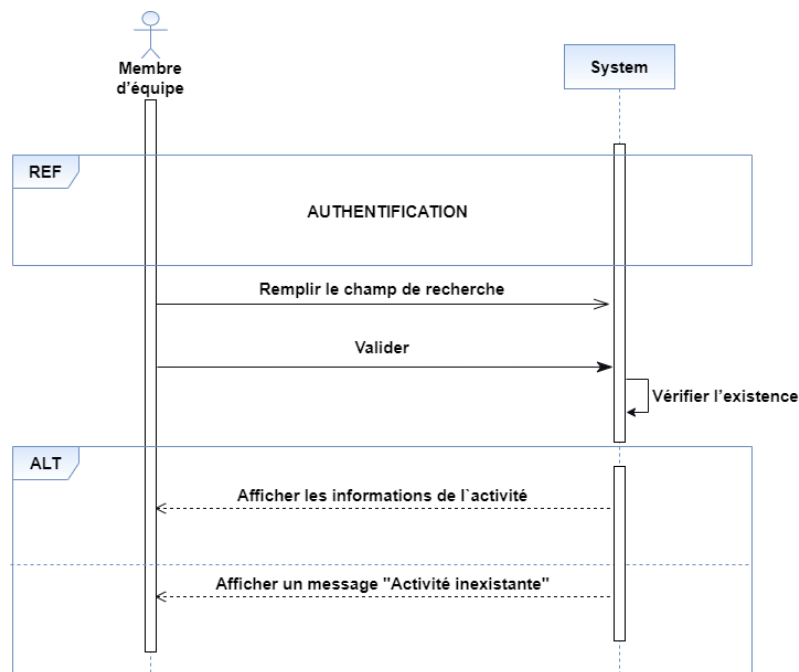


FIGURE A.9 – Diagramme de séquence système ” Rechercher une activité ”

A.3 Diagramme de séquence système " Générer le bilan de l'unité "

La figure suivante représente le diagramme de séquence système "Générer le bilan de l'unité".

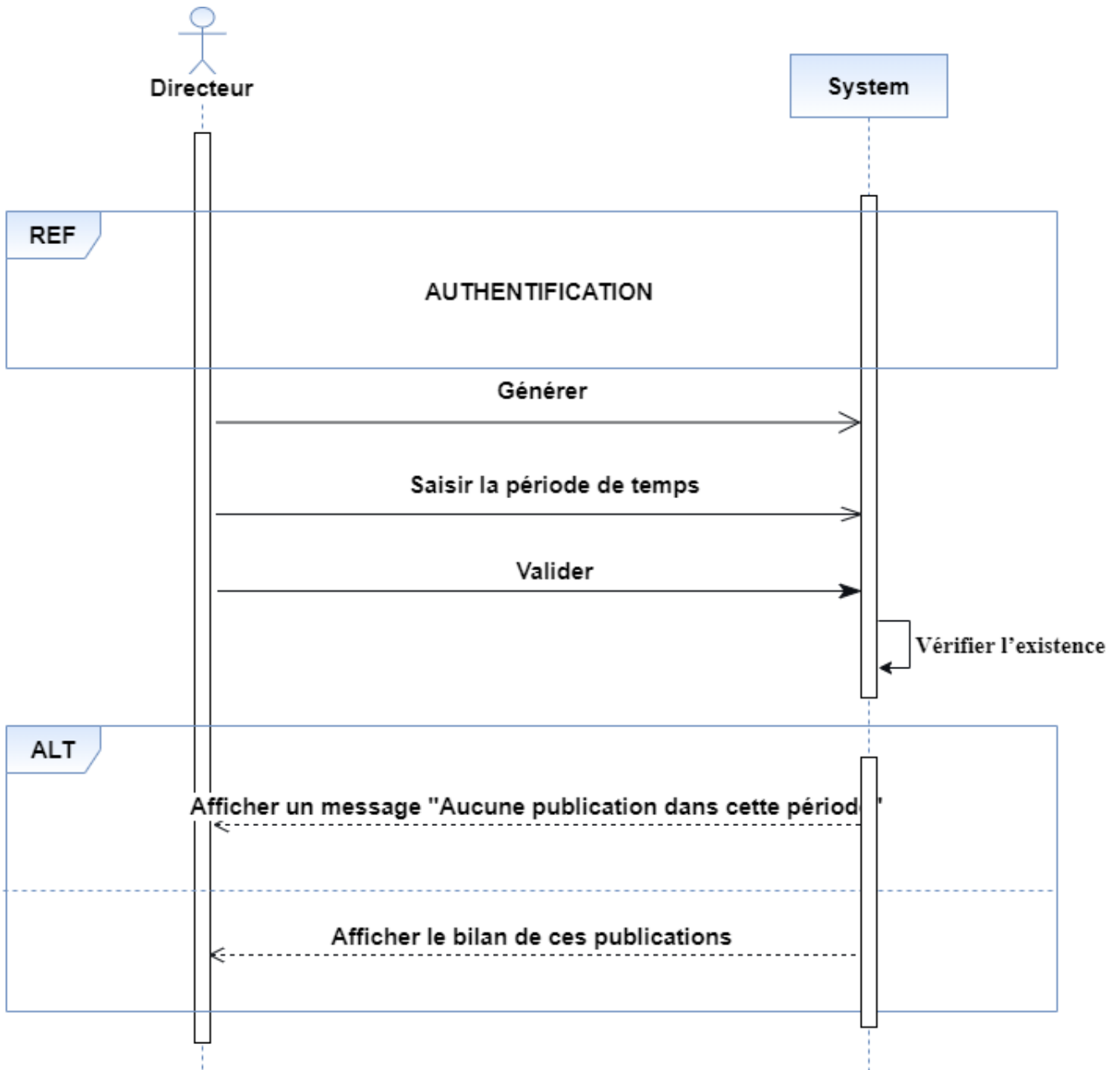


FIGURE A.10 – Diagramme de séquence système " Générer le bilan de l'unité "

A.4 Diagramme de séquence système ” S’authentifier ”

La figure suivante représente le diagramme de séquence système ”S’authentifier”.

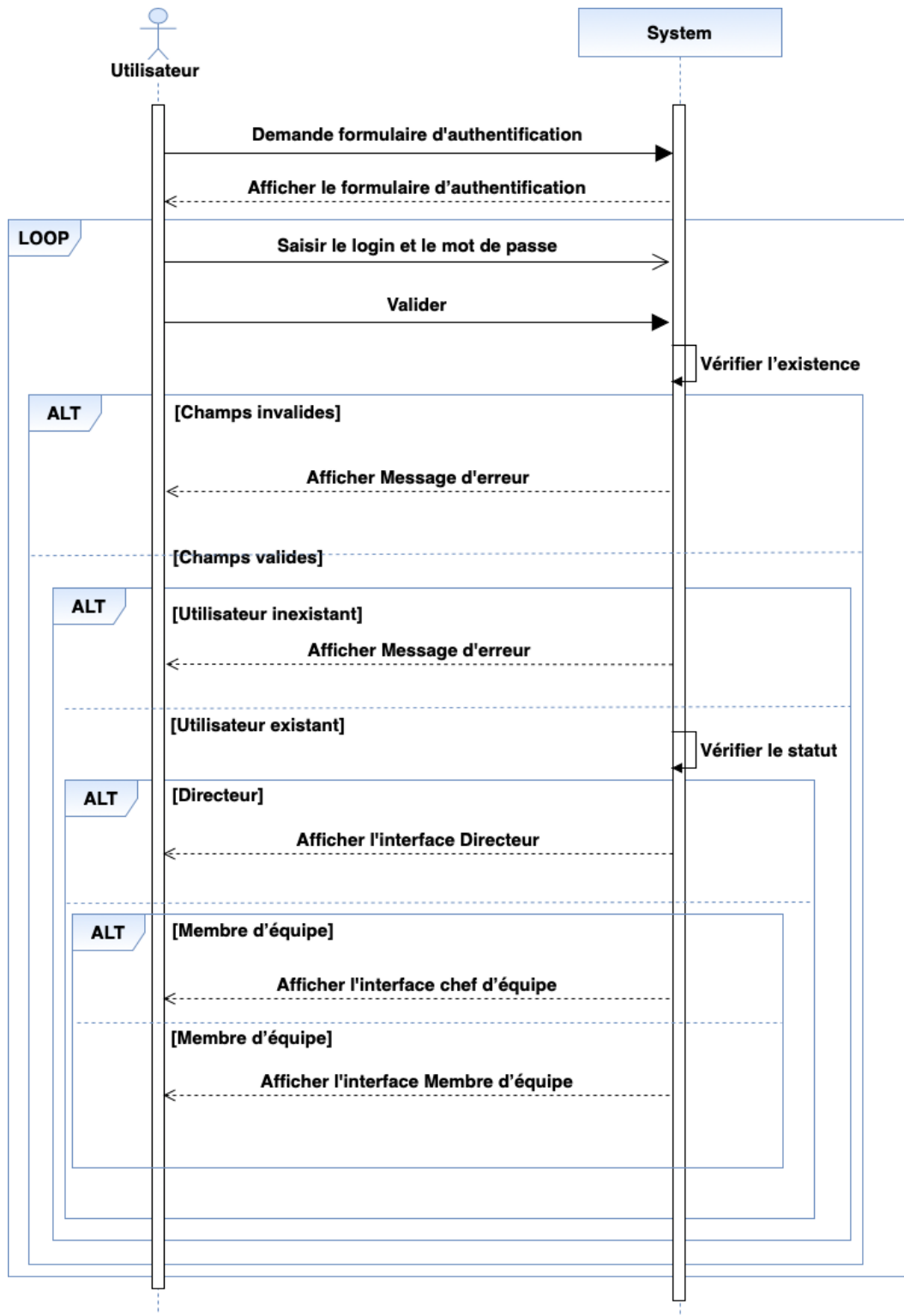


FIGURE A.11 – Diagramme de séquence système ” S’authentifier ”

Annexe B

Présentation des interfaces

B.1 Interface "Profil d'un Membre d'équipe "

Cette interface s'affiche après la connexion d'un membre d'équipe , elle est composée du menu des activités que le membre d'équipe peut effectuer(Ajouter une activité , Générer le bilan de ses activités)

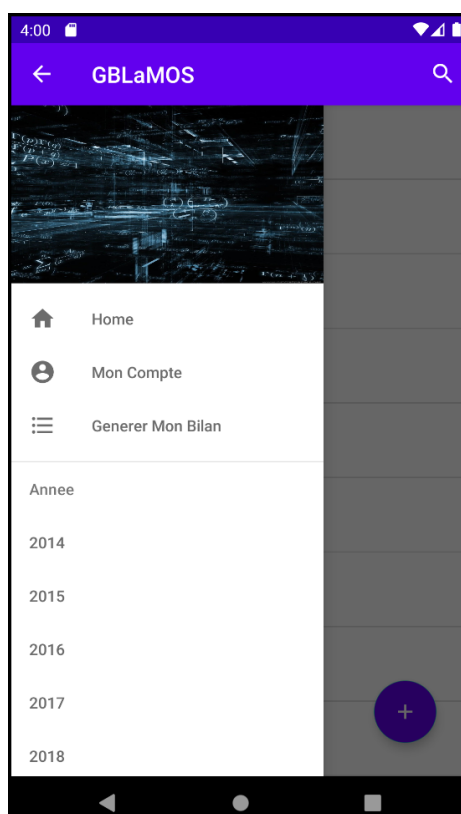


FIGURE B.1 – Interface "Profil d'un Membre d'équipe "

B.2 Interface "Profil d'un Chef d'équipe "

Cette interface s'affiche après la connexion d'un chef d'équipe , elle est composée du menu des activités que le chef d'équipe peut effectuer(Ajouter une activité , Ajouter un membre d'équipe et Générer le bilan de son équipe)

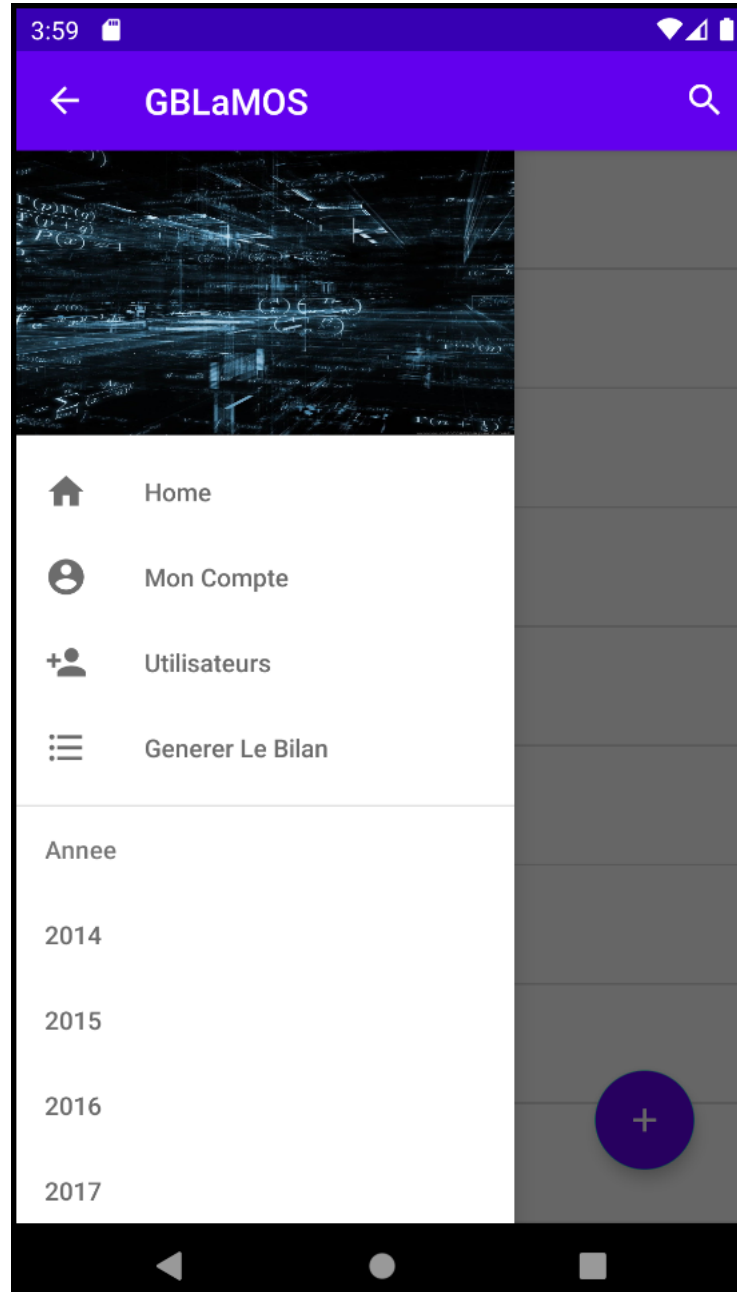


FIGURE B.2 – Interface "Profil d'un Chef d'équipe "

B.3 Interface "Profil d'un Directeur "

Cette interface s'affiche après la connexion d'un directeur , elle est composée du menu des activités que le directeur peut effectuer(Ajouter une activité , Ajouter un membre d'équipe , Ajouter une équipe , Ajouter une sortie pédagogique et Générer le bilan de l'unité)

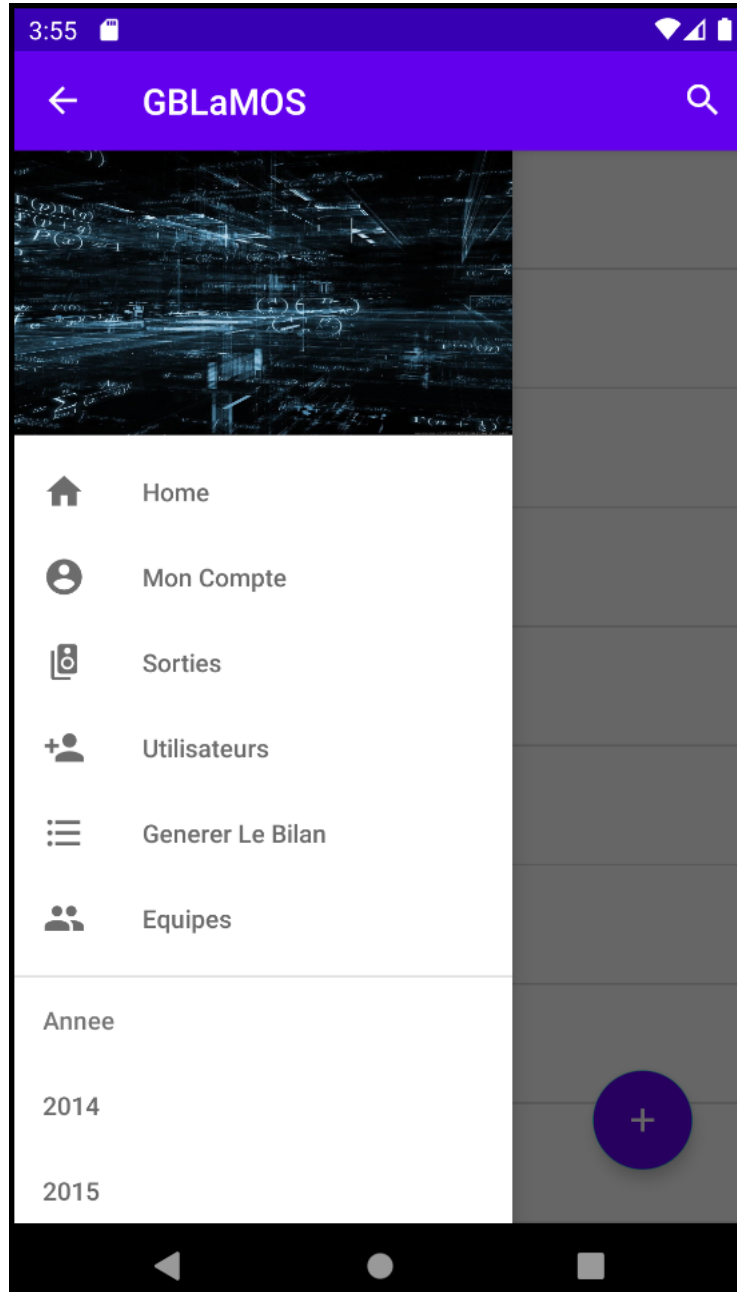


FIGURE B.3 – textinterface "Profil d'un Directeur "

Résumé

L'objectif principal de ce projet est de réaliser une application mobile fiable et aisée qui sert essentiellement à gérer les bilans de l'unité de recherche « LaMOS » et la gestion des membres de l'unité, assurer la sûreté et sécurité des données, le suivi des membres de l'unité, la gestion des activités effectuées par chaque membre d'équipe et la gestion des équipes.

Pour mettre en œuvre notre solution, nous avons utilisé une méthode agile appelée Scrum , qui se base sur UML comme langage de modélisation et le pattern de conception MVC.

Cette conception est mise en œuvre sous l'environnement de développement Intégré (Android Studio) , JAVA et PHP comme langages de programmation, un système de gestion de base de données PhpMyAdmin.

Mots clés : L'unité de recherche LaMOS, Scrum , PHP, Android , gestion des bilans.

Abstract

The main objective of this project is to realize a reliable and easy mobile application which is used primarily to manage the reports of the research unit "LaMOS" and the management of the members of the unit, to ensure the safety and security of the data, monitoring unit members, managing the activities carried out by each team member and managing teams.

To implement our solution, we used an agile method called Scrum, which is based on UML as the modeling language and the MVC design pattern.

This design is implemented under the Integrated development environment (Android Studio), JAVA and PHP as programming languages, a PhpMyAdmin database management system.

Keywords : LaMOS research unit, Scrum, PHP, Android, balance sheet management.