



Université Joseph KI-ZERBO (UJKZ)

**Unité de Formation et de Recherche en
Sciences Exactes et Appliquées
(U.F.R./S.E.A.)**

Département : Informatique

Année académique : 2018-2019

UJKZ/SEA/Dept-Infor/LSIR/2019

**AGENCE NATIONALE DE PROMOTION
DES TECHNOLOGIES DE
L'INFORMATION ET DE LA
COMMUNICATION**

RAPPORT DE STAGE DE FIN DE CYCLE



**Thème : DÉVELOPPEMENT D'UNE APPLICATION MOBILE DE
GESTION DES MISSIONS : CAS DU CIM/ANPTIC**



Pour l'obtention de la ***Licence Informatique***
Option : ***Système d'Information et Réseau (SIR)***

Stage effectué par :

ZONGO Souleymane

Période de stage : du 13 Avril au 13 Juillet 2022

Lieu de Stage : ANPTIC (Patte d'oie, Ouagadougou – Burkina Faso)

Superviseur :

M. BARA Ousmane,

Enseignant en informatique à
l'Université Joseph KI-ZERBO

Maitre de Stage :

M. Antoine SAMPEBGO,

Ingénieur de Conception en ILSI à
l'ANPTIC

DÉDICACE

A mes parents qui m'ont soutenu et encouragé tout le long de mon cursus.

A mon oncle et ma tante pour leur soutien.

A mes frères et sœurs

A mes ami(e)s

REMERCIEMENTS

Je rends grâce à Allah pour la force et le courage qu'il m'a accordé pour avoir tenu bon pendant ces longues années. Je remercie tous ceux qui ont été avec moi de près ou de loin et qui m'ont apporté leur aide.

Ma gratitude va à l'endroit de :

- **Dr Aminata ZERBO/SABANE**, Ministre et Enseignante à l'U.F.R./S.E.A. ;
- **M. Boukary YOUNGBARE**, Directeur des Systèmes Applicatifs (DSA)/ANPTIC ;
- **M. Antoine SAMPEBGO**, NOTRE maître de stage et Ingénieur de Conception en ILSI à l'ANPTIC ;
- **M. Diloman Hermann Ascension SOMA**, Chef de département à la DSA/ANPTIC ;
- **M. Ousmane BARA**, Enseignant à l'U.F.R./S.E.A. et mon superviseur ;
- **M. Dominique BAGRE**, agent de l'ANPTIC ;
- **Mlle Korotimi SÉRÉ**, responsable de l'application CIM ;
- **ZEBA Mohamed, NIKIEMA Issouf**, pour leur accompagnement ;
- A tous les enseignants du département Informatique ;

AVANT-PROPOS

Créée en 1974, L'Université Joseph KI-ZERBO, première Université publique au Burkina Faso, est un établissement à caractère scientifique, culturel et technique. Elle se compose :

- ✓ De cinq (05) Unités de Formation et de Recherche (UFR) : Sciences Exactes et Appliquées (UFR/SEA), Sciences de la Vie et de la Terre (UFR/SVT), Sciences de la Santé (UFR/SDS), Lettres, Arts et Communication (UFR/LAC) et Sciences Humaines (UFR/SH) ;
- ✓ De trois (03) instituts : l'Institut Burkinabè des Arts et Métiers (IBAM), l'Institut Supérieur des Sciences de la Population (ISSP), l'Institut Panafricain d'Etude et de Recherche sur les Médias, l'Information et la Communication (IPERMIC) ;
- ✓ De trois écoles doctorales dans les domaines : Lettre, Sciences Humaines et Communication, Sciences de la Santé et Sciences et Technologie.

Chaque Unité de Formation et de Recherche est subdivisée en départements ; Ainsi l'UFR/SEA comprend les départements **Informatique, Mathématique, Physique et Chimie**.

Le Département Informatique a été créé en 2013 avec le passage au système Licence Master Doctorat (LMD). Ce département a pour objectif la formation dans les domaines de l'Informatique.

Au département Informatique, l'obtention de la Licence à la fin du premier cycle est soumise à un stage d'une durée minimale de trois (03) mois. L'objectif de ce stage est de permettre aux étudiants en fin de cycle de s'imprégner des réalités de la vie professionnelle et de parfaire leurs connaissances théoriques et pratiques acquises au cours des trois années de formation. C'est dans cette optique que nous avons été accueilli dans les locaux de l'**ANPTIC** pour notre stage.

SIGLES ET ABRÉVIATIONS

Acronyme	Signification
ANPTIC	Agence Nationale de Promotion des TIC
API	Application programming Interface
BF	Burkina Faso
CIM	Circuit intégré des missions
COCOMO	COConstructive COst MOdel
CSS	Cascading Style Sheets
CU	Cas d'Utilisation
GHZ	Gigahertz
GNU	GNU's Not UNIX
Go	Gigaoctet
HP	Hewlett-Packard
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
iOS	iPhone OS
JSX	Javascript Syntax eXtension
OHA	Open Handset Alliance
OS	Operating System
RAM	Random Access Memory
REST	Représentationnel State Transfer
ROM	Read Only Memory
SDK	Software Development Kit
SGBDRO	Système de Gestion de Base de Donnée Relationnel -Objet
SIR	Système d'information et Réseau
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SSD	Solid State Drive
TIC	Technologies de l'information et de la communication
UJKZ	Université Joseph Ki-Zerbo
UML	Unified Modeling Language
XML	eXtensible Markup Language

INDEX DES FIGURES

Figure 1 : Organigramme de l'ANPTIC	12
Figure 2 : Étapes de la méthode SCRUM.....	17
Figure 3 : Les diagrammes UML	18
Figure 4 : Plan prévisionnel.....	20
Figure 5 : Diagramme de cas d'utilisation.....	24
Figure 6 : diagramme de séquence « S'authentifier ».....	29
Figure 7 : Diagramme de séquence de « Consulter liste des missions »	30
Figure 8 : Diagramme de séquence de « Approuver mission ».....	31
Figure 9 : Diagramme de séquence de « Budgétiser mission ».....	32
Figure 10 : Diagramme de séquence de « Valider mission »	33
Figure 11 : Diagramme de séquence de « Clôturer mission »	34
Figure 12 : Diagramme global d'activité.....	35
Figure 13 : Diagramme de classe.....	36
Figure 14 : logo Android.....	37
Figure 15 : logo iOS.....	37
Figure 16 : logo Windows phone.....	38
Figure 17 : BlackBerry OS	38
Figure 18 : logo de Flutter.....	41
Figure 19 : logo de React Native	41
Figure 20 : logo de Ionic.....	42
Figure 21 : le logo de Dart	44
Figure 22 : logo de Node.js.....	44
Figure 23 : logo de SQL.....	45
Figure 24 : logo API.....	45
Figure 25 : Informations sur l'ordinateur utilisé	46
Figure 26 : Informations du système du Smartphone utilisé	47
Figure 27 : logo de VSCode	48
Figure 28 : logo de Enterprise Architect.....	48
Figure 29 : logo de Git.....	49
Figure 30 : logo de GitHub	49
Figure 31 : logo de Trello.....	49
Figure 32 : logo de PostgreSQL	50
Figure 33 : logo de Postman.....	50
Figure 34 : page de connexion.....	53
Figure 35 : page d'accueil.....	54
Figure 36 : menu de gestion de mission	54
Figure 40 : option de Budgétisation.....	55
Figure 42 : détails d'une mission.....	55
Figure 44 : page de recherche d'une Facture.....	56

INDEX DES TABLEAUX

Tableau 1 : Comparaison des Méthodes Agiles.....	16
Tableau 2 : CU S'authentifier.....	25
Tableau 3 : CU Liste des missions	26
Tableau 4 : CU Approuver mission	26
Tableau 5 :CU budgétiser mission	27
Tableau 6 : CU valider mission.....	28
Tableau 7 : CU clôturer mission.....	29
Tableau 8 : Model COCOMO.....	51

TABLES DES MATIERES

DÉDICACE.....	2
REMERCIEMENTS	3
AVANT-PROPOS	4
SIGLES ET ABRÉVIATIONS	5
INDEX DES FIGURES	6
INDEX DES TABLEAUX	7
TABLES DES MATIERES	8
INTRODUCTION.....	Erreur ! Signet non défini.
CHAPITRE I : GÉNÉRALITÉS.....	11
I. Présentation et Objectifs de la structure d'accueil	11
1. Présentation de la structure d'accueil	11
2. Objectifs de la structure d'accueil	11
II. Thème.....	13
1. Présentation du projet	13
2. Problématique	14
3. Résultats attendus	14
CHAPITRE II : SPÉCIFICATIONS ET ANALYSE DES BESOINS	15
I. Méthode de conduite du projet.....	15
1. Les méthodologies agiles	15
2. Étude Comparative des différentes méthodes agiles	15
3. Choix de la méthodologie SCRUM	16
4. Équipe de Projet	17
5. Langage de Modélisation	18
6. Planning prévisionnel	19
II. Etude de l'existant	21
CHAPITRE III : ANALYSE ET CONCEPTION DU SYSTEME	22
I. Domaine d'étude et fonctionnalités.....	22
1. Domaine d'étude	22
2. Fonctionnalités	22
II. Modélisation.....	Erreur ! Signet non défini.
1. Diagramme des cas d'utilisations	Erreur ! Signet non défini.

DÉVELOPPEMENT D'UNE APPLICATION MOBILE DE GESTION DES MISSIONS : CAS DU CIM/ANPTIC

2.	Description textuelle de quelques cas d'utilisations.....	24
3.	Diagramme de séquence de quelques cas d'utilisation	29
4.	Diagramme d'activité	34
5.	Diagrammes de classes	35
CHAPITRE IV : MISE EN ŒUVRE DE L'APPLICATION		37
I.	Outils technologiques	37
1.	Applications mobiles et systèmes d'exploitation	37
2.	Application native et Cross-platforms	38
2.1	Application native	40
2.2	Application cross-platforms	40
2.3	Choix du type d'application	40
3.	Framework	40
3.1	Qu'est ce qu'un Framework	40
3.2	Etude comparative des Framework	42
3.2.1	Les différents Framework	42
3.2.2	Etude comparée	44
3.2.3	Choix du Framework	45
4.	Langage informatique	43
5.	Environnement matériel.....	46
6.	Environnement logiciel.....	48
II.	Estimation des Coûts	51
1.	Méthode COCOMO et coût de prestation de l'équipe	51
2.	Coût du matériel de développement.....	52
3.	Coût du matériel de déploiement.....	52
III.	Mise en œuvre	53
1.	Page de connexion	53
2.	Page d'accueil	53
3.	Page de gestion des missions	54
4.	Détails d'une mission	55
5.	Recherche d'une mission	56
CONCLUSION		57
BIBLIOGRAPHIE ET WEBOGRAPHIE		58

INTRODUCTION

L'administration publique compte de nos jours plus de cent quatre-vingt mille (180 000) agents repartis dans les ministères, institutions et collectivités territoriales. Ces agents ont la lourde tâche de réaliser plusieurs activités pour l'atteinte des objectifs des programmes de leur structure. En effet, chaque jour, diverses structures administratives organisent des ateliers de réflexion, des sessions de formation, des colloques, etc. dont prennent part des agents de l'Etat. L'exécution de ces missions, occupe une grande partie du Budget de l'Etat.

Afin d'optimiser l'allocation des budgets aux missions des structures de l'Etat, de suivre de façon efficace l'exécution des missions par les agents, il est apparu opportun de mettre en place un système informatique de gestion des missions. C'est ainsi que la plateforme de Circuit Intégré de Mission (CIM) a été mise en place. CIM est une application de gestion des missions adoptée dans l'administration Publique. Certaines fonctionnalités de CIM ne sont pas adaptées à une utilisation mobile.

Cependant, au vu de la mobilité des agents de la fonction publique et dans la recherche de réactivité quant au traitement des missions, le développement d'une instance mobile s'impose.

C'est dans cette optique qu'intervient nos trois (03) mois de travail au sein de L'Agence Nationale de Promotion des Technologies de l'Information et de la Communication (ANPTIC). Notre tâche consiste à mettre en place une version mobile de CIM. Notre travail s'est articulé sur le thème suivant : « **Développement d'une application mobile de gestion des missions : cas du CIM/ANPTIC.** »

Le présent rapport qui décrit nos travaux pour la mise en place de la solution est structuré en quatre (04) chapitres.

Le premier chapitre porte sur la présentation de notre structure d'accueil et du thème. Le second chapitre spécifie les besoins et les méthodologies utilisés pour leurs analyses. Le troisième chapitre est consacré à l'analyse et à la conception du système. Le quatrième et dernier chapitre de notre document décrit la mise en œuvre de notre système.

CHAPITRE I : GÉNÉRALITÉS

I. Présentation et Objectifs de la structure d'accueil

1. Présentation de la structure d'accueil

L'ANPTIC a été créée par Décret 2014-055/PRES/PM/MEF/MDENP/MFPTSS du 07 février 2014. C'est la concrétisation de la volonté politique des autorités burkinabè de faire des TIC un levier de développement de l'économie nationale et s'inscrit dans les référentiels de développement du pays.

Elle est l'autorité nationale en matière de réalisation de grands programmes TIC. Elle a pour objectif d'assurer la mise en œuvre des grands programmes de développement des Technologies de l'Information et de la Communication (TIC).

2. Objectifs de la structure d'accueil

L'ANPTIC a pour objectifs de :

- ❖ Favoriser l'approfondissement de l'appropriation des technologies de l'information et de la communication ;
- ❖ Favoriser l'innovation à travers le développement d'une expertise nationale reconnue et d'une industrie locale de services performantes et compétitives, basés sur ces technologies ;
- ❖ Assurer la mise en œuvre des projets et programmes de développement des technologies de l'information et de la communication ;
- ❖ Opérationnaliser la stratégie du Gouvernement en matière d'administration électronique ;
- ❖ Promouvoir l'utilisation des technologies de l'information et de la communication dans les autres domaines de développement économique, social, scientifique et culturel ;
- ❖ Assurer l'opérationnalisation de la cyber-stratégie nationale adoptée en 2004.

DÉVELOPPEMENT D'UNE APPLICATION MOBILE DE GESTION DES MISSIONS : CAS DU CIM/ANPTIC

- Organigramme de l'ANPTIC

L'équipe de l'ANPTIC se compose selon l'organigramme ci-dessous :

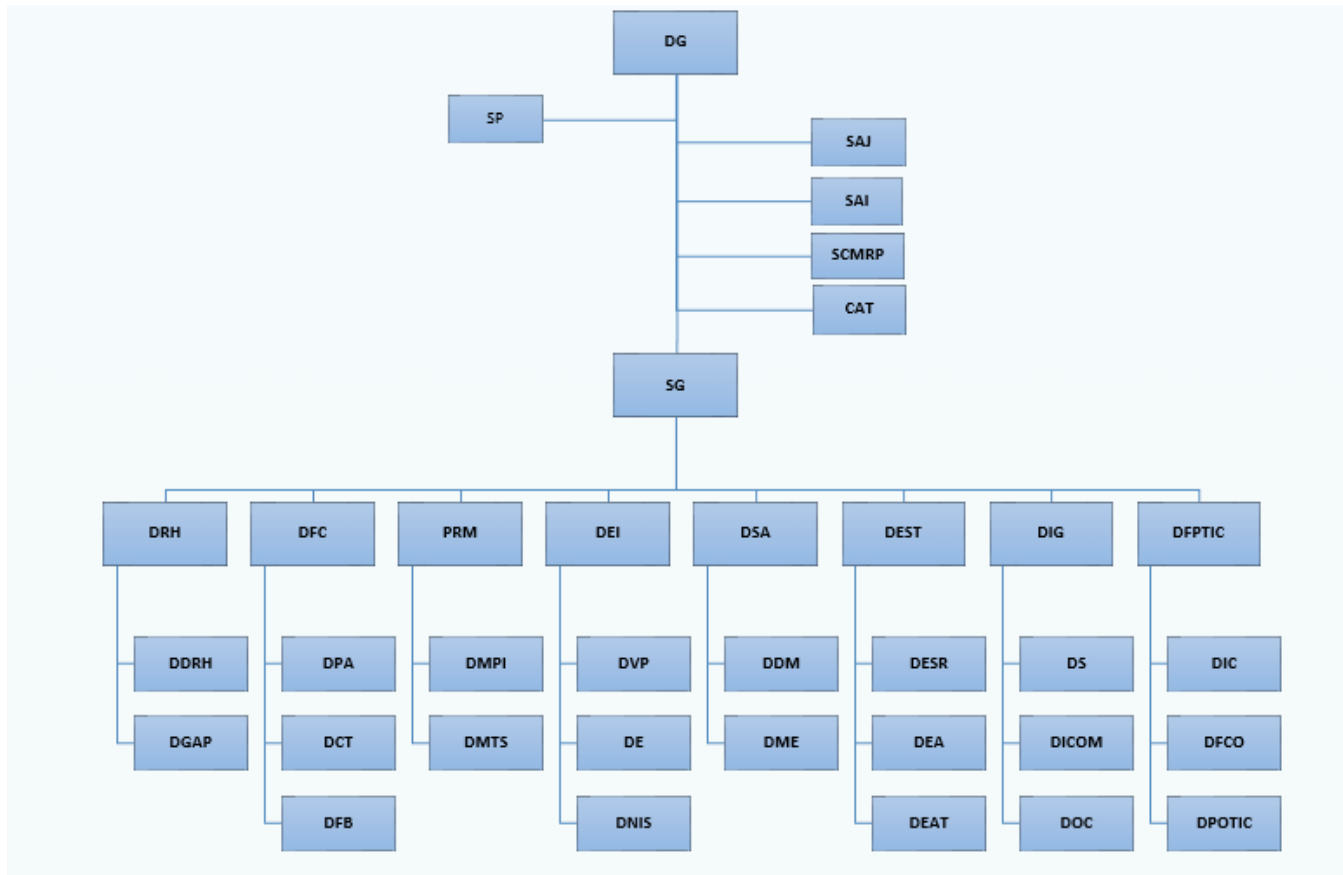


Figure 1 : Organigramme de l'ANPTIC

CA : Conseil d'Administration	DE : Département des Études	DICOM : Département des Infrastructures de Communication	DNIS : Département de la Normalisation et de l'Intégration des Systèmes
DG : Direction Générale	DEA : Département de l'Exploitation des Applications	PRM : Personne Responsable des Marchés	DOC : Département des Outils de Communication
SG : Secrétariat Général	DEAT : Département des Équipements et de l'Assistance Technique	SP : Secrétariat Particulier	DPA : Département du Patrimoine et de l'Approvisionnement
DFC : Direction des Finances et de la Comptabilité	DESR : Département de l'Exploitation et du	DS : Département de la Sécurité	
DRH : Direction des Ressources Humaines			
DCT : Département de			

DÉVELOPPEMENT D'UNE APPLICATION MOBILE DE GESTION DES MISSIONS : CAS DU CIM/ANPTIC

la Comptabilité et de la Trésorerie DDM : Département du Développement et de la Maintenance DDRH : Département du Développement des Ressources Humaines DSA : Direction des Systèmes Applicatifs	Support Réseaux DFB : Département des Finances et du Budget DFCO : Département de la Formation Continue DGAP : Département de Gestion Administrative et de la Paie SAI : Service d'Audit Interne	SAJ : Service des Affaires Juridiques DME : Département de la Mise en Exploitation DMPI : Département des Marchés de Prestations Intellectuelles DMTS : Département des Marchés de Travaux et biens de Services	DPOTIC : Département de la Promotion des Outils des TIC CAT : Cellule d'Appui Technique DVP : Département de la Veille et de la Prospective DIC : Département de l'Innovation et de la Certification SCMRP : Service de la Communication, du Marketing et des Relations Publiques
--	--	--	--

Pour notre stage nous avons été accueillis dans la Direction des Systèmes Applicatifs (DSA) qui est chargée d'assurer la maintenance des applications, le développement des services en ligne et la maîtrise d'œuvre pour la mise en place des applications.

II. Thème

1. Présentation du projet

L'ANPTIC possède une plateforme web de gestion de mission adoptée dans l'administration publique qu'elle met à la disposition de ses agents. En effet, CIM permet de gérer les missions.

Le projet qui nous a été soumis entre dans le cadre d'une solution mobile.

Tout en collaborant avec les agents de l'ANPTIC, le projet consiste donc à mettre en place une version mobile de cette plateforme basée sur la gestion des missions. Le but étant de permettre aux agents de mieux gérer leurs missions, d'être plus productifs et plus réactifs tout en étant sur le terrain.

2. Problématique

Pour le suivi d'une mission, les agents sont priés de se rendre sur la plateforme web CIM à l'aide d'un ordinateur. Cette façon de faire engendre quelques difficultés telles que :

- La réactivité lente des agents dans l'élaboration des ordres de missions : il faut disposer d'un ordinateur car l'initiation d'une mission n'est pas adaptée ;
- Le non suivi des missions via des notifications régulières.

3. Résultats attendus

L'objectif est de concevoir et de réaliser une version mobile dans le but de gagner en temps et en productivité afin d'accroître les performances des agents.

L'application développée devra permettre :

- d'approuver les missions après vérifications des informations de la mission ;
- de budgétiser les missions ;
- de valider des missions (envoi de notifications aux participants de la mission avec les informations) ;
- de prévisualiser l'ordre de mission généré ;
- de vérifier les missions ;
- de clôturer les missions ;
- de consulter les missions ;
- de notifier l'initiateur à chaque changement d'état de la mission ;
- de notifier les utilisateurs devant accomplir des actions pour faire évoluer le traitement de la mission.

L'application devra offrir également certains éléments de paramétrage tels que la mise à jour des véhicules, des localités, des marques et modèles, des motifs de mission, ...

Elle doit être simple d'utilisation et permettre d'obtenir des statistiques de base à définir sur les missions. Les notifications pourront se faire via le mail, le SMS ou WhatsApp.

CHAPITRE II : SPÉCIFICATIONS ET ANALYSE DES BESOINS

I. Méthode de conduite du projet

1. Les méthodologies agiles

Le choix d'une méthodologie de conduite de projet permet à tous les acteurs de travailler en parfaite symbiose tout en respectant les règles définies à cet effet. Ce choix dépend de la nature du projet et de sa taille. Lorsqu'il s'agit d'un projet, où toutes les données ne sont pas réunies dès le départ et où les besoins peuvent évoluer, il est recommandé de s'orienter vers une méthode agile. Se basant sur ces arguments susmentionnés, nous avons opté pour une méthode agile dans le cadre de ce projet. C'est une méthode itérative et incrémentale où les objectifs vont être réalisés pas à pas, tout en veillant à intégrer le client à la phase de développement afin d'obtenir un produit selon les attentes du client ainsi que la cohérence entre les membres de l'équipe.

2. Étude Comparative des différentes méthodes agiles

Il existe plusieurs méthodes AGILES dont les plus connues sont :

- XP (eXtreme Programming) ;
- FDD (Feature Driven Development);
- Scrum.

Ci-dessous, un tableau comparatif des différentes méthodologies agiles précédemment citées :

Méthode	Caractéristiques	Avantages	Désavantages
XP	Refaire à chaque fois la conception afin d'obtenir un système performant et capable de répondre aux changements	Développement réduit par le client, petites équipes, développement en binôme, construction quotidienne	Les pratiques individuelles sont bien adaptées dans plusieurs situations, les pratiques de gestion le sont moins.

DÉVELOPPEMENT D'UNE APPLICATION MOBILE DE GESTION DES MISSIONS : CAS DU CIM/ANPTIC

Méthode	Caractéristiques	Avantages	Désavantages
			Ne s'intéresse pas à l'étude fonctionnelle.
FDD	Développement dirigé par les fonctionnalités	Processus en 5 étapes de développement se basant sur les composants orientés objets ; période d'itération très courte allant de quelques heures à 2 semaines	L'équipe a tendance à abandonner les éléments non-fonctionnels tels que la refactorisation ou la négligence de maintenance. Cela pourrait rendre le logiciel difficile à changer au fil du temps.
SCRUM	Focaliser l'équipe de façon itérative sur un ensemble de fonctionnalités à réaliser	Indépendant, petites équipes à organisation automatique de développement, cycles d'un mois	Scrum spécifie en détail comment gérer les cycles d'un mois mais l'intégration et les tests d'acceptation ne sont pas détaillés.

Tableau 1 : Comparaison des Méthodes Agiles

Pour la conduite de ce projet, nous avons opté pour la méthodologie agile « Scrum » qui est beaucoup plus adaptable aux changements de fonctionnalités et qui est la méthode adoptée par les acteurs de la DSA.

3. Choix de la méthodologie SCRUM

SCRUM est un cadre ou canevas (Framework en anglais) qui est utilisé pour implémenter la méthode Agile de développement et de gestion de projet. Elle s'appuie sur le découpage de projet en itérations nommées « Sprint ». Un Sprint peut avoir une durée qui varie généralement entre deux (2) semaines et un (1) mois. Avant chaque Sprint les tâches sont estimées en temps et en complexité. Ces estimations permettent à la fois de planifier les

livraisons mais aussi d'estimer le coût de ces tâches auprès du client. Avec des livraisons fréquentes, le client reçoit un logiciel fonctionnel à chaque Sprint. Plus le projet avance, plus le logiciel est complet et possède toujours de plus en plus de fonctionnalités.

Scrum Process

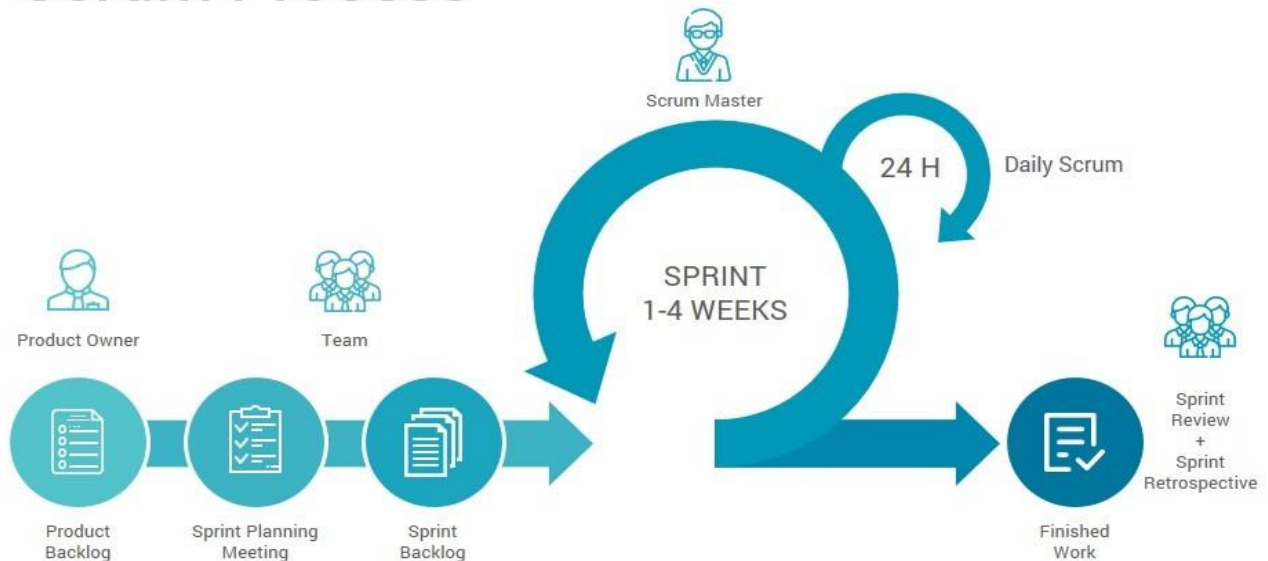


Figure 2 : Étapes de la méthode SCRUM

- **Le Product Backlog** est l'inventaire des tâches (fonctionnalités) à réaliser.
- **Le Sprint Backlogs** est l'ensemble des sous-tâches à effectuer pour réaliser une tâche.
- **Le Sprint Review** est défini comme un produit présentant suffisamment de fonctions aux utilisateurs finaux.

4. Équipe de Projet

Scrum définit trois (03) rôles :

- ✓ **Le « Product Owner »** : chargé de donner la vision des utilisateurs du système. Il s'agira ici de M. Antoine SAMPEBGO.

- ✓ **Le « Scrum Master »** : garant du respect du cadre méthodologique Scrum en aidant l'équipe à travailler de façon autonome. Ce rôle sera assuré par M. Dominique BAGRE.
- ✓ **L'équipe de développement** : qui réalise le produit. Cette tâche sera assurée par M. Souleymane ZONGO.
- ✓ M. Ousmane BARA, enseignant à l'UJKZ présent en tant que Superviseur.

5. Langage de Modélisation

Pour réaliser ce projet, nous avons utilisé le langage de modélisation unifié (UML) « Unified Modeling Language » qui est défini comme un outil de modélisation graphique. Il s'articule autour de quatorze (14) types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. C'est une représentation abstraite et simplifiée d'une entité qui pourrait être un processus, un phénomène, un objet du monde réel en vue de le décrire, de l'expliquer ou de le prévoir. De nos jours, UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet en offrant un standard logiciel pour la représentation de l'architecture logicielle.

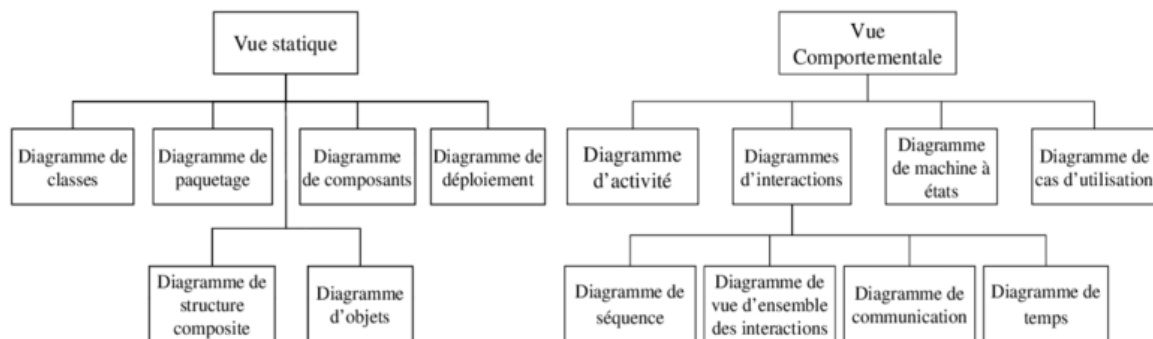


Figure 3 : Les diagrammes UML

Les diagrammes que nous allons présenter dans notre document sont ceux que nous avons jugés nécessaire à utiliser dans notre conception.

- Diagramme des cas d'utilisations ;
- Diagramme de séquences ;
- Diagramme d'activités ;
- Diagramme de classe ;

Ainsi, en accumulant UML et SCRUM, nous avons pu réaliser les différents diagrammes de modélisation nécessaire pour notre projet.

6. Planning prévisionnel

La réalisation d'un projet passe par l'élaboration d'une planification prévisionnelle bien définie en tenant compte des contraintes liées à l'organisation de la structure d'accueil. Ainsi dans notre cas, l'équipe de pilotage a opté pour le planning suivant :

- Prise en main de la méthodologie Scrum ;

Cette étape est relative à l'étude de la méthode agile Scrum avec ces concepts, les termes techniques ainsi que les différentes étapes de progression ;

- Choix des technologies ;

Ici, nous avons fait une étude comparative entre les différentes langages et Framework entrant dans le cadre de notre solution, l'objectif étant de choisir la technologie la plus adaptée à notre projet ;

- Rédaction du Product Backlog ;

Cette partie est indispensable car permet de relever les différentes fonctionnalités qu'exigent le client ;

- Conception du système ;

Ici, nous définissons le fonctionnement du système qui sera mis en place et sa faisabilité. Cette phase est ensuite représentée sous plusieurs formes de diagramme afin de faciliter sa compréhension au plus large public et en particulier les utilisateurs ;

- Sprint 0 ;

Le Sprint 0 concerne la mise en place des environnements de développements ainsi que l'installation des outils pouvant être utiles pour l'implémentation de notre solution. Cette partie prend en compte également la conception du système.

- Les autres Sprints ;

Cette dernière partie se concentre sur l'implémentation des différentes fonctionnalités traduites sous formes de code. Elle contient trois (3) Sprints qui seront testés afin de s'assurer du bon fonctionnement et de la conformité du système vis-à-vis des besoins exprimés.

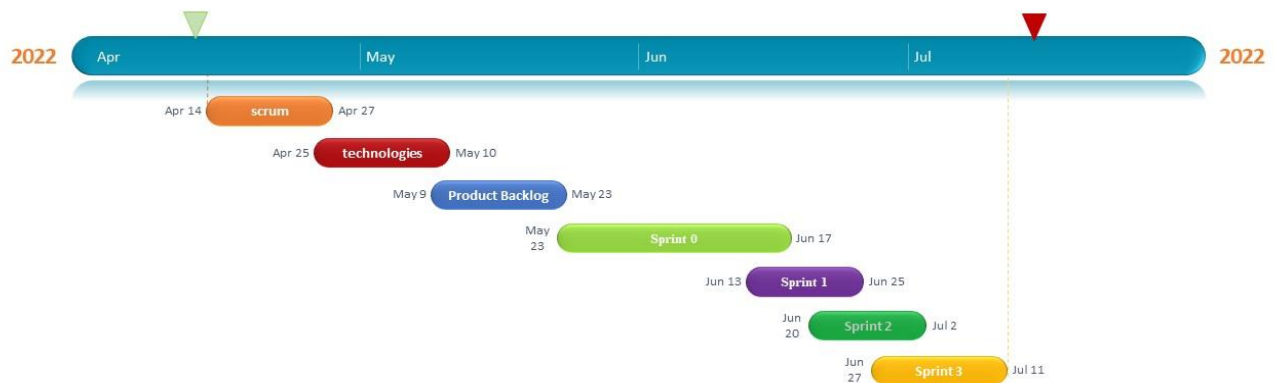


Figure 4 : Plan prévisionnel

II. Etude de l'existant

La gestion des frais de mission a pendant longtemps, été faite à travers le Circuit Intégré de la Dépense (CID). Pour des besoins de célérité et d'efficacité, la gestion des frais de mission comme bien d'autres dépenses telles que les frais d'abonnement aux journaux, ont été transférées aux régies d'avance. Aucune application métier n'a été déployée auprès des acteurs chargés de gestion des frais de mission. Certains ministères ont essayé de développer en interne des applications afin de mieux suivre les crédits budgétaires destinés aux missions. C'est le cas du Ministère de l'agriculture et des aménagements hydrauliques.

Ainsi, Afin de rationaliser les dépenses des structures dans la programmation des missions des établissements publics, L'État a décidé de concevoir un système intégré robuste, fiable et évolutif. C'est ainsi que la plateforme de Circuit Intégré de Mission (CIM) vit le jour. CIM est une application de gestion des missions adoptée dans l'administration Publique. Elle permet de gérer les missions. Il s'agit d'une application web dont certaines fonctionnalités ne sont pas adaptées à une utilisation mobile.

Cependant, au vu de la mobilité des agents de la fonction publique sur le terrain et dans la recherche de réactivité quant au traitement des missions, le développement d'une instance mobile s'impose.

CHAPITRE III : ANALYSE ET CONCEPTION DU SYSTEME

I. Domaine d'étude et fonctionnalités

1. Domaine d'étude

Grâce aux différentes analyses précédemment effectuées, notre travail s'est orienté sur :

- La gestion des Missions

Nos principales tâches seront de :

- consulter la liste des missions : les missions seront listées selon leur statut afin de faciliter la fluidité du système ;
- voir les détails d'une mission ;
- gérer les missions : c'est la fonction principale de notre application.

2. Fonctionnalités

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée / sortie du système. Ici nous avons trois (03) statuts d'utilisateurs à savoir :

- Un responsable de direction peut :
 - consulter ces différentes missions ;
 - approuver une mission ;
 - générer un rapport de mission ;
 - clôturer une mission ;
 - rejeter une mission ;
- Un responsable financier peut :
 - consulter ces différentes missions ;
 - budgétiser une mission ;
 - approuver une mission ;
 - payer une mission ;
 - vérifier une mission ;

- rejeter une mission ;
- Un ordonnateur peut :
 - valider une mission ;
 - rejeter une mission ;

II. Modélisation

1. Diagramme des cas d'utilisations

Le diagramme des cas d'utilisations est un modèle UML (Unified Modeling Language) qui représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, assurant la relation entre l'utilisateur et les objets via la mise en œuvre du système. Il est basé sur certains concepts tels que : acteur (utilisateur), cas d'utilisations (fonctionnalité) et les relations qui relient les acteurs, les cas d'utilisations ou les acteurs et les cas d'utilisations.

Les différents acteurs du système sont :

- le responsable de direction ;
- le responsable financier ;
- l'ordonnateur.

Les différents cas d'utilisations sont :

- s'authentifier ;
- consulter la liste des missions ;
- approuver une mission ;
- budgétiser une mission ;
- payer une mission ;
- vérifier une mission ;
- valider une mission ;
- clôturer une mission ;

Notre diagramme de cas d'utilisation pour la gestion des missions se présente comme suite :

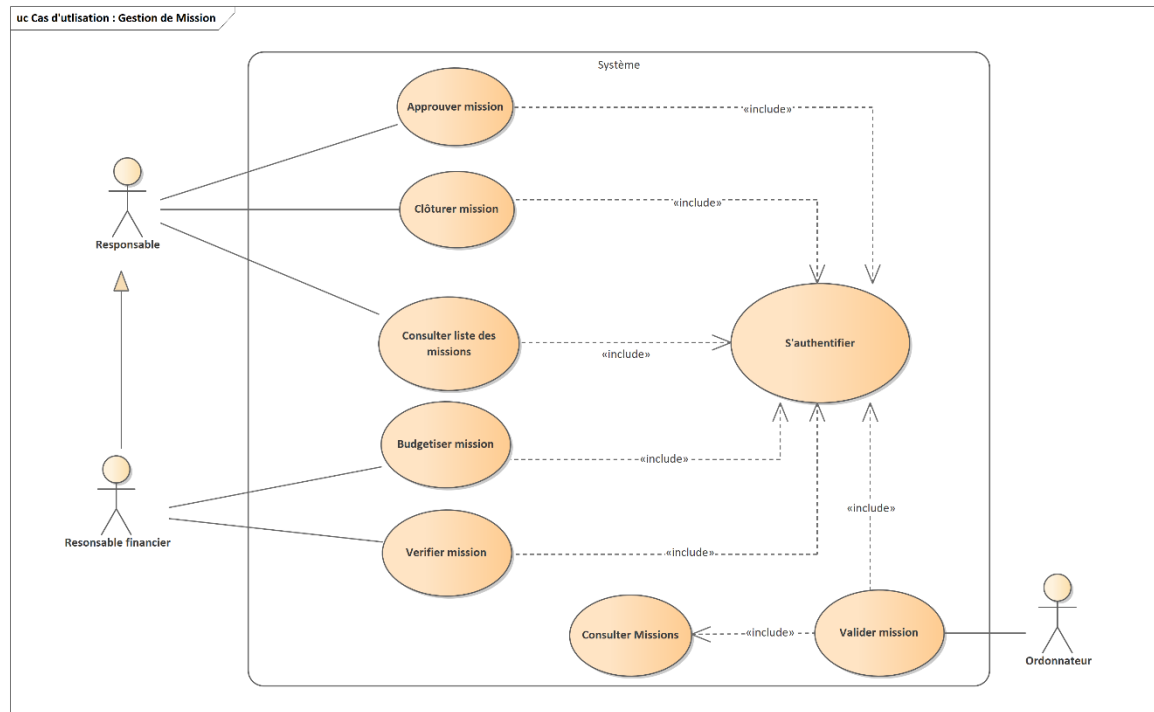


Figure 5 : Diagramme de cas d'utilisation

2. Description textuelle de quelques cas d'utilisations

Le diagramme de cas d'utilisation sert à décrire les grandes fonctions du système du point de vue des acteurs, mais ne donne pas d'explication clair et précis sur le dialogue entre les acteurs et les cas d'utilisation.

La description textuelle est une forme plus légère pour l'illustration d'un cas d'utilisation. Elle comporte trois (3) étapes dont les deux premières sont obligatoires et la dernière facultative à savoir :

- L'identification du cas d'utilisation,
- Le séquençement
- Les contraintes

◆ Description textuelle du Cas d'Utilisation (CU) « S'authentifier »

IDENTIFICATION	Nom : S'authentifier
	Objectif : Permettre la connexion au système
	Acteur : Utilisateur (principal), Système (secondaire)
	Date : 18/06/2022
	Responsable : Souleymane ZONGO
	Version : 1.0
SEQUENCEMENT	Ce cas d'utilisation commence à l'ouverture de l'Application
SCENARIO NOMINAL (SN)	<ol style="list-style-type: none"> 1) L'utilisateur saisit son identifiant et son mot de passe 2) Le système vérifie les Informations saisies 3) L'utilisateur affiche la page d'Accueil de l'application
SCENARIO D'EXCEPTION	<p style="text-align: center;">- Les Informations sont Incorrectes</p> <ul style="list-style-type: none"> ✓ Le système affiche un message d'erreur ✓ Le Scenario reprend au point 1
PRECONDITION	<ul style="list-style-type: none"> • L'utilisateur doit être enregistré dans la base de données et avoir un identifiant et un mot de passe préétabli
POSTCONDITION	<ul style="list-style-type: none"> ▪ Authentification réussie en cas d'informations correctes

Tableau 2 : CU S'authentifier

◆ Description textuelle du Cas d'Utilisation (CU) « Consulter liste des missions »

IDENTIFICATION	Nom : Consulter liste des missions
	Objectif : Permettre l'affichage la liste des missions
	Acteur : Utilisateur (principal), Système (secondaire)
	Date : 18/06/2022
	Responsable : Souleymane ZONGO
	Version : 1.0
SEQUENCEMENT	Ce cas d'utilisation commence lorsque l'utilisateur clique sur liste des missions (en cours, terminés, validés)
SCENARIO NOMINAL (SN)	<ol style="list-style-type: none"> 1) L'Utilisateur clique sur liste des missions 2) Le système récupère la liste des missions dans la base de données 3) Le système affiche la liste des missions 4) L'utilisateur choisit une mission 5) Le système récupère les informations de la mission 6) Le système affiche les informations de la mission
SCENARIO	- Liste de mission Vide

DÉVELOPPEMENT D'UNE APPLICATION MOBILE DE GESTION DES MISSIONS : CAS DU CIM/ANPTIC

ALTERNATIF	<ul style="list-style-type: none"> ✓ Le système retourne une liste Vide ✓ Le système n'affiche rien
PRECONDITION	<ul style="list-style-type: none"> • L'utilisateur doit disposer d'un compte • L'utilisateur doit être authentifié • L'utilisateur doit consulter Liste des missions
POSTCONDITION	<ul style="list-style-type: none"> ▪ Affichage de la liste des missions effectué avec succès

Tableau 3 : CU Liste des missions

◆ Description textuelle du Cas d'Utilisation (CU) « Approuver mission »

IDENTIFICATION	Nom : Approuver mission
	Objectif : Permettre au responsable de donner son approbation après vérification
	Acteur : Utilisateur (principal), Système (secondaire)
	Date : 10/06/2022
	Responsable : Souleymane ZONGO
	Version : 1.0
SEQUENCEMENT	Ce cas d'utilisation commence lorsque l'utilisateur clique sur approuver mission
SCENARIO NOMINAL (SN)	<ol style="list-style-type: none"> 1) Le responsable clique sur approuver mission 2) Le système récupère la liste des missions en attente d'approbation 3) Le système affiche la liste des missions à approuver 4) Le responsable choisit une mission 5) Le système récupère les informations de la mission 6) Le système affiche les informations de la mission 7) Le responsable approuve la demande de mission
SCENARIO ALTERNATIF	<ul style="list-style-type: none"> - <i>Liste de mission à approuver Vide</i> ✓ Le système retourne une liste Vide ✓ Le système n'affiche rien - <i>Rejeter mission à approuver</i> ✓ Le responsable rejette la mission avec un motif de refus ✓ La mission change de statut
PRECONDITION	<ul style="list-style-type: none"> • L'utilisateur doit être authentifié • L'utilisateur doit disposer d'un accès de responsable
POSTCONDITION	<ul style="list-style-type: none"> ▪ Mission approuvée avec succès et changement de statut

Tableau 4 : CU Approuver mission

DÉVELOPPEMENT D'UNE APPLICATION MOBILE DE GESTION DES MISSIONS : CAS DU CIM/ANPTIC

◆ Description textuelle du Cas d'Utilisation (CU) « Budgétiser mission »

IDENTIFICATION	Nom : Budgétiser mission
	Objectif : confirmation du budget de la mission par le responsable financier
	Acteur : Utilisateur (principal), Système (secondaire)
	Date : 20/06/2022
	Responsable : Souleymane ZONGO
	Version : 1.0
SÉQUENCEMENT	Ce cas d'utilisation commence lorsque l'utilisateur clique sur budgétiser mission
SCENARIO NOMINAL (SN)	<ol style="list-style-type: none"> 1) Le responsable financier clique sur approuver mission 2) Le système récupère la liste des missions en attente de budgétisation 3) Le système affiche la liste des missions à budgétiser 4) Le responsable financier choisit une mission 5) Le système récupère les informations de la mission 6) Le système affiche les informations de la mission 7) Le responsable financier budgétise la mission
SCENARIO ALTERNATIF	<ul style="list-style-type: none"> - <i>Liste de mission à budgétiser Vide</i> <ul style="list-style-type: none"> ✓ Le système retourne une liste Vide ✓ Le système n'affiche rien - <i>Rejeter mission à budgétiser</i> <ul style="list-style-type: none"> ✓ Le responsable rejette la mission avec un motif de refus ✓ La mission change de statut
PRECONDITION	<ul style="list-style-type: none"> • L'utilisateur doit être authentifié • L'utilisateur doit disposer d'un accès de responsable financier
POSTCONDITION	<ul style="list-style-type: none"> ▪ Mission budgétisée avec succès et changement de statut

Tableau 5 :CU budgétiser mission

◆ Description textuelle du Cas d'Utilisation (CU) « Valider mission »

IDENTIFICATION	Nom : Valider mission
	Objectif : Permettre l'exécution et le démarrage de la mission
	Acteur : Utilisateur (principal), Système (secondaire)
	Date : 20/06/2022
	Responsable : Souleymane ZONGO
	Version : 1.0
SEQUENCEMENT	Ce cas d'utilisation commence lorsque l'utilisateur clique sur valider mission

DÉVELOPPEMENT D'UNE APPLICATION MOBILE DE GESTION DES MISSIONS : CAS DU CIM/ANPTIC

SCENARIO NOMINAL (SN)	<ol style="list-style-type: none"> 1) L'ordonnateur clique sur valider mission 2) Le système récupère la liste des missions en attente de validation 3) Le système affiche la liste des missions à valider 4) Le responsable choisit une mission 5) Le système récupère les informations de la mission 6) Le système affiche les informations de la mission 7) Le responsable valide la demande de mission
SCENARIO ALTERNATIF	<ul style="list-style-type: none"> - <i>Liste de mission à valider Vide</i> ✓ Le système retourne une liste Vide ✓ Le système n'affiche rien - <i>Rejeter mission à valider</i> ✓ Le responsable rejette la mission avec un motif de refus ✓ La mission change de statut
PRECONDITION	<ul style="list-style-type: none"> • L'utilisateur doit être authentifié • L'utilisateur doit disposer d'un accès d'ordonnateur
POSTCONDITION	<ul style="list-style-type: none"> ▪ Mission validée avec succès et changement de statut

Tableau 6 : CU valider mission

◆ Description textuelle du Cas d'Utilisation (CU) « Clôturer mission »

IDENTIFICATION	Nom : Clôturer mission
	Objectif : Permettre de mettre fin à d'une mission
	Acteur : Utilisateur (principal), Système (secondaire)
	Date : 21/06/2022
	Responsable : Souleymane ZONGO
	Version : 1.0
SEQUENCEMENT	Ce cas d'utilisation commence lorsque l'utilisateur clique sur liste des missions (en cours, terminés, validés)
SCENARIO NOMINAL (SN)	<ol style="list-style-type: none"> 1) Le responsable clique sur clôturer mission 2) Le système récupère la liste des missions à clôturer 3) Le système affiche la liste des missions à clôturer 4) Le responsable choisit une mission 5) Le système récupère les informations de la mission 6) Le système affiche les informations de la mission 7) Le responsable clôture la mission
SCENARIO ALTERNATIF	<ul style="list-style-type: none"> - <i>Liste de mission Vide</i> ✓ Le système retourne une liste Vide ✓ Le système n'affiche rien
PRECONDITION	<ul style="list-style-type: none"> • L'utilisateur doit disposer d'un compte • L'utilisateur doit être authentifié
POSTCONDITION	<ul style="list-style-type: none"> ▪ Mission Clôturée avec succès et Changement de statut

Tableau 7 : CU clôturer mission

3. Diagramme de séquence de quelques cas d'utilisation

Le Diagramme de séquence est un graphe montrant les interactions entre les objets en mettant l'accent sur l'aspect temporel, c'est-à-dire la chronologie des envois de messages. Il permet de modéliser :

- Les scenarios d'utilisations : une façon d'utiliser le système ;
- La logique des méthodes : en parti dans le cas des opérations complexes ;
- La logique des services : Méthodes de haut niveau invocable par plusieurs variétés de clients

◆ Diagramme de Séquence du Cas d'Utilisation (CU) « S'authentifier »

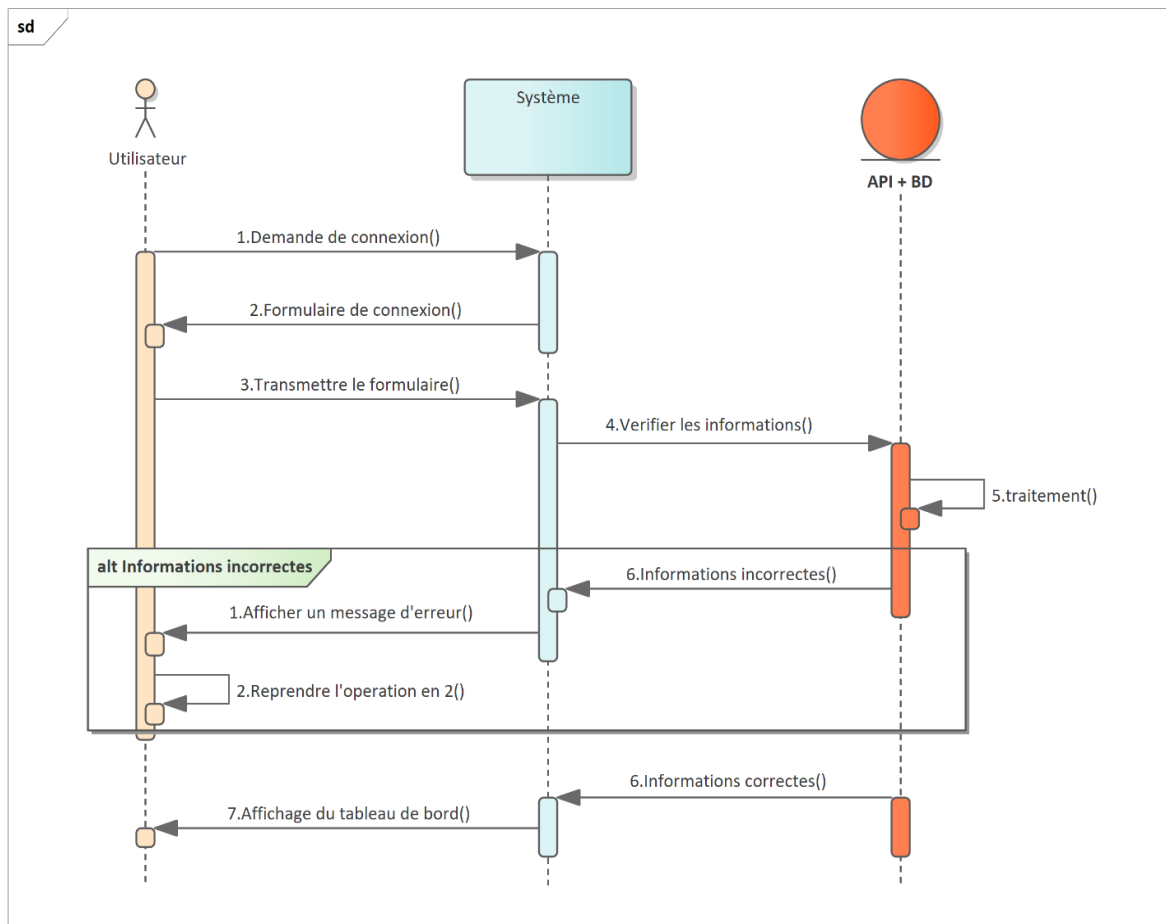


Figure 6 : diagramme de séquence « S'authentifier »

◆ Diagramme de Séquence du Cas d'Utilisation (CU) « Consulter liste des missions »

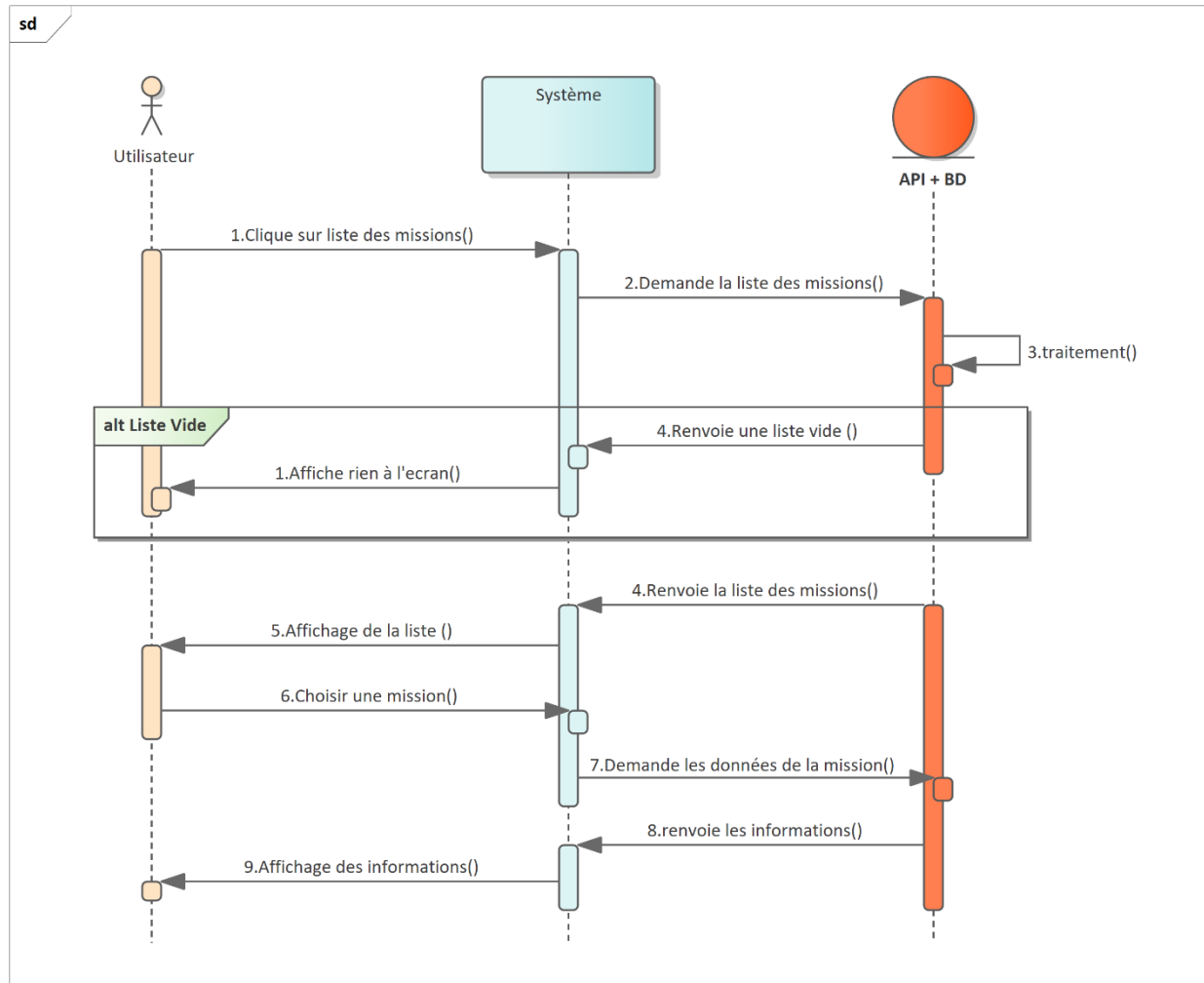


Figure 7 : Diagramme de séquence de « Consulter liste des missions »

◆ Diagramme de Séquence du Cas d'Utilisation (CU) « Approuver mission »

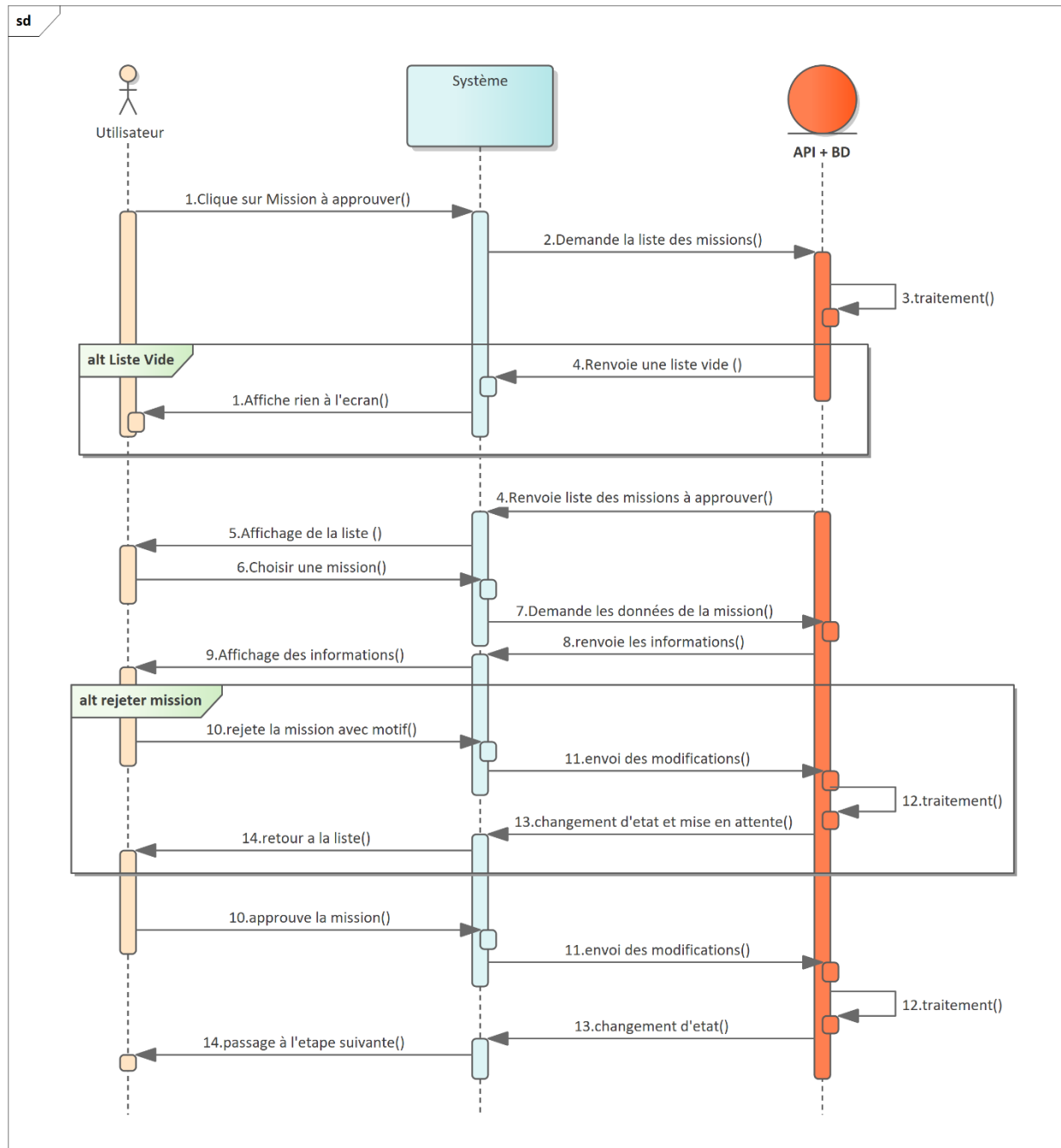


Figure 8 : Diagramme de séquence de « Approuver mission »

◆ Diagramme de Séquence du Cas d'Utilisation (CU) « Budgétiser mission »

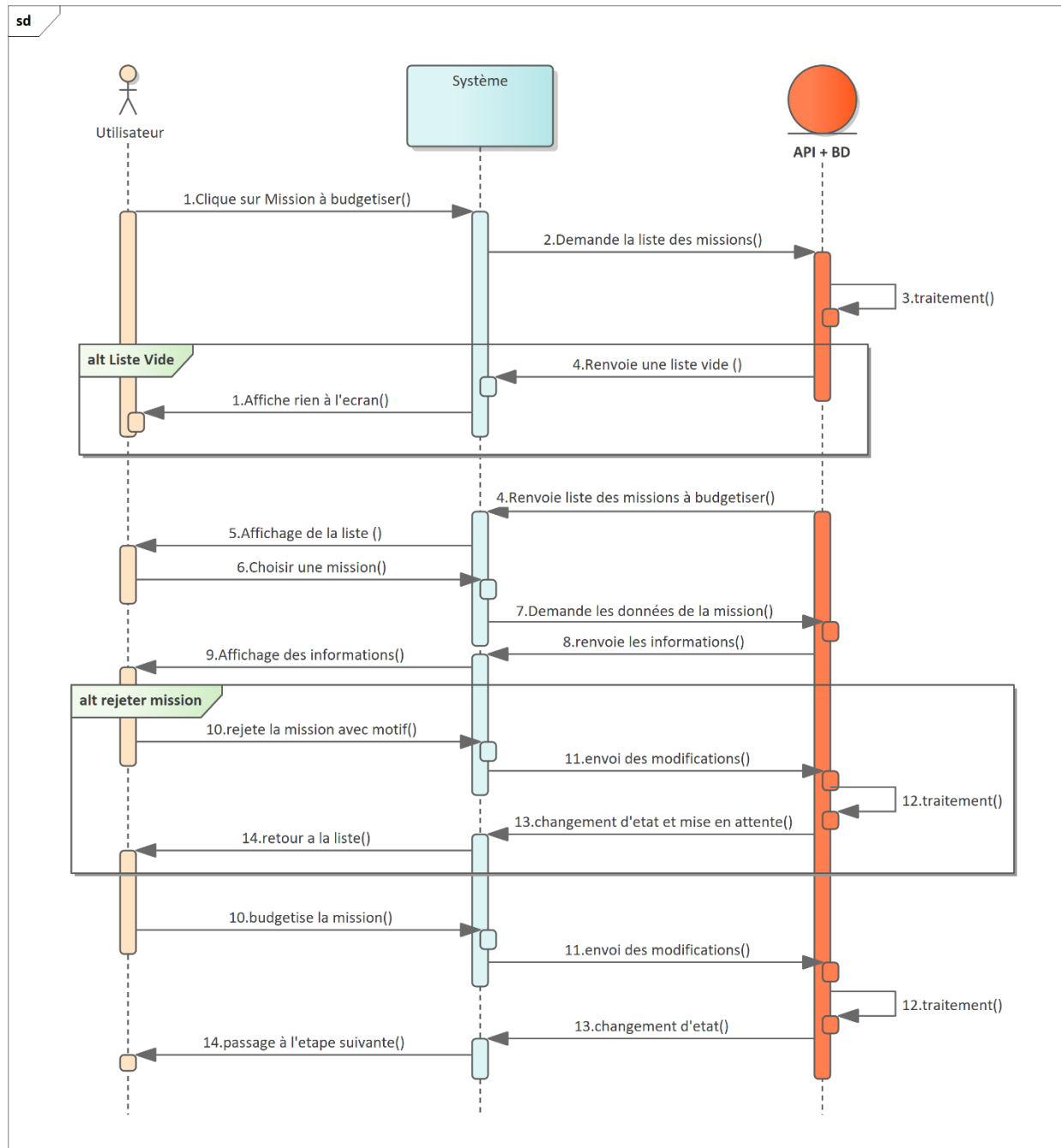


Figure 9 : Diagramme de séquence de « Budgétiser mission »

◆ Diagramme de Séquence du Cas d'Utilisation (CU) « Valider mission »

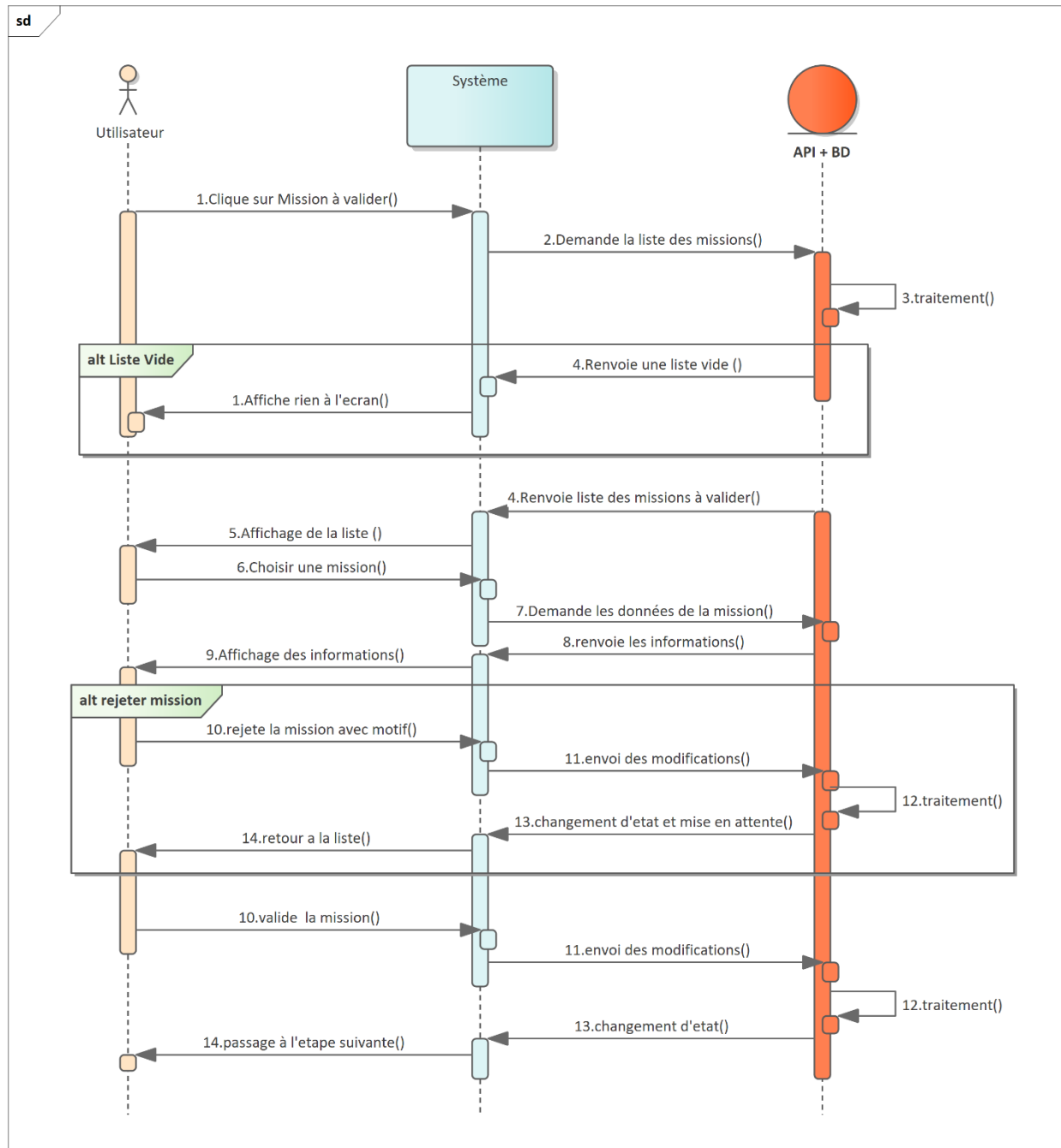


Figure 10 : Diagramme de séquence de « Valider mission »

◆ Diagramme de Séquence du Cas d'Utilisation (CU) « Clôturer mission »

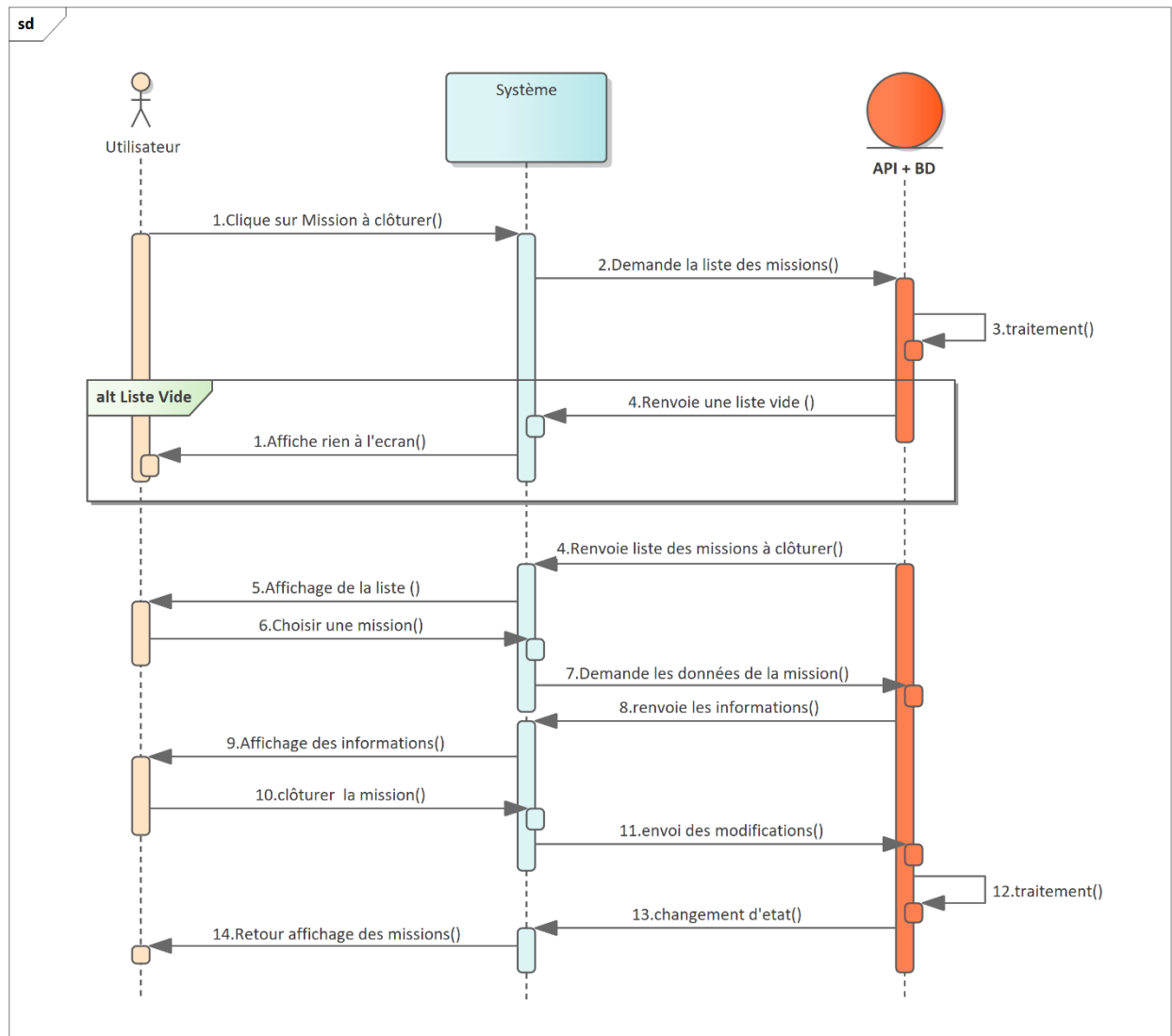


Figure 11 : Diagramme de séquence de « Clôturer mission »

4. Diagramme d'activité

Les diagrammes d'activités permettent de décrire les traitements. C'est le diagramme le plus adapté à la modélisation du cheminement de flots de contrôle et de flots de données, car il permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. Les diagrammes d'activités viennent illustrer et consolider la description textuelle des cas d'utilisation en donnant une description plus précise des traitements.

Dans notre cas, nous allons représenter le diagramme global d'activité de notre application.

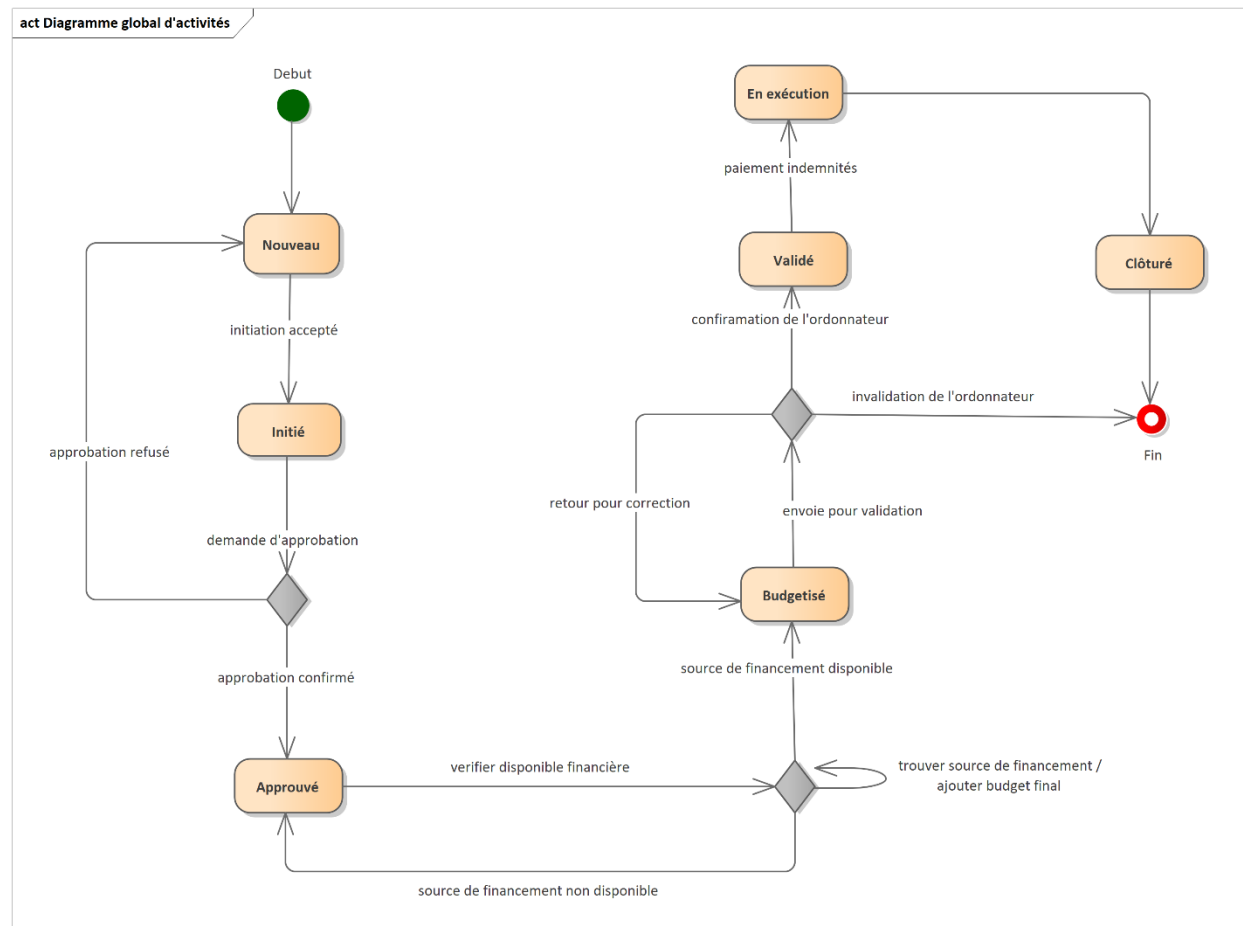


Figure 12 : Diagramme global d'activité

5. Diagrammes de classes

Le diagramme de classe est le diagramme le plus important de la modélisation orientée objet. Il est le seul obligatoire lors de la modélisation d'un système. Il montre la structure interne d'un système en fournissant une représentation abstraite des objets du système qui vont interagir ensemble pour la réalisation des cas d'utilisations. C'est un graphe montrant les classes du système, les relations entre elles ainsi que les attributs et les opérations de ces classes. Le diagramme de classe est utilisé généralement pour explorer les concepts du domaine, analyser les besoins et pour définir la conception détaillée du système.

DÉVELOPPEMENT D'UNE APPLICATION MOBILE DE GESTION DES MISSIONS : CAS DU CIM/ANPTIC

Le graphique suivant représente le diagramme de classes :

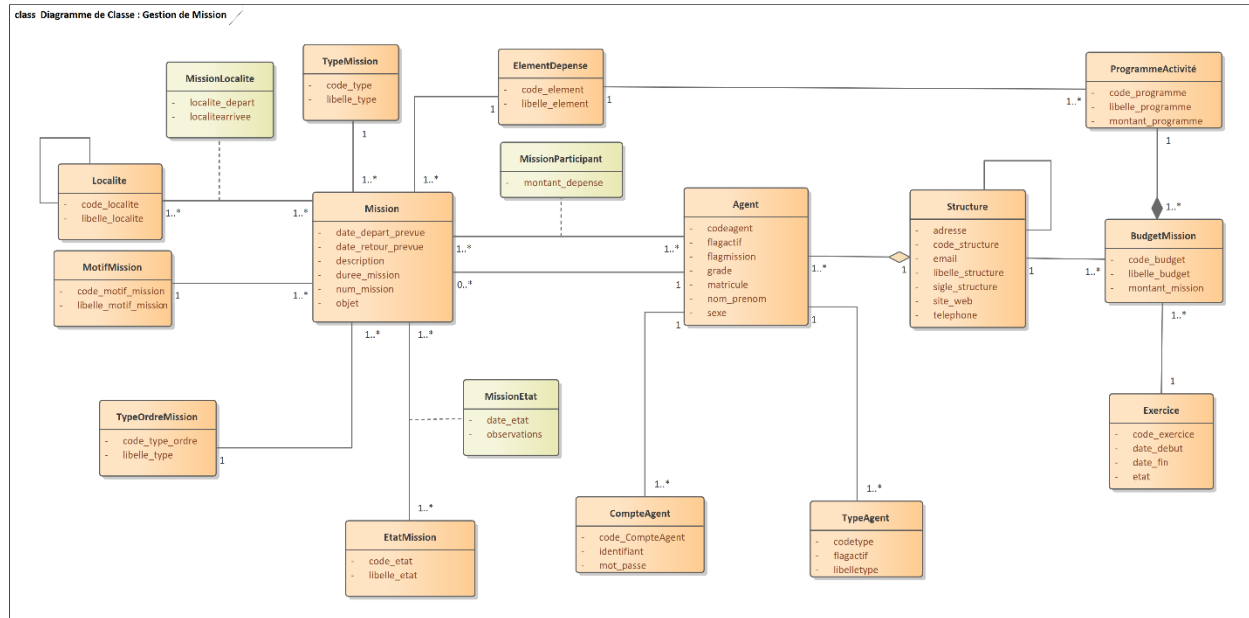


Figure 13 : Diagramme de classe

CHAPITRE IV : MISE EN ŒUVRE DE L'APPLICATION

I. Outils technologiques

1. Applications mobiles et systèmes d'exploitation

On appelle Application mobile tout logiciel applicatif conçu et développé pour fonctionner sur les appareils électroniques mobiles tels que les smartphones ou les tablettes dotés d'un système d'exploitation mobile.

❖ **Système d'exploitation mobile**

Les systèmes d'exploitation mobiles (OS) peuvent être définis comme un ensemble de logiciels permettant à un smartphone ou un téléphone mobile basique de fonctionner. Ils permettent aux utilisateurs de passer des appels téléphoniques, de naviguer sur leurs téléphones, de télécharger des applications ou encore de paramétrer et personnaliser leurs smartphones. On distingue une vingtaine de système d'exploitation sur le marché dont les plus populaires sont :

➤ Android de Google

Android est considéré comme l'OS de référence du marché. Créé en 2004 par une startup puis racheté par Google en 2005, Android est fondé sur le noyau Linux et a été lancé officiellement 2007 sous la responsabilité de l'Open Handset Alliance (OHA). Il est à sa version **12.0.0** à la date de rédaction de ce document.



Figure 14 : logo Android

➤ iOS d'Apple

iOS anciennement appelé iPhone OS, est le système d'exploitation mobile développé par Apple pour plusieurs de ses appareils. Avec Android, il occupe une très grande majorité du marché des smartphones de la planète. Sa version actuelle est **15.5** à la date de rédaction de ce document.



Figure 15 : logo iOS

➤ Windows Phone de Microsoft

C'est un système d'exploitation mobile développé par Microsoft pour succéder à Windows Mobile. Son interface et ses fonctionnalités le rapprochent plus du système d'exploitation Windows. Il a vu le jour en octobre 2010 et en novembre 2015, il se voit remplacer par Windows 10 mobile.



Figure 16 : logo Windows phone

➤ BlackBerry OS de RIM

Ancienne star de la téléphonie, prisé par le monde professionnel, l'OS BlackBerry de RIM a perdu ses parts du marché depuis la dernière décennie au profit des géants que sont Android et iOS. Lancé en 1999, il est remplacé par une version Unix du système d'exploitation. L'un des plus grands aspects de ce système est sa sécurité. BlackBerry quitte officiellement le marché de la téléphonie en février 2022 en vendant ses brevets de téléphonies mobile.



Figure 17 : BlackBerry OS

2. Application native et Cross-platforms

2.1 Application native

Une application native est développée spécifiquement pour une plateforme Android ou iOS etc....De ce fait, elle utilise des logiciels et du matériel spécifique à l'appareil. Ainsi pour toucher un plus grand nombre de mobinaute, il convient de multiplier les applications natives sous différents systèmes d'exploitation car une application formulée pour iOS ne fonctionnera pas sur Android. Le langage de programmation est différent : Java ou Kotlin pour Android et Swift ou Objective-C pour iOS.

○ **Avantages**

iPhone ou téléphone Android, l'application native reste accessible et offre une expérience de navigation optimale. Les Applications natives sont plus performantes, plus rapides et plus sécurisées tout en offrant un large éventail de fonctionnalités.

- **Inconvénients**

Les applications natives doivent avoir plusieurs bases de code, car chaque appareil inclut sa version de l'application entraînant ainsi une élévation du coût du matériel et du développement. Leur flexibilité est limitée par rapport aux applications hybrides.

2.2 Application cross-platform

Le développement des applications cross-platform se fait à l'aide d'un Framework qui est généralement constitué d'un ensemble d'outils. Contrairement aux applications natives, les applications cross-platform sont des applications développées une seule fois avec un seul et unique code source pour plusieurs plateformes telles qu'Android ou iOS. Parmi les Framework les plus connus et utilisés, on retrouve [React Native](#), [Ionic](#), [Flutter](#), [PhoneGap](#), [Titanium](#) et [Xamarin](#). On n'utilise pas de langage natif pour le développement mais plutôt des langages simples tels que Javascript, C#, Dart etc...

- **Avantages**

La réutilisation du code est probablement le principal avantage des frameworks permettant de déployer l'application sur plusieurs plateformes. Ce qui permet de trouver un bon équilibre entre qualité et coût avec un gain de temps considérable.

- **Inconvénients**

Les applications mobiles multiplateformes sont réputées pour être moins performantes et moins fluides que les applications natives. Il y a souvent des différences d'affichage et de comportement potentiel selon la plateforme d'exécution sans oublier les ajustements complexes à faire dans le code en vue de l'adaptation de l'application à chaque plateforme. On les reproche également d'avoir un rendu plus proche web que du mobile.

2.3 Choix du type d'application

Au Burkina Faso, Android et iOS sont les plateformes mobiles les plus utilisées. De ce fait et au vu de leur énormes progrès, il nous incombe de diriger notre choix vers une application cross-platform.

3. Framework

3.1 Qu'est-ce qu'un Framework

Un Framework (appelé aussi infrastructure logicielle en français) désigne en programmation informatique un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application. Il a pour objectif de simplifier le travail des développeurs en informatique tout en optimisant et en améliorant leur productivité.

Les principaux avantages sont donc :

- La réutilisation des codes
- La standardisation de la programmation
- La formalisation d'une architecture adaptée aux besoins de chaque entreprise.

3.2 Étude comparative des Framework

3.2.1 Les différents Framework

◆ Flutter

Présenté officiellement par Google en décembre 2018, Flutter est un Framework d'application multiplateforme open source. Il est spécialisé dans le développement des applications et offre aux développeurs un moyen simple de créer et de déployer des applications visuellement attrayantes, compilées en mode natif destinées aux téléphones mobiles (Android, iOS), le web et les ordinateurs de bureau (desktop), le tout à partir d'un seul code. Il est basé sur Dart, un langage de programmation rapide et orienté objet sorti en 2011 et est facile à apprendre. Flutter fournit ses propres widgets, tirés de son propre moteur de rendu haute performance : ces widgets sont rapides, attrayantes et bien sûr personnalisables. Grâce à ces widgets et à son moteur de rendu visuel [SKIA](#), les applications Flutter ont un bel aspect et une convivialité appréciée.

Il est maintenu par Google et la communauté Flutter.



Figure 18 : logo de Flutter

◆ React Native

React Native est un Framework JavaScript créé en 2015 permettant d'écrire de véritables applications mobiles de rendu natif pour iOS et Android. Il est basé sur React, la bibliothèque JavaScript de Facebook pour la création d'interfaces utilisateurs, mais au lieu de cibler le navigateur, il cible les plateformes mobiles. Les développeurs Web peuvent désormais écrire des applications mobiles qui semblent et se sentent vraiment "natives", le tout dans le confort d'une bibliothèque JavaScript. Les applications React Native sont écrites à l'aide d'un mélange de balisage JavaScript et XML, connu sous le nom de JSX.

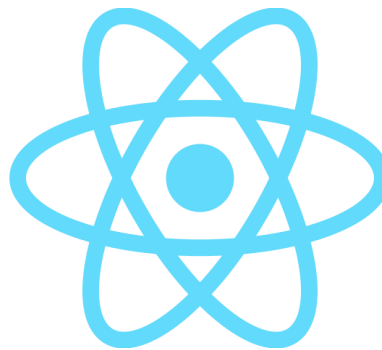


Figure 19 : logo de React Native

◆ Ionic

Ionic est un SDK open source complet pour le développement d'applications mobiles hybrides créé en 2013. Ionic fournit des outils et des services pour développer des applications mobiles hybrides à l'aide de technologies Web telles que CSS, HTML5 et Sass. Les applications peuvent être créées avec ces technologies Web, puis distribuées via des magasins d'applications natifs pour être installées sur des appareils en tirant parti de Cordova.



Figure 20 : logo de Ionic

3.2.2 Etude comparée

Ici nous essaierons de comparer les frameworks Flutter, React Native et Ionic. Les trois (3) sont d'excellents frameworks.

Framework	Avantages	Inconvénients
React native	<ul style="list-style-type: none">- Portée par Facebook donc Grande communauté- Flexible- Vue native- Gratuit	<ul style="list-style-type: none">- Moins sécurisé- Temps d'exécution long pour le rendu- API trop légère- Prise en main un peu difficile

Ionic	<ul style="list-style-type: none">- Développement rapide- Utilisation des technologies développement web standard- Développé avec Javascript et Angular- Gratuit	<ul style="list-style-type: none">- Vue web- Problème de sécurité- Débogage difficile
Flutter	<ul style="list-style-type: none">- Facile à apprendre et à déboguer- Entièrement compilé- Utilisation du Hot Reload- Dispose d'un ensemble de widgets de style d'Android de Google et d'Apple- Communauté active et grandissante- Gratuit	-ne fonctionne pas sur tous les appareils (tv OS, Watch OS)

3.2.3 Choix du Framework

Au vu de tous ses caractéristiques et avantages, notre choix s'est dirigé vers le Framework Flutter de Google.

4. Langage informatique

Un langage de programmation est un ensemble de notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques. Composée d'un alphabet, d'un vocabulaire, de règles de grammaire et de significations, il rend sa syntaxe compréhensible par la machine grâce un environnement de traduction intégré.

Les langages de programmation utilisés dans notre projet sont :

➤ Dart

Développé par Google en 2013, Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est utilisé pour créer des applications mobiles, de bureau, de serveur et web. Dart est un langage orienté objet à ramasse-miettes avec une syntaxe de type C++. Il peut se compiler en code natif ou en JavaScript.



Figure 21 : le logo de Dart

➤ Node.js et Express.js

Node.js est une plateforme logicielle libre en JavaScript orientée vers les applications réseau évènementielles. Elle utilise la machine virtuelle [V8](#), la librairie [libuv](#) pour sa boucle d'évènements. Elle a été créée en 2009 par l'entreprise Joyent.

Express.js est un Framework pour construire des applications web basée sur Node.js. C'est le Framework standard pour le développement de serveur en Node.js. Il est relativement minimaliste en ressource tout en permettant d'étendre ses fonctionnalités via des plugins.



Figure 22 : logo de Node.js



Figure 23 : logo de Express.js

➤ SQL

SQL (Structured Query Language) est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.



Figure 23 : logo de SQL

➤ API

Une API (Application Programming Interface) est un ensemble de services et de protocoles utilisés par une application permettant à celle-ci de communiquer avec une autre application. Il existe deux (2) type de grandes tendances en matière de style architectural d'API que sont : SOAP et REST.

Pour notre projet, nous avons opté pour le style REST qui est le plus populaire et le plus adapté à notre solution vu son approche orientée donnée qui utilise des requêtes HTTP.



Figure 24 : logo API

5. Environnement matériel

Nous avons utilisé deux (02) environnements matériels pour le développement de notre projet.

➤ Ordinateur Portable

L'appareil utilisé pour réaliser notre projet est un ordinateur HP équipé d'un processeur *Intel(R) Core (TM) i3-3110M CPU @ 2.40GHz 2.40GHz* avec une mémoire RAM de huit (8,00) Go et un système d'exploitation Windows 10 Entreprise version 20H2.

Spécifications de l'appareil

Nom de l'appareil	ISSA-ZONGO
Processeur	Intel(R) Core(TM) i3-3110M CPU @ 2.40GHz 2.40 GHz
Mémoire RAM installée	8,00 Go (7,89 Go utilisable)
ID de périphérique	1E0222AD-8CFE-4522-B67A-52065A4C7CD8
ID de produit	00329-00000-00003-AA269
Type du système	Système d'exploitation 64 bits, processeur x64
Stylet et fonction tactile	La fonctionnalité d'entrée tactile ou avec un stylet n'est pas disponible sur cet écran

Copier

Renommer ce PC

Spécifications de Windows

Édition	Windows 10 Entreprise
Version	20H2
Installé le	16/04/2022
Build du système d'exploitation	19042.1766
Expérience	Windows Feature Experience Pack 120.2212.4180.0

Figure 25 : Informations sur l'ordinateur utilisé

➤ Smartphone

Nous avons utilisé un smartphone de marque Tecno K7 pour nos différents tests de l'application. Il est doté d'un système d'exploitation Android 10 avec une capacité de stockage de 32 GB et d'une mémoire RAM de 2 Go.

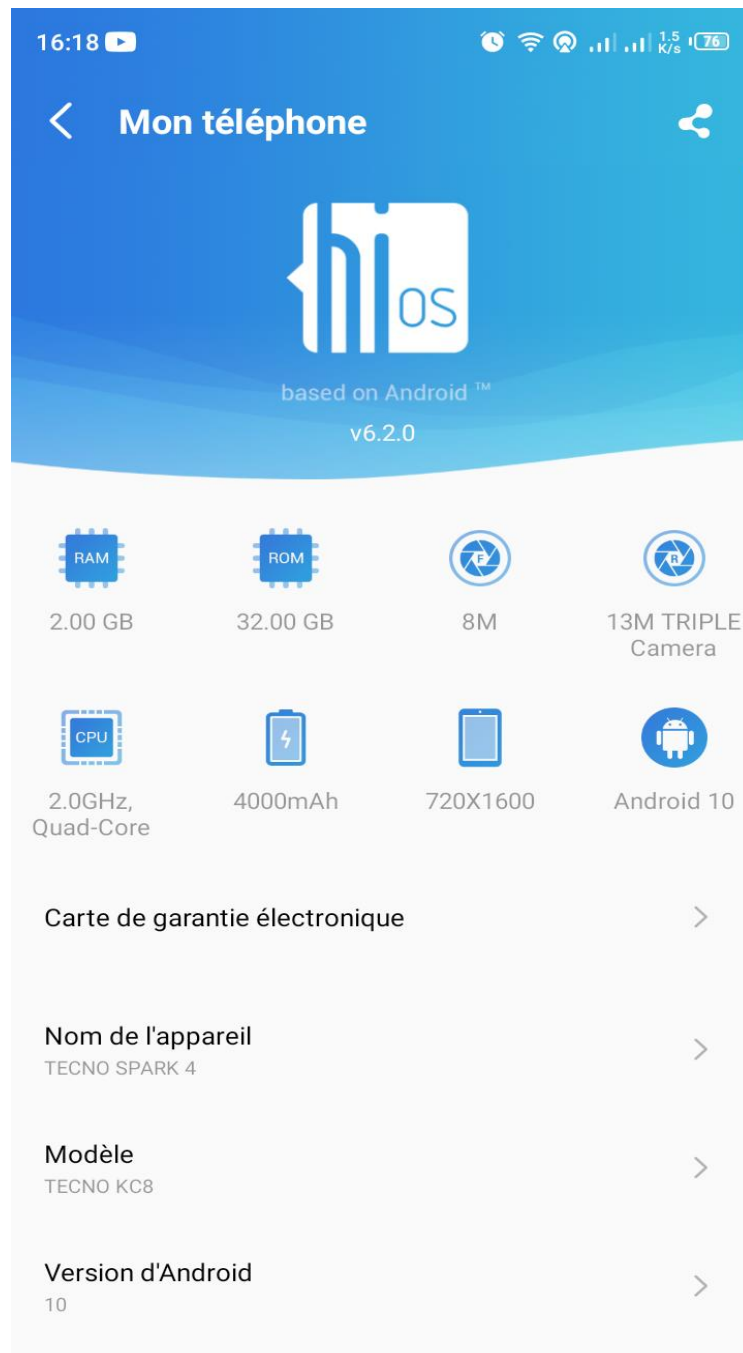


Figure 26 : Informations du système du Smartphone utilisé

6. Environnement logiciel

➤ Visual Studio Code

C'est un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence des syntaxes, la complétion intelligente du code, la refactorisation du code et Git intégré. Les utilisateurs peuvent personnaliser l'interface en modifiant le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

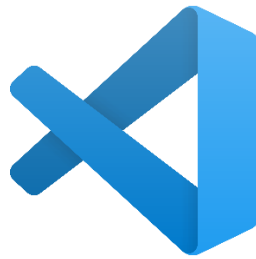


Figure 27 : logo de VSCode

➤ Enterprise Architect

Enterprise Architect est un logiciel de modélisation et de conception UML, édité par la société australienne Sparx Systems. Couvrant par ses fonctionnalités l'ensemble des étapes du cycle de conception d'application, il est l'un des logiciels de conception et de modélisation les plus reconnus.



Figure 28 : logo de Enterprise Architect

➤ Git

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvald, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2.



Figure 29 : logo de Git

➤ GitHub

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Le site assure également un contrôle d'accès et des fonctionnalités à la collaboration comme le suivi des bugs, les demandes de fonctionnalités pour chaque projet.



Figure 30 : logo de GitHub

➤ Trello

Trello est un outil de gestion de projet en ligne, lancé en septembre 2011 et inspiré par la méthode Kanban de Toyota. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches.



Figure 31 : logo de Trello

➤ PostgreSQL 11

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD. Ce système est comparable à d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MariaDB), ou propriétaires (comme Oracle, Microsoft SQL Server).

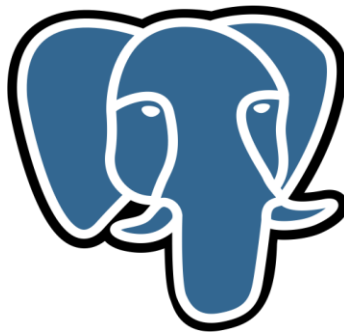


Figure 32 : logo de PostgreSQL

➤ Postman

Postman est une application permettant de tester des API, créée en 2012 pour répondre à une problématique de test d'API partageable. D'abord module complémentaire de Google Chrome, puis client lourd, et finalement client léger, elle est à présent utilisée par plus de 500 000 entreprises dans le monde et son siège se trouve à San Francisco.



Figure 33 : logo de Postman

II. Estimation des Coûts

1. Méthode COCOMO et coût de prestation de l'équipe

Pour l'évaluation du coût de prestation de l'équipe, nous utiliserons la méthode COCOMO. La méthode CONstructive COSt MOdel (COCOMO) a été proposée par Barry. W. Boehm en 1981 en fonction des hypothèses suivantes :

- Il est facile pour un informaticien d'estimer le nombre de ligne du code source.
- La complexité d'écriture d'un programme est la même quel que soit le langage de programmation.

Cette méthode permet d'obtenir la charge de réalisation en Mois/Homme et le délai normal recommandé. Nous avons les formules de calculs suivants :

- Charge en Mois/Homme = $a [Kisl]^b$, avec : Kisl = kilo instruction source livrée
- Délai = $c [Charge]^d$
- Taille moyenne d'équipe = Charge / Délai

Les paramètres (a, b, c et d) dépendent de la catégorie du projet.

Soit I la taille du projet, on a :

- Un projet simple si $I < 50$ Kisl, spécifications stables, petite équipe ;
- Un projet moyen si $50 \text{ Kisl} \leq I < 300$ Kisl, spécifications stables, petite équipe ;
- Un projet complexe si $I > 300$ Kisl, grande équipe.

Type de projet	Charge en mois-personne	Délais en mois
Simple	$\text{Charge} = 2,4(Kisl)^{1,05}$	$D = 2,5(\text{charge})^{0,38}$
Moyen	$\text{Charge} = 3 (Kisl)^{1,12}$	$D = 2,5 (Charge)^{0,35}$
Complexe	$\text{Charge} = 3,6 (Kisl)^{1,2}$	$D = 2,5 (Charge)^{0,32}$

Tableau 8 : Model COCOMO

Pour ce projet, nous estimons Kisl à 7 et à 300 00 FCFA comme salaire moyen d'un informaticien au Burkina Faso. Ce qui donne :

DÉVELOPPEMENT D'UNE APPLICATION MOBILE DE GESTION DES MISSIONS : CAS DU CIM/ANPTIC

- Charge en Mois/Homme : $2,4 \times (7)^{1,05} = 18,51$
- Délai normal en mois : $2,5 \times (10,45)^{0,38} = 7,57 \approx 7$ mois
- Taille moyenne de l'équipe : $18,51 / 7,57 = 2.45 \approx 3$ personnes

D'après nos calculs, il faudrait une équipe de 3 personnes pendant 7 mois pour réaliser ce projet.

- Coût des ressources humaines : $300\,000 \times 7 \times 3 = 6\,300\,000$

2. Coût du matériel de développement

La conception de notre application a nécessité un cout en matériels énumérés dans le tableau suivant :

Matériels	Description/Quantité	Coût (en F CFA)
Ordinateur Portable	Core i7 16Go Ram SSD 250 et DDR / 01	500.000
Smartphone Android	8Go Ram 32Go ROM /01	150.000
iPhone	8Go Ram 32Go ROM /01	250.000
Smartphone Tablette	8Go Ram 32Go ROM /01	200.000
Total	04	1.100.000

3. Coût du matériel de déploiement

Description	Quantité	Coût (en F CFA)
Serveur physique	01	00
Switch	01	00
Onduleur	01	00
Maintenance	01	300.000 / mois

❖ **Estimation Coût Total :** *Coût de développement + Coût du matériel + Cout du déploiement*

$6.300.000 + 1.100.000 + (300.000 \times 7) = 9\,500\,000$ FCFA (neuf millions cinq cent mille).

III. Mise en œuvre

1. Page de connexion

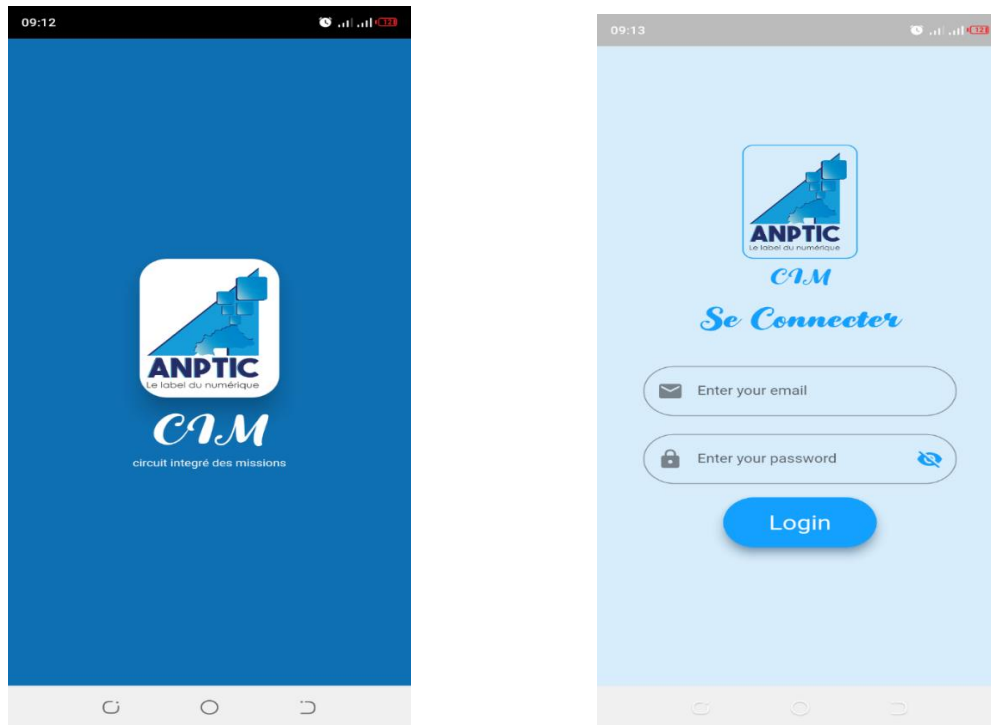


Figure 34 : page de connexion

2. Page d'accueil

DÉVELOPPEMENT D'UNE APPLICATION MOBILE DE GESTION DES MISSIONS : CAS DU CIM/ANPTIC

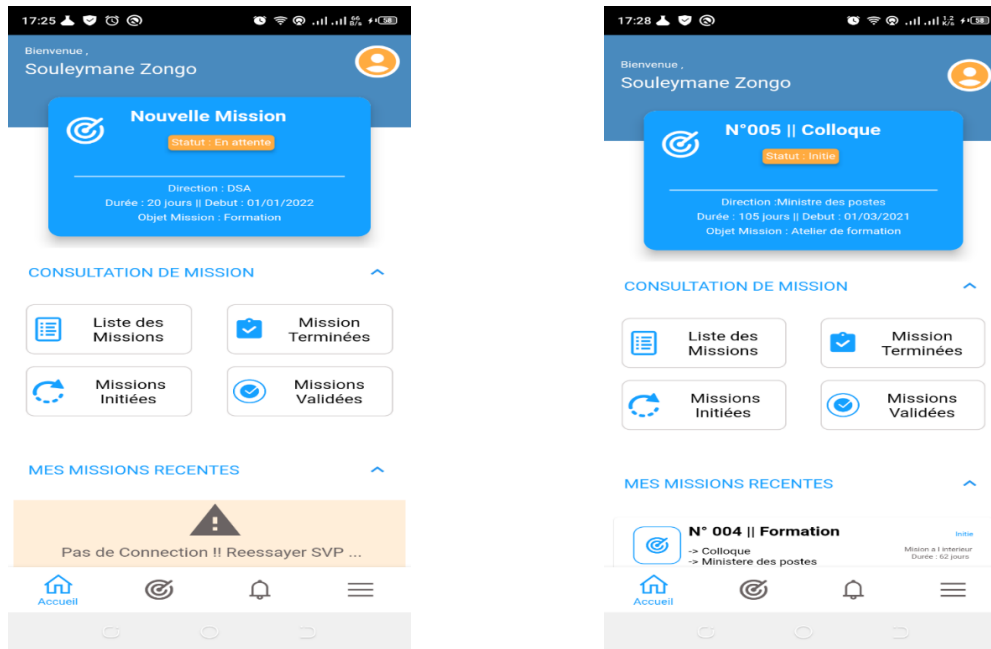


Figure 35 : page d'accueil

3. Page de gestion des missions

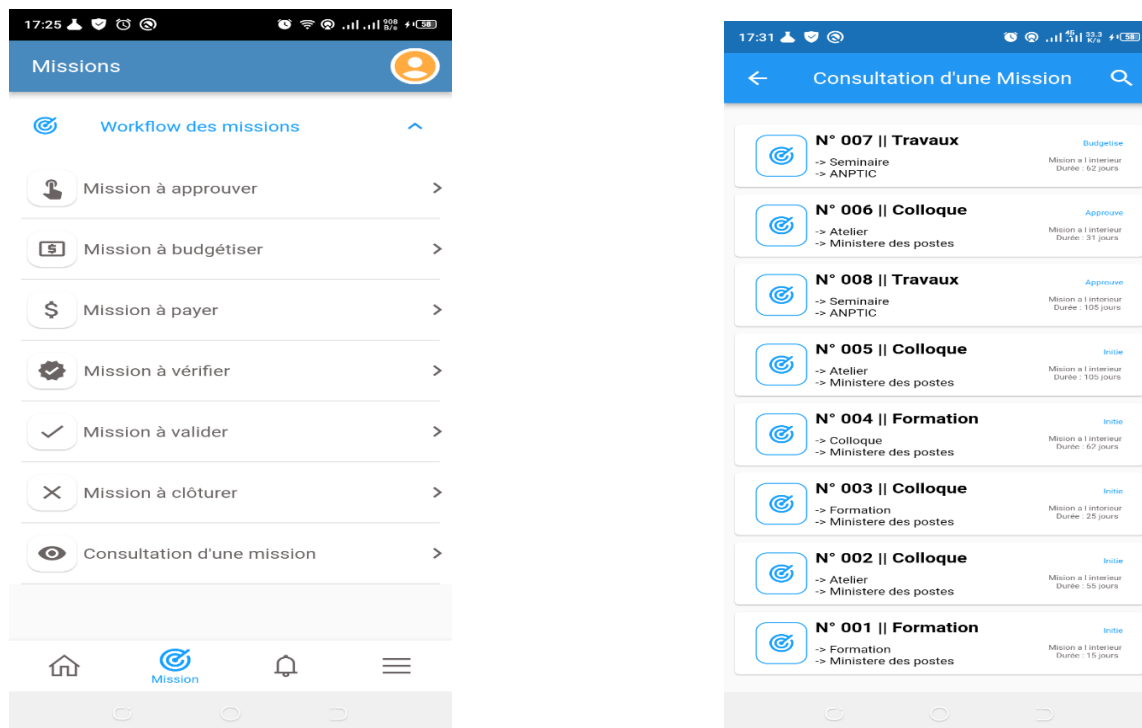


Figure 36 : menu de gestion de mission

DÉVELOPPEMENT D'UNE APPLICATION MOBILE DE GESTION DES MISSIONS : CAS DU CIM/ANPTIC

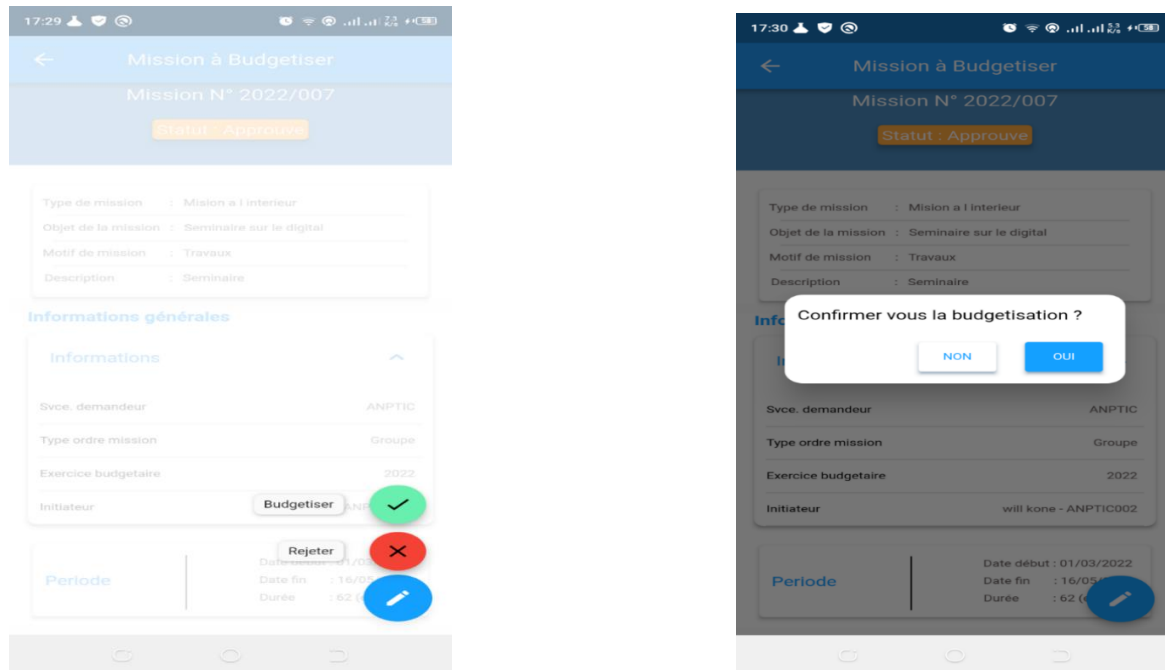


Figure 37 : option de Budgetisation

4. Détails d'une mission

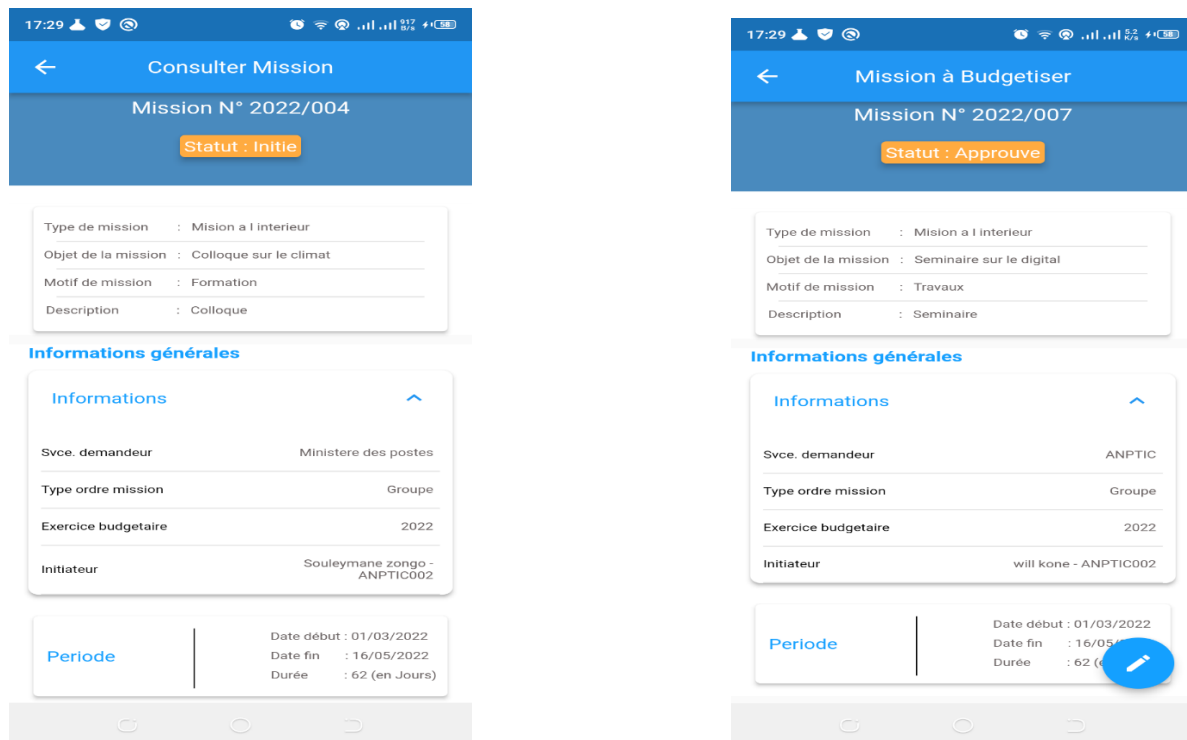


Figure 38 : détails d'une mission

5. Recherche d'une mission

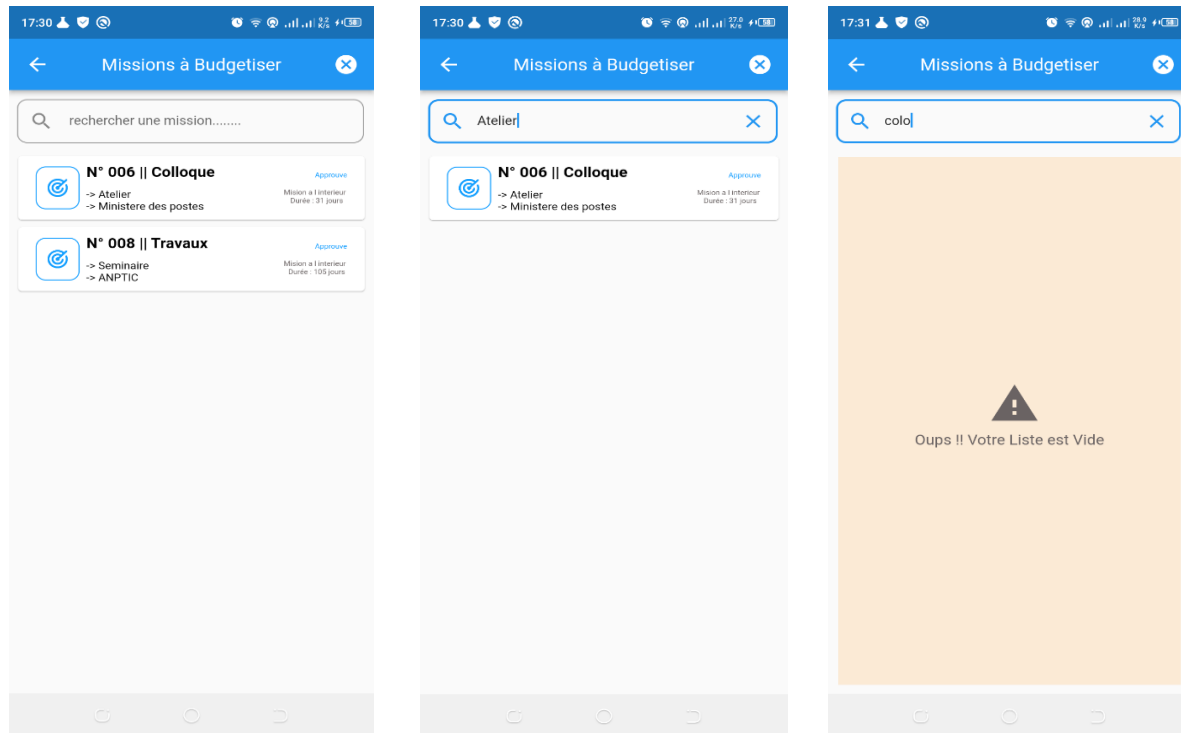


Figure 39 : page de recherche d'une Facture

CONCLUSION

L'étude du thème : « **Développement d'une application mobile de gestion des missions : cas du CIM/ANPTIC.** » qui nous a été soumis dans le cadre de notre stage de fin de cycle nous a permis de mettre en pratique toutes les connaissances acquises durant ces trois (03) années à l'Unité de Formation et de Recherche en Sciences Exactes et Appliquées (UFR/SEA).

CIM est une application qui ambitionne de devenir la plateforme de référence de gestion de mission au sein de l'administration publique au Burkina Faso. Nous y avons participé en développant sa version mobile à l'aide du Framework Flutter développé par Google.

Au terme de nos trois (03) mois de stage, nous avons pu mettre en place une application qui répond à la plupart des besoins potentiels des utilisateurs à savoir la consultation et la gestion des missions.

Ce stage nous a également permis de nous imprégner des méthodes de travail du monde professionnel. Cette application sera utilisée par les agents de l'administration publique, ce qui leur permettra de mieux accroître leurs productivités et leurs réactivités. Ce projet a nécessité l'étude du Framework Flutter et Dart, de Node.js et de plusieurs outils logiciels. Nous nous sommes également familiarisés avec les bases de données de type SQL notamment PostgreSQL, à suivre le rythme de travail d'une entreprise ; à être dans une équipe de pilotage de projet avec des mentors proches pour nous enrichir de conseils.

Notons aussi que nous avons testé notre application « CIM » sur un appareil Android. Le Test sur iOS n'a pas été possible car nous n'avions pas d'appareil iPhone à notre disposition.

Le système étant assez vaste, les principales perspectives à considérer sont, dans un premier temps l'achèvement des notifications par email ou sms, la réalisation des interfaces de paramétrage.

BIBLIOGRAPHIE ET WEBOGRAPHIE

[B1] “Développement d’une application web d’extraction et de structuration de textes juridiques à partir de fichiers PDF et images” de **OUILY Hamed Joseph**, Rapport de Licence/Département Informatique/UFR-SEA/UJKZ.

[B2] “Intégration d’une application de paiement mobile destinée à une solution de gestion commerciale” de **OUEDRAOGO Abdoulaye**, Rapport de Licence/Département Informatique/UFR-SEA/UJKZ

[B3] “Mise en place d’une plateforme web de veille citoyenne: suivre les plans de travail et du budget annuel des ministères” de **ZOMBRE Payiri Grégoire Wenceslas**, Rapport de Licence/Département Informatique/UFR-SEA/UJKZ

[B4] “Système Intégré de Gestion des Annuaire et Décorations du Burkina” de **SANOU Drissa**, Mémoire de fin de cycle/Département Informatique/ESI/UNB

[B5] “Mise en place d’une solution cloud de gestion de la NIDAP” de **GUIRE Boureima**, Mémoire de fin de cycle/Département Informatique/ESI/UNB



[W1] <https://everlaab.com/methode-agile/> consulté le 17/04/2022

[W2] <https://www.nutcache.com/fr/blog/les-methodes-agiles/> consulté le 19/04/2022

[W3] <https://www.monpetitmobile.com/choisir-mobile/systemes-exploitation-smartphones> consulté le 25/04/2022

[W4] <https://www.tuleap.org/fr/agile/comprendre-methode-agile-scrum-10-minutes/> consulté le 18/04/2022

[W5] <https://blog.trello.com/fr/methode-agile-scrum-gestion-projet> consulté le 10/05/2022

[W6] <https://cynoteck.com/fr/blog-post/hybrid-apps-vs-native-apps-the-checklist> consulté le 18/05/2022

[W7] <https://mobiskill.fr/blog/conseils-emploi-tech/top-des-frameworks-de-developpement-multiplateforme/> consulté le 20/05/2022

- [W8] <https://sparxsystems.com/products/ea/> consulté le 22/05/2022
- [W9] [https://fr.wikipedia.org/wiki/Flutter_\(logiciel\)](https://fr.wikipedia.org/wiki/Flutter_(logiciel)) consulté le 28/05/2022
- [W10] <https://docs.flutter.dev/> consulté le 10/05/2022
- [W11] <https://fr.wikipedia.org/wiki/Framework> consulté le 25/05/2022
- [W12] <https://fr.wikipedia.org/wiki/Postman> consulté le 10/06/2022
- [W13] <https://fr.wikipedia.org/wiki/Git> consulté le 22/06/2022
- [W14] <https://fr.wikipedia.org/wiki/GitHub> consulté le 23/06/2022
- [W15] <https://fr.wikipedia.org/wiki/PostgreSQL> consulté le 05/07/2022