

REPUBLIQUE TOGOLAISE

Travail – Liberté – Patrie

PRESIDENCE DE LA REPUBLIQUE
MINISTERE DE LA PLANIFICATION
DU DEVELOPPEMENT ET DE LA
COOPERATION

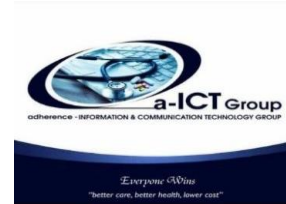


Institut Africain d'Informatique
Représentation du TOGO
(IAI-TOGO)

07 BP 12456 Lomé 07

Tel : (+228) 22 20 47 00

Email : iaitogo@iai-togo.tg



a-ICT Group

Adresse : Portland, OR (USA)

Tel : (+1) 503 840 4334

Email : yvims@aol.com

**PROJET DE FIN DE FORMATION POUR L'OBTENTION DU DIPLOME
D'INGENIEUR DES TRAVAUX INFORMATIQUES**

OPTION : GENIE LOGICIEL ET SYSTEMES D'INFORMATIONS

**PLATEFORME DE COLLECTE, DE GESTION ET DE PRISE DE
RENDEZ-VOUS EN TEMPS REEL POUR L'ACCES AUX SOINS
DE SANTE A LA POLYCLINIQUE MELIA**

Période : Du 22 Mai au 19 Août 2023

Rédigé et soutenu par :

ADZRAKOU Kossiwa Sergine

Etudiante en 3^e année de Génie Logiciel

Année académique : 2022 – 2023

Superviseur :

M. AGBOKA Komlan

Enseignant à l'IAI-TOGO

Maître de stage :

M. EFOUE Jules

Analyste Programmeur à a-ICT Group

DEDICACES

À Dieu, notre rocher et notre refuge,

Nous dédions humblement ce rapport de stage à ta gloire et à ta louange. En tant que source de vie et de sagesse, tu as été notre guide constant tout au long de cette expérience.

À ma source d'inspiration éternelle, à mes parents,

Votre amour inconditionnel et votre soutien infatigable m'ont permis d'atteindre cet accomplissement. Votre foi en moi a été ma force motrice tout au long de ce parcours.

À mes grandes sœurs et grands frères,

Vos conseils perspicaces et vos encouragements constants ont illuminé mon chemin vers le succès. Votre présence a été un rappel constant que je ne suis jamais seul dans cette aventure.

À tous ceux qui croient en moi,

Cette modeste contribution est pour vous. Vos encouragements ont allégé les moments difficiles et ont amplifié les joies. Vous êtes la toile de fond de cette réalisation.

REMERCIEMENTS

Avant tout, nous tenons à exprimer notre profonde gratitude à Dieu, notre Créateur et source de toute grâce et bénédiction. C'est grâce à Sa guidance et à Sa présence constante dans nos vies que nous avons pu accomplir notre parcours de stage avec succès. Nous le remercions pour Sa protection, Sa sagesse et Sa bienveillance infinies. Nos remerciements vont également à :

- M. AGBETI Kodjo, Représentant Résident de l'IAI-TOGO pour son dévouement à la tâche de direction ainsi que ses efforts à l'égard des acteurs de l'institut ;
- M. AMEYIKPO Kossi, Directeur des Affaires Académiques et de la Scolarité de l'IAI-TOGO, pour son sens de la discipline, sa rigueur et ses efforts qu'il déploie ;
- Dr VIMEGNON, Directeur Général de l'a-ICT Group, auprès de qui nous souhaitons exprimer notre sincère gratitude pour son soutien inconditionnel tout au long de notre stage. Sa foi en nos capacités et son engagement envers notre développement professionnel ont été une source d'inspiration et de motivation pour nous ;
- M. EFOUE Jules, notre maître de stage, que nous tenons à remercier chaleureusement pour son encadrement précieux et ses conseils avisés. Sa passion pour son travail et sa volonté de transmettre ses connaissances ont grandement contribué à notre apprentissage et à notre épanouissement professionnel ;
- M. AGBOKA Komlan, notre superviseur, envers qui nous exprimons notre profonde gratitude pour son suivi attentif et son accompagnement tout au long de notre parcours de stage. Ses conseils éclairés et son soutien constant nous ont permis de surmonter les défis et d'atteindre nos objectifs ;
- Dr AMETITIVI Joel, Directeur de la Polyclinique Melia, que nous tenons à remercier pour avoir ouvert les portes de la Polyclinique Melia et pour son

accueil chaleureux. Sa confiance en notre travail et son soutien ont été essentiels pour notre réussite ;

- M. WOGLO Koffi Boris Gilbride, développeur d'applications web backend à SOCIAL INFORMATIQUE pour sa disponibilité, son oreille attentive, nos discussions fructueuses, ses remarques et son soutien durant notre stage;
- Notre famille et nos proches, envers qui nous exprimons notre profonde reconnaissance pour leur amour, leur soutien indéfectible et leurs prières tout au long de notre parcours. Leur encouragement et leur présence ont été une source d'inspiration et de réconfort dans les moments de doute et de difficulté.

Enfin, nous tenons à remercier tous ceux qui ont contribué de près ou de loin à notre réussite. Que ce soit par leurs enseignements, leurs encouragements, leurs conseils ou leur amitié, ils ont tous joué un rôle important dans notre parcours et nous leur sommes infiniment reconnaissantes.

Que Dieu bénisse chacune de ces personnes et qu'il continue de guider nos pas dans nos futurs projets et défis !

SOMMAIRE

DEDICACES.....	i
REMERCIEMENTS	ii
SOMMAIRE.....	iv
RESUME	vi
ABSTRACT.....	vii
GLOSSAIRE.....	viii
LISTE DES FIGURES.....	ix
LISTE DES TABLEAUX.....	xi
LISTE DES PARTICIPANTS AU PROJET	xii
INTRODUCTION.....	1
1 CAHIER DES CHARGES.....	3
1.1 PRESENTATIONS.....	3
1.2 THEME DU STAGE	8
1.3 ETUDE DE L’EXISTANT	11
1.4 CRITIQUE DE L’EXISTANT.....	12
1.5 PROPOSITIONS ET CHOIX DE SOLUTIONS	13
1.6 PLANNING PREVISIONNEL DE REALISATION.....	18
2 ANALYSE ET CONCEPTION.....	21
2.1 CHOIX DE LA METHODE D’ANALYSE ET JUSTIFICATION	21
2.2 CHOIX DE L’OUTIL DE MODELISATION ET JUSTIFICATION.....	25
2.3 ETUDE DETAILLEE DE LA SOLUTION	26
3 REALISATION ET MISE EN ŒUVRE	45
3.1 MATERIELS ET LOGICIELS UTILISES.....	45
3.2 ARCHITECTURES MATERIELLES ET LOGICIELLES DE L’APPLICATION	
50	
3.3 SECURITE DE L’APPLICATION	52
3.4 MISE EN PLACE DE LA BASE DE DONNEES.....	52

4	EXPLOITATION (GUIDE D'EXPLOITATION)	57
4.1	CONFIGURATION LOGICIELLE ET MATERIELLE	57
4.2	DEPLOIEMENT ET SUIVI.....	58
4.3	MAINTENANCE	60
5	GUIDE D'UTILISATION	63
5.1	DESCRIPTION TEXTUELLE DU LOGICIEL.....	63
5.2	PLAN DE NAVIGATION	63
5.3	PRESENTATION DES DIFFERENTES INTERFACES DE L'APPLICATION	
	63	
	CONCLUSION	78
	BIBLIOGRAPHIE INDICATIVE	79
	WEBOGRAPHIE INDICATIVE	80
	TABLE DES MATIERES.....	81

RESUME

Dans le cadre de l'obtention du diplôme d'Ingénieur des Travaux Informatiques, nous avons effectué notre stage de mémoire dans la société a-ICT group afin d'y développer une solution informatique. C'est dans cette optique que nous a été confié le thème : « MASANTE228 : Plateforme de collecte, de gestion et de prise de rendez-vous en temps réel pour l'accès aux soins de santé à la polyclinique Melia ». Ce thème consistera à développer une application mobile pour la meilleure gestion de prise de rendez-vous. Ce projet offre un moyen d'atteindre un plus large public, d'optimiser le processus de prise de rendez-vous. Il permettra également aux patients de faire des demandes de rendez-vous à distance et en toute sécurité. La première étape dans la réalisation de ce projet sera la conception et modélisation de la plateforme à mettre en place avec le langage UML couplé au processus 2TUP. La seconde étape a consisté au développement de la plateforme ; suivie de la rédaction des guides d'exploitation et d'utilisation. Le Système de Gestion de Base de Données utilisé est MySQL et les technologies utilisées sont le Framework flutter pour le frontend, Python, Dart pour le backend. Ainsi, nous avons pu mettre en place la solution grâce à l'application de nos connaissances.

ABSTRACT

As part of our studies to obtain the degree of Computer Works Engineer, we completed our internship and thesis at the company a-ICT group to develop an IT solution. In this context, we were entrusted with the theme: “MASANTE228: Real-time Platform for Collection, Management, and Appointment Scheduling for Access to Healthcare at Polyclinique Melia.” This theme involves developing a mobile application for efficient appointment scheduling. The project aims to reach a broader audience and optimize the appointment scheduling process. It will also enable patients to request appointments remotely and securely.

The first step in realizing this project was the design and modeling of the platform, using the UML language coupled with the 2TUP process. The second step involved the development of the platform, followed by the drafting of operation and user guides. The Database Management System used was MySQL, and the technologies employed included the Flutter framework for the frontend and Python and Dart for the backend. Through the application of our knowledge, we successfully implemented the solution.

GLOSSAIRE

API (Interface de Programmation d'Application) : Un ensemble de règles et de protocoles permettant aux différentes parties d'une application de communiquer entre elles.

Authentification : Le processus de vérification de l'identité des utilisateurs pour accéder aux fonctionnalités sécurisées de l'application.

Base de données : Un système de stockage persistant utilisé pour stocker et gérer les données de l'application.

Django : Un framework de développement web en Python, offrant des fonctionnalités avancées pour la création d'applications web robustes et évolutives.

Framework frontend : Un ensemble d'outils et de bibliothèques préconfigurés pour le développement de l'interface utilisateur d'une application web.

Migration : Un processus automatisé dans Django qui permet de mettre à jour la structure de la base de données en fonction des modifications apportées aux modèles.

Modèle : Dans le contexte de Django, un modèle représente une structure de données qui définit les entités et les relations d'une application.

Python : Un langage de programmation interprété, largement utilisé dans le développement web, connu pour sa simplicité et sa lisibilité.

Template : Un fichier qui contient le code HTML avec des éléments dynamiques pour afficher les données récupérées à partir des vues.

Vue : Dans Django, une vue est une fonction ou une classe qui gère la logique métier d'une page web spécifique, récupère les données nécessaires auprès du modèle et les rend à un template.

LISTE DES FIGURES

Figure 1 : Situation géographique de l'IAI-TOGO	4
Figure 2 : Organigramme de a-ICT Group	6
Figure 3 : Organigramme de la Polyclinique Melia	7
Figure 4 : Situation géographique de la Polyclinique MELIA	7
Figure 5 : Logo UML	22
Figure 6 : Illustration du 2TUP	23
Figure 7 : Logo PowerAMC	25
Figure 8 : Diagramme de cas d'utilisation de « Secrétaire »	28
Figure 9 : Diagramme de cas d'utilisation de « Médecin »	28
Figure 10 : Diagramme de cas d'utilisation de « Patient »	29
Figure 11 : Diagramme de séquences du cas « s'authentifier »	35
Figure 12 : Diagramme de séquences du cas « créer un compte »	36
Figure 13 : Diagramme de séquences du cas « prendre un rendez-vous »	37
Figure 14 : Diagramme d'activités du cas « s'authentifier »	38
Figure 15 : Diagramme d'activités du cas « créer un compte »	39
Figure 16 : Diagramme de déploiement	41
Figure 17 : Diagramme de classes du système	42
Figure 18 : Diagramme de composants	43
Figure 19 : Logo Django Rest Framework	46
Figure 20 : Logo Visual Studio Code	48
Figure 21 : Logo XAMPP	48
Figure 22 : Logo git	49
Figure 23 : Logo Postman	49
Figure 24 : Logo Flutter	50
Figure 25 : Architecture logicielle de l'application	51
Figure 26 : Plan de navigation de l'application	63
Figure 27 : Page d'accueil de l'application mobile	64

Figure 28 : Page de connexion de l'application mobile.....	65
Figure 29 : Page d'inscription à travers l'application mobile	66

LISTE DES TABLEAUX

Tableau 1 : Liste des participants au projet	xii
Tableau 2 : Ressources matérielles de la première solution.....	14
Tableau 3 : Ressources logicielles de la première solution.....	15
Tableau 4 : Ressources matérielles de la deuxième solution.....	16
Tableau 5 : Ressources logicielles de la deuxième solution	17
Tableau 6 : Evaluation financière de la première solution	17
Tableau 7 : Evaluation financière de la deuxième solution.....	18
Tableau 8 : Récapitulatif du planning prévisionnel	18
Tableau 9 : Tableau des matériels utilisés.....	45
Tableau 10 : Cahier des évènements.....	60
Tableau 11 : Codes d'erreur, description et actions correctives.....	61

LISTE DES PARTICIPANTS AU PROJET

Tableau 1 : Liste des participants au projet

NOM & PRENOMS	PROFESSION	ROLE
AGBOKA Komlan	Enseignant à l'IAI-TOGO	Superviseur
EFOUE Jules	Analyste Programmeur à a- ICT Group	Maître de stage
ADZRAKOU Sergine Kossiwa	Etudiante en 3 ^e année de Génie Logiciel et Systèmes d'Informations à l'IAI-TOGO	Chargée de l'étude et de la réalisation

INTRODUCTION

L'informatique permet de simplifier l'exécution des tâches quotidiennes dans tous les domaines par sa rapidité dans le traitement des données, sa fiabilité et la possibilité qu'elle donne d'automatiser certaines tâches.

Cette discipline est inscrite au programme de l'Institut Africain d'Informatique, Représentation du Togo (IAI-Togo), école supérieure qui forme des informaticiens afin de doter notre pays le Togo de cadres informaticiens qualifiés. Cette formation se déroule sur une durée de trois (3) années après lesquelles, lesdits étudiants obtiennent un diplôme d'Ingénieurs des Travaux Informatiques ou de Licence professionnelle en fonction du parcours choisi.

Afin de valider cette formation, chaque étudiant, à la fin de sa troisième année, est soumis à un stage pratique d'une durée de douze (12) semaines dans une entreprise, où il devra prouver toutes les compétences acquises par la conception et le développement d'une solution informatique.

C'est ce qui nous a amené dans les locaux de l'entreprise a-ICT Group où nous avons eu à développer et étudier le thème qui suit : « MASANTE228 : Plateforme de collecte, de gestion et de prise de rendez-vous en temps réel pour l'accès aux soins de santé à la polyclinique Melia ».

Notre document sera réparti en cinq (05) grandes parties qui sont : le cahier des charges, l'analyse et conception, la réalisation et mise en œuvre, le guide d'exploitation et le guide d'utilisation. L'obtention de deux plateformes fonctionnelles, une en web et l'autre en mobile, mettra fin à notre étude.

PARTIE 1 :
CAHIER DES CHARGES

1 CAHIER DES CHARGES

1.1 PRESENTATIONS

1.1.1 Brève présentation de l'IAI-TOGO

L'Institut Africain d'Informatique, Représentation du Togo (IAI-TOGO) est une école inter-Etats d'enseignement supérieur en informatique. Il est membre du réseau créé le 29 janvier 1971 à Fort Lamy (actuel Ndjamena) en République du TCHAD. En application de la décision du Conseil d'Administration de délocaliser l'IAI, la Représentation du Togo (IAI-TOGO) a ouvert ses portes le 24 Octobre 2002. L'accord d'établissement entre la République Togolaise et l'Institut Africain d'Informatique a été signé le 12 mai 2006.

1.1.1.1 Formation à l'IAI-TOGO

L'IAI-TOGO propose trois (03) filières à savoir : « Génie Logiciel et Systèmes d'Informations (GLSI) », « Administration des Systèmes et Réseaux (ASR) » puis « Multimédia, Technologie Web et Infographie (M-TWI) ». L'IAI-TOGO propose actuellement le cycle de formation des ingénieurs de travaux informatiques (Licence professionnelle en informatique). Toujours dans le souci de former des cadres de qualité et très compétitifs sur le marché, l'IAI-TOGO a ouvert le lundi 14 Mai 2012 une formation dotée de certification : les formations CISCO. L'IAI-TOGO se distingue parmi les grandes écoles locales par son approche d'adéquation emploi – formation qui amène les étudiants de la troisième année à effectuer un stage d'une durée de trois (3) mois en entreprise dans le cadre d'un projet de fin de formation pour l'obtention du diplôme.

1.1.1.2 Plan de localisation

L'IAI-TOGO est situé à Kojoviakopé, derrière le bâtiment de l'UTB Circulaire et dans les locaux du Centre National des Etudes et Traitements Informatiques (CENETI).



Figure 1 : Situation géographique de l'IAI-TOGO

1.1.2 Présentation du cadre de stage

Notre stage clôturant notre cycle d'Ingénieur de Travaux Informatique a été effectué à a-ICT Group, qui est une entreprise américaine située à Portland et œuvrant dans le domaine de la santé publique.

1.1.2.1 Statut

Elle a deux pôles d'activité régionale, en Asie et en Afrique. Le pôle asiatique est basé en Malaisie en Kuala Lumpur tandis que le pôle d'Afrique est basé à Abidjan en République de Côte d'Ivoire. Chaque pôle a son Conseil d'Administration sous la présidence du co-fondateur et Chairman, Dr Yves VIMEGNON.

Adresse : Portland, OR (USA),

Téléphone : + 1 503 840 4334

Email : yvims@aol.com

1.1.2.2 Mission

a-ICT Group existe pour aider à améliorer les résultats des soins de santé pour tous grâce à la puissance humaine et informatique combinée. Elle a pour vision d'aider les systèmes de santé du monde entier à disposer des programmes d'intervention les plus rentables pour des résultats de santé optimaux.

Sa mission consiste à soutenir BETTER CARE par les professionnels de la santé et à améliorer la santé des patients à un coût relativement inférieur pour tous les payeurs. Tout en prônant les valeurs fondamentales qui sont : excellence, dévouement, honnêteté, confiance, collaboration gagnant-gagnant.

1.1.2.3 Activités

La prestation de soins de santé avec des capacités numériques de pointe, renforcer le capital humain dans la gestion de la santé de la population grâce à des possibilités de création d'emplois, soutenir la lutte contre les maladies non transmissibles tout en améliorant la couverture universelle des soins de santé sont ses principales activités.

1.1.2.4 Quelques réalisations

L'EHP (Electronic Health Platform), développée par a-ICT Group, est une plateforme novatrice de Gestion de Santé de Population conçue pour révolutionner la manière dont la santé est surveillée, gérée et améliorée à l'échelle d'une population. Cette plateforme technologique avant-gardiste vise à offrir une solution complète pour la gestion de la santé et du bien-être, en permettant une approche proactive et personnalisée de la prise en charge de la santé.

Avec 1854 utilisateurs finaux et 97377 assurés couverts utilisé par 11 pays qui sont :

- Mali
- Burkina Faso
- Gabon
- Cameroun
- Centrafrique
- Guinée
- RDC
- Mali
- Mauritanie
- Niger
- Benin

- **Organigramme**

L'organigramme présentant la structure hiérarchique et les rôles clés au sein du cadre de stage est le suivant :

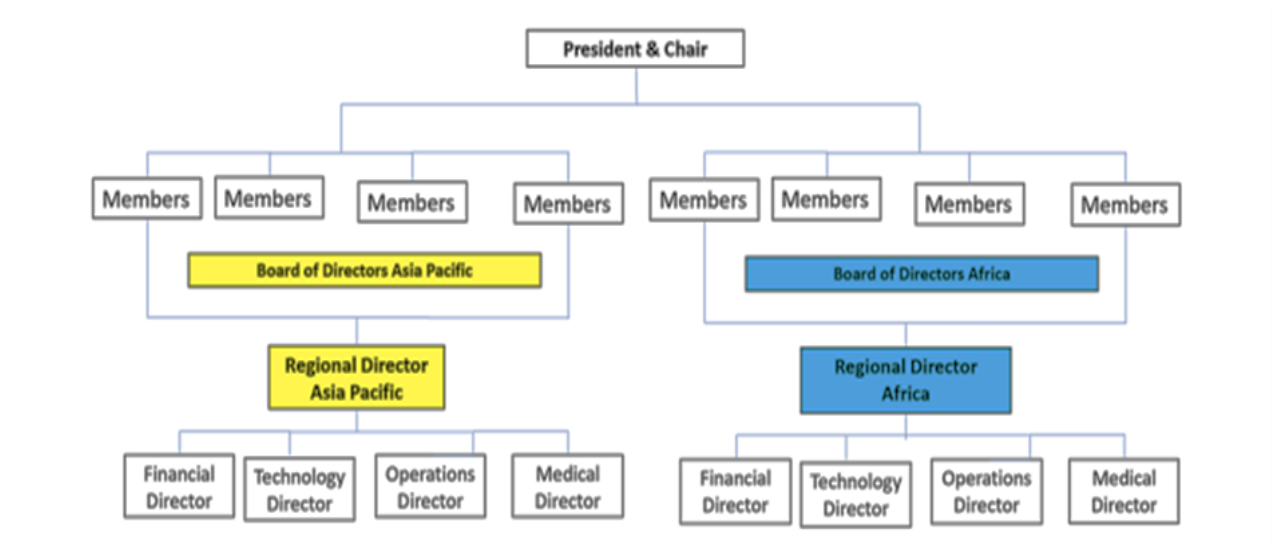


Figure 2 : Organigramme de a-ICT Group

1.1.2.5 Service d'accueil

Nous avons effectué notre stage au sein de la Polyclinique MELIA. Située à Adidogomé, elle est agréée par le ministère de la Santé sous le numéro 70/2007/MS/CAB/DGS. Afin de répondre aux besoins de la population et à la feuille de route de a-ICT Group, la Polyclinique MELIA s'est engagée à offrir des soins d'excellence grâce au recrutement de praticiens reconnus et à l'évolution permanente de son plateau technique, dans le but de favoriser une prise en charge pluridisciplinaire des pathologies.

La Polyclinique MELIA est dirigée par une directrice qui supervise les principaux services, à savoir :

- le secrétariat – caisse ;
- la comptabilité ;
- la médecine générale ;
- la médecine spécialisée ;
- la radiologie ;
- le laboratoire ;

➤ la chirurgie.

L'organigramme de la Polyclinique Melia présente la structure organisationnelle et les différents départements qui contribuent à son bon fonctionnement.

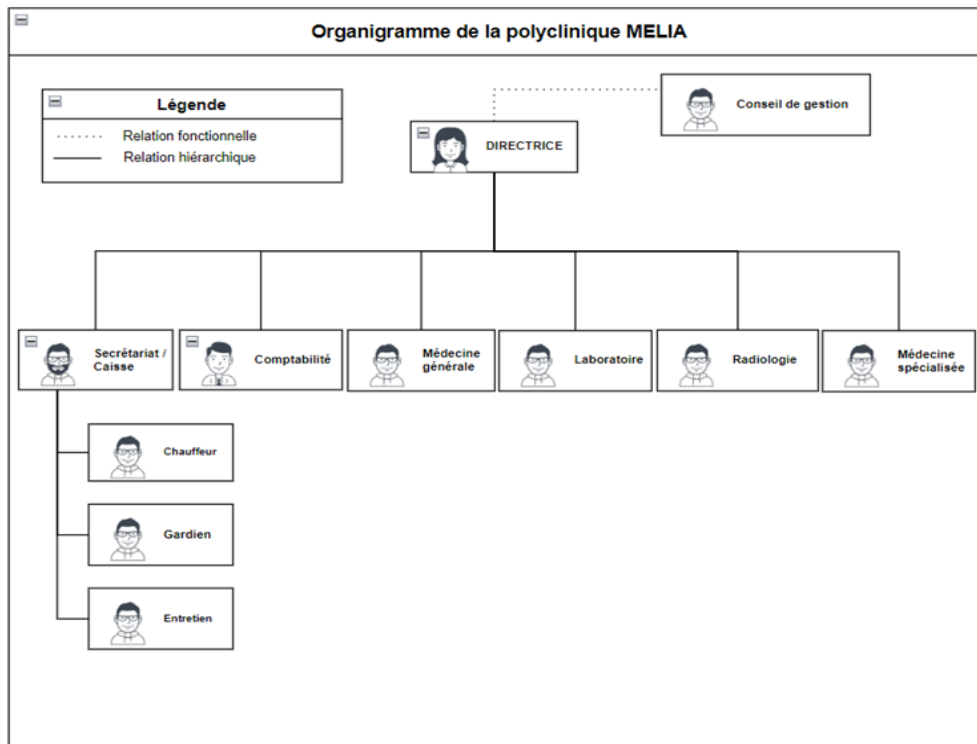


Figure 3 : Organigramme de la Polyclinique Melia

1.1.2.6 Plan de localisation

Le centre d'accueil est situé à Adidogome, franciscains, 3^e rue à gauche, BP: 8925, Lomé, Togo.

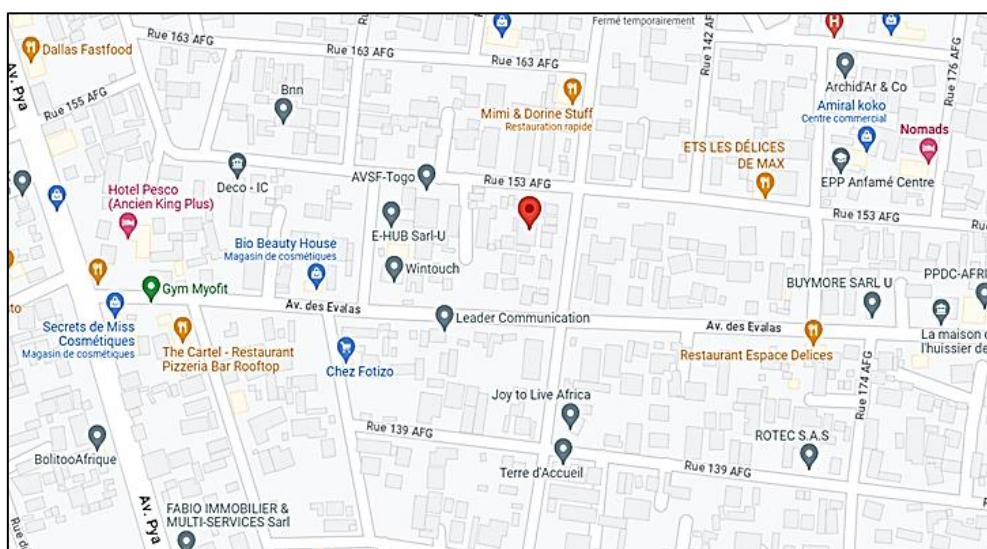


Figure 4 : Situation géographique de la Polyclinique MELIA

1.2 THEME DU STAGE

1.2.1 Présentation du sujet

Le développement des technologies mobiles a engendré une demande croissante d'applications pratiques et conviviales. Notre stage se concentre sur la création d'une application mobile permettant de prendre des rendez-vous en ligne. Avec la progression de la numérisation, les utilisateurs recherchent des solutions plus pratiques et flexibles pour réserver des rendez-vous, notamment dans le domaine médical. L'objectif principal de notre projet consiste à concevoir une application mobile offrant une interface conviviale et intuitive, permettant aux utilisateurs de prendre facilement des rendez-vous depuis leur téléphone portable.

1.2.2 Problématique du sujet

Dans un contexte où les technologies mobiles connaissent un essor sans précédent, la demande croissante d'applications pratiques et conviviales se fait sentir dans de nombreux domaines. Parmi ceux-ci, la gestion des rendez-vous médicaux constitue un enjeu majeur. Face aux problèmes récurrents liés à la prise de rendez-vous, tels que la disponibilité limitée des médecins, il devient impératif de garantir une gestion plus efficace et optimisée de ces rendez-vous cruciaux. C'est dans cette perspective que notre projet se concentre sur le développement d'une application de prise de rendez-vous en ligne, visant à résoudre ces problématiques. Les questions qui se posent alors sont multiples :

- Comment concevoir et mettre en œuvre une application de prise de rendez-vous en ligne efficace afin d'optimiser la gestion des rendez-vous ?
- Comment mettre en place une solution efficace et conviviale permettant aux patients de prendre des rendez-vous en ligne avec les médecins de l'hôpital, tout en optimisant la gestion des rendez-vous, en assurant la disponibilité des médecins, et en réduisant les absences et les annulations de dernière minute ?
- Comment réduire les temps d'attente pour les rendez-vous médicaux ?

- Comment améliorer l'expérience des patients et maximiser l'utilisation des ressources médicales au sein de A-ICT GROUP, tout en garantissant la sécurité des données et en répondant aux besoins spécifiques des patients et du personnel médical ?

1.2.3 Intérêt du sujet

1.2.3.1 Objectifs

1.2.3.1.1 Objectif général

L'objectif principal de notre projet consiste à mettre en place une application de prise de rendez-vous en ligne pour A-ICT GROUP afin de faciliter le processus de prise de rendez-vous pour les patients.

1.2.3.1.2 Objectifs spécifiques

Nous aspirons à améliorer l'expérience patient en offrant une solution pratique, accessible et efficace pour la planification des rendez-vous médicaux. Ainsi, les objectifs spécifiques de cette application sont les suivants :

- **Simplifier le processus de prise de rendez-vous** : Nous visons à permettre aux patients de prendre rendez-vous en ligne de manière simple, rapide et intuitive, éliminant ainsi la nécessité de passer par des appels téléphoniques ou des déplacements physiques ;
- **Réduire les temps d'attente** : Notre objectif est d'optimiser la gestion des rendez-vous afin de diminuer les délais d'attente pour les patients. En offrant la possibilité aux patients de choisir parmi les créneaux horaires disponibles, l'application contribue à une meilleure planification des ressources et à une réduction des temps d'attente ;
- **Améliorer la satisfaction des patients** : Nous souhaitons offrir une expérience utilisateur positive et satisfaisante aux patients en leur offrant la flexibilité de choisir des rendez-vous selon leurs préférences et disponibilités. Cela renforcera la confiance et la fidélité des patients envers A-ICT GROUP.

- **Faciliter la communication et les rappels** : Nous nous efforçons de faciliter la communication entre A-ICT GROUP et les patients en fournissant des rappels automatisés de rendez-vous par le biais de notifications. Cela réduira les oublis de rendez-vous et permettra une meilleure organisation des soins de santé.

En résumé, notre objectif global avec cette application de prise de rendez-vous en ligne est de simplifier et d'optimiser le processus de prise de rendez-vous médicaux, d'améliorer l'expérience patient et d'optimiser l'utilisation des ressources médicales au sein d'A-ICT GROUP. Nous nous engageons également à garantir la sécurité des données et à répondre aux besoins spécifiques des patients et du personnel médical tout au long du développement de cette application

1.2.3.2 Résultats

Les résultats attendus pour ce projet sont notamment :

- **Simplification du processus de prise de rendez-vous** : Notre application permettra aux patients de prendre rendez-vous en ligne de manière simple, rapide et intuitive. Ils pourront choisir des créneaux horaires disponibles et effectuer la réservation sans contrainte de temps ou d'emplacement. Cette simplification contribuera à une meilleure accessibilité aux soins médicaux.
- **Réduction des temps d'attente** : Grâce à une gestion optimisée des rendez-vous, les temps d'attente seront réduits pour les patients. En leur permettant de choisir des créneaux horaires précis, l'application évitera les délais excessifs et optimisera l'utilisation des ressources médicales, assurant ainsi un flux de patients plus fluide et efficace.
- **Amélioration de la satisfaction des patients** : En offrant une expérience utilisateur positive et personnalisée, notre application vise à augmenter la satisfaction des patients. Les fonctionnalités telles que la flexibilité dans le choix des rendez-vous, les rappels automatisés et la communication fluide contribueront à renforcer la confiance et la fidélité des patients envers A-ICT GROUP.

- **Facilitation de la communication et des rappels** : Grâce à des fonctionnalités de communication intégrées, l'application facilitera les échanges entre A-ICT GROUP et les patients. Les rappels automatisés de rendez-vous, les notifications et les informations importantes seront transmis de manière claire, réduisant ainsi les oublis et les erreurs de planification

1.3 ETUDE DE L'EXISTANT

La polyclinique Melia utilise actuellement un système de prise de rendez-vous basé sur des appels téléphoniques et des fiches papier. Les patients sont tenus d'appeler un numéro spécifique pour prendre rendez-vous avec la polyclinique, et les détails des rendez-vous sont enregistrés manuellement lors de ces appels téléphoniques. Une fois planifiés, les rendez-vous sont communiqués aux patients par le personnel médical.

Ce système présente certains avantages, notamment une interaction directe avec un opérateur lors de la prise de rendez-vous, ce qui permet aux patients de poser des questions et d'obtenir des informations en temps réel.

Cependant, des limitations ont été identifiées. Les patients peuvent parfois faire face à des temps d'attente prolongés lorsqu'ils essaient de joindre le service de prise de rendez-vous par téléphone. De plus, la gestion manuelle des fiches papier peut entraîner des retards et des erreurs dans la planification des rendez-vous. Le suivi de l'historique des rendez-vous pour chaque patient peut également être difficile avec le système actuel.

Une enquête a été menée auprès des utilisateurs pour recueillir leurs opinions et suggestions concernant le système de prise de rendez-vous existant. Les résultats de cette enquête ont révélé des préoccupations communes exprimées par les patients et le personnel médical. Ils ont exprimé le besoin d'une meilleure accessibilité, de temps d'attente réduits et d'une gestion plus efficace des rendez-vous.

Sur la base de ces constatations, il devient important de trouver des solutions qui répondent aux attentes des utilisateurs et qui améliorent le processus de prise de rendez-vous. En prenant en compte les avantages du système actuel et les

préoccupations des utilisateurs, une analyse plus approfondie peut-être entreprise pour identifier les améliorations nécessaires et proposer une solution plus efficace et adaptée pour la polyclinique Melia.

1.4 CRITIQUE DE L'EXISTANT

L'une des principales limitations de ce système est les temps d'attente prolongés pour les patients qui tentent de joindre le service de prise de rendez-vous par téléphone. Cela peut entraîner de la frustration et une perte de temps pour les patients, qui doivent attendre de longues périodes avant d'obtenir un rendez-vous.

De plus, le processus de planification des rendez-vous repose sur une gestion manuelle des fiches papier, ce qui peut entraîner des retards et des erreurs. La saisie manuelle des détails des rendez-vous expose le système à des risques d'inexactitudes, compromettant ainsi la fiabilité de la planification.

Une autre lacune importante réside dans la difficulté de suivre l'historique des rendez-vous pour chaque patient. Avec le système actuel, il peut être laborieux et complexe de retrouver les informations et l'historique des rendez-vous antérieurs, ce qui rend la coordination et la continuité des soins plus difficiles. Le suivi est possible car chaque rendez-vous tiens compte d'une date.

Les préoccupations exprimées par les patients et le personnel médical soulignent la nécessité d'une meilleure accessibilité, de temps d'attente réduits et d'une gestion plus efficace des rendez-vous. Ces problèmes doivent être résolus pour améliorer l'expérience des patients et optimiser l'utilisation des ressources médicales.

Il est donc essentiel que la polyclinique Melia explore des solutions alternatives pour surmonter les limitations du système existant et adopter une approche plus efficace et moderne de la gestion des rendez-vous, en se tournant, par exemple, vers une application de prise de rendez-vous en ligne qui répondra aux attentes des patients et du personnel médical.

1.5 PROPOSITIONS ET CHOIX DE SOLUTIONS

1.5.1 Evaluation technique des solutions

1.5.1.1 Première solution

La première solution proposée ici est Masanté@228, une application mobile conviviale qui permet aux patients de prendre des rendez-vous en ligne. L'application devrait offrir une interface intuitive, la possibilité de rechercher des médecins par spécialité, d'afficher les disponibilités en temps réel et de réserver un rendez-vous à un créneau horaire qui convient au patient.

1.5.1.1.1 Avantages

Cette proposition de solution est une application web/mobile pour la prise de rendez-vous en ligne offre des avantages tels que :

- l'accessibilité 24h/24, 7j/7 ;
- la réduction des files d'attente téléphoniques ;
- la visualisation des disponibilités en temps réel ;
- la réduction des erreurs de communication ;
- la simplification du processus de prise de rendez-vous ;
- les rappels automatisés et la centralisation des informations.

1.5.1.1.2 Inconvénients

Les inconvénients associés à la proposition de solution d'une application mobile pour la prise de rendez-vous en ligne :

- barrières technologiques peuvent exclure certains patients qui ne sont pas familiers avec la technologie ou n'ont pas accès régulier à Internet.
- certains patients sans accès à Internet peuvent être exclus de la possibilité de prendre des rendez-vous en ligne.
- l'application peut nécessiter une assistance technique pour son installation, sa configuration ou son utilisation.
- la confidentialité et la sécurité des données des patients doivent être soigneusement gérées pour éviter les violations de données.

- les problèmes de connectivité Internet peuvent empêcher les patients d'accéder à l'application ou de prendre rendez-vous en ligne.
- l'adaptation du personnel administratif et des médecins peut être nécessaire pour utiliser efficacement l'application.

1.5.1.1.3 Ressources matérielles

Afin de mettre en œuvre la solution informatique répondant à notre besoin spécifique, nous avons identifié les ressources matérielles suivantes, chacune jouant un rôle crucial dans le bon fonctionnement du système :

Tableau 2 : Ressources matérielles de la première solution

N°	DESIGNATION	QTE	ROLE
1	Serveur de production	1	Hébergement de l'application et des bases de données, gestion du trafic, réponse aux requêtes des utilisateurs
2	Ordinateurs de développement	1	Permettre aux développeurs de concevoir, programmer, déboguer et tester l'application
3	Smartphones	1	Outil de test de l'application mobile
4	Système de sauvegarde automatisé	1	Effectuer des sauvegardes régulières des données pour éviter les pertes en cas de problème technique ou de défaillance

1.5.1.1.4 Ressources logicielles

Pour assurer une mise en œuvre optimale de la première solution, diverses ressources logicielles sont mobilisées. Le tableau ci-après présente en détails les éléments clés, leurs rôles spécifiques et leur contribution au bon fonctionnement du système.

Tableau 3 : Ressources logicielles de la première solution

N°	NOM	ROLE
1	Microsoft Windows 10 professionnel	Système d'exploitation
2	Android 12	Système d'exploitation pour smartphone

1.5.1.2 Deuxième solution

La seconde proposition consiste à intégrer un chatbot de prise de rendez-vous pour la Polyclinique MELIA. Le chatbot serait accessible sur le site web de la clinique ou via des plateformes de messagerie populaires. Il permettrait aux patients d'interagir, de poser des questions et de prendre des rendez-vous de manière automatisée.

1.5.1.2.1 Avantages

Les avantages de cette solution sont nombreux. Elle offre une accessibilité et une commodité aux patients, qui peuvent interagir à tout moment et de n'importe où. Le chatbot fournit des réponses instantanées et automatisées, réduisant la charge de travail du personnel administratif et simplifiant le processus de prise de rendez-vous. De plus, il peut être personnalisé en fonction des préférences des patients, offrir un support multilingue et envoyer des rappels de rendez-vous.

1.5.1.2.2 Inconvénients

Cependant, il y a aussi des inconvénients à prendre en compte. Les interactions automatisées du chatbot peuvent présenter des limites, nécessitant parfois une intervention humaine supplémentaire. Il peut manquer de personnalisation par rapport à une assistance humaine et avoir des difficultés à comprendre les demandes complexes des patients. De plus, il existe des risques techniques, une résistance au changement de la part des patients habitués aux méthodes traditionnelles, ainsi que des préoccupations en matière de confidentialité et de sécurité des données.

En résumé, l'intégration d'un chatbot de prise de rendez-vous présente des avantages significatifs en termes d'accessibilité, d'efficacité et de personnalisation de l'expérience

des patients. Cependant, il est important de prendre en compte les inconvénients potentiels et de mettre en place des mesures pour atténuer ces risques.

1.5.1.2.3 Ressources matérielles

Pour assurer le bon fonctionnement du chatbot de prise de rendez-vous, un ensemble de ressources matérielles stratégiquement sélectionnées est utilisé. Le tableau ci-dessous présente en détail les désignations, les quantités et les rôles de ces ressources matérielles clés.

Tableau 4 : Ressources matérielles de la deuxième solution

N°	DESIGNATION	QTE	ROLE
1	Serveur de chatbot	1	Hébergement du chatbot et des bases de données, gestion du trafic, réponse aux requêtes des utilisateurs
2	Ordinateurs de développement	1	Permettre aux développeurs de concevoir, programmer, déboguer et tester le chatbot
3	Pare-feu, systèmes de détection d'intrusion, certificats SSL	Selon les normes de sécurité	Protéger les données des utilisateurs et garantir la confidentialité lors des échanges avec le chatbot

1.5.1.2.4 Ressources logicielles

L'intégration réussie du chatbot de prise de rendez-vous repose sur une sélection méticuleuse de ressources logicielles essentielles. Les éléments listés ci-dessous décrivent les noms et les rôles de ces composants logiciels, garantissant ainsi le bon fonctionnement et la performance du chatbot.

Tableau 5 : Ressources logicielles de la deuxième solution

N°	NOM	ROLE
1	Microsoft Windows 10 professionnel	Fournir l'environnement de base pour exécuter et héberger le chatbot, ainsi que d'autres composants logiciels associés.
2	Android 12	Système d'exploitation pour smartphone

1.5.2 Evaluation financière des solutions proposées

Dans cette partie, nous ferons l'étude des coûts financiers des différentes solutions proposées ci-dessus.

1.5.2.1 Première solution

Le tableau ci-dessous présente une analyse détaillée des coûts financiers associés, incluant les désignations, les unités, les quantités, les prix unitaires, et les totaux pour chaque élément envisagé dans la première solution.

Tableau 6 : Evaluation financière de la première solution

DESIGNATION	UNITE	QTE	PRIX UNITAIRE	TOTAL
Laptop Marque Dell				
- Processeur Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz, 1190 MHz,		1	480 000	480 000
- 4 cœur(s), 8 processeur(s) logique(s)				
- (RAM) 12,0 Go				
Android Pixel 3a , 64 Go		1	70 000	70 000
Coût de l'abonnement internet	Mois	12	15 000	180 000
TOTAL				730 000

1.5.2.2 Deuxième solution

De la même manière, nous procédons maintenant à l'évaluation financière de la deuxième solution.

Tableau 7 : Evaluation financière de la deuxième solution

DESIGNATION	UNITE	QTE	PRIX UNITAIRE	TOTAL
Serveur de chatbot		1	800 000	800 000
Ordinateur de développement		1	460 000	460 000
Pare-feu		1	300 000	300 000
TOTAL				1 560 000

Sources :

Prix du Google Pixel 3a : <https://wa.me/message/6XKT5M2N2XSWI1>

1.5.3 Choix de la solution

Au vu des deux solutions et avec l'accord du service Trésorerie, nous optons pour la première solution. Cette solution consiste à développer une application mobile conviviale qui permet aux patients de prendre des rendez-vous en ligne. En plus d'être le plus sûr, cette solution permettra de répondre aux besoins spécifiques fonctionnels.

1.6 PLANNING PREVISIONNEL DE REALISATION

La planification déterminera le bon déroulement et l'enchaînement des activités. Cela requiert une représentation graphique du projet qui permettra d'atteindre les objectifs de délai, de coût et de qualité. Dans le tableau ci-dessous, sont consignées les activités à réaliser avec leurs dates de début et de fin.

Tableau 8 : Récapitulatif du planning prévisionnel

N°	Activité	Date de début	Date de fin	Durée (jours)
1	Prise de contact avec le centre d'accueil ainsi que le thème de stage	22/05/2023	26/05/2023	05
2	Discussion et compréhension du thème de stage	29/05/2023	31/05/2023	03
3	Réalisation et validation du cahier des charges	01/06/2023	03/06/2023	04

N°	Activité	Date de début	Date de fin	Durée (jours)
4	Analyse du thème selon une méthode de conception suivie de la validation	05/06/2023	10/06/2023	05
5	Conception de l'interface utilisateur	13/06/2023	20/06/2023	07
6	Développement de l'application et validation de la réalisation	20/06/2023	20/07/2023	30
7	Rédaction du dossier d'exploitation et guide d'utilisation	22/07/2023	28/07/2023	06
8	Test fonctionnel	01/08/2023	03/08/2023	03
9	Autres corrections et finalisation	03/08/2023	05/08/2023	03
10	Intégration de différentes parties du document2	07/08/2023	10/08/2023	04
11	Dernières formalités avant le dépôt définitif	10/08/2023	16/08/2023	07
12	Dépôt définitif du dossier de mémoire			

PARTIE 2 :
ANALYSE ET CONCEPTION

2 ANALYSE ET CONCEPTION

Les phases d'analyse et de conception sont une étape très importante dans la conception d'un logiciel. Elle constitue la seconde phase de notre projet et est précédée de la phase de rédaction du cahier des charges qui sera notre guide tout au long de la conception de notre système. Cette nouvelle partie du projet nous permettra de décrire explicitement les différents processus à informatiser en utilisant une méthode d'étude et un outil de modélisation appropriés. Notre dossier d'analyse et de conception est divisé en trois grandes parties : le choix de la méthode d'analyse et justification, choix de l'outil de modélisation et justification et l'étude détaillée de la solution.

2.1 CHOIX DE LA METHODE D'ANALYSE ET JUSTIFICATION

2.1.1 Langage UML

Le langage de modélisation unifié, en anglais Unified Modeling Language (UML) est un langage de modélisation graphique à base de pictogrammes (dessins figuratifs stylisés ayant fonction de signe) conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet. UML est le résultat de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG).

UML est un langage formel, défini par un méta modèle. Le méta modèle d'UML décrit de manière très précise tous les éléments de modélisation (les concepts véhiculés et manipulés par le langage) et la sémantique de ces éléments (leur définition et le sens de leur utilisation). C'est en cela qu'on dit qu'UML normalise les concepts objet. Un méta modèle permet de limiter les ambiguïtés et encourage la construction d'outils. Il permet aussi de classer les différents concepts du langage (selon leur niveau d'abstraction ou leur domaine d'application) et expose ainsi clairement sa structure.



Figure 5 : Logo UML

Nous allons ici associer le processus 2TUP au langage UML.

2.1.2 Diagrammes UML

UML 2.3 propose 14 types de diagrammes (contre 9 en UML 1.3). Ces diagrammes sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie. On a :

- Les diagrammes structurels ou statiques (07) qui rassemblent : le diagramme de classes (intervenant dans le système), le diagramme d'objets (instances de classes), le diagramme de composants (du point de vue physique), le diagramme de déploiement, le diagramme des paquetages (ensembles de définitions), le diagramme de structure composite (relations entre les composants) et le diagramme de profils ;
- Les diagrammes comportementaux (03) qui rassemblent : le diagramme des cas d'utilisation (acteurs et interactions avec le système), le diagramme d'états-transitions (description sous forme de machine à états finis du comportement du système et ses composants), le diagramme d'activité (description sous forme d'activités du comportement du système et ses composants) ;
- Les diagrammes d'interaction ou dynamiques (04) qui rassemblent : le diagramme de séquences, le diagramme de communication (échange de messages entre les objets), le diagramme global d'interaction (variante du diagramme d'activité), diagramme de temps (variation d'une donnée au cours du temps).

UML est un formalisme

UML est un langage de conception objet qui permet de modéliser les solutions informatiques par des diagrammes (cités ci-dessus). Il ne propose pas l'ordre qui doit

régir ces diagrammes. C'est en cela que les méthodes viennent appuyer le langage pour rendre totale et parfaite l'approche objet. Les méthodes imposent au langage l'ordre et la démarche. Et comme indiqué précédemment, c'est avec le 2TUP que nous avons décidé de le coupler.

2.1.3 Processus 2TUP

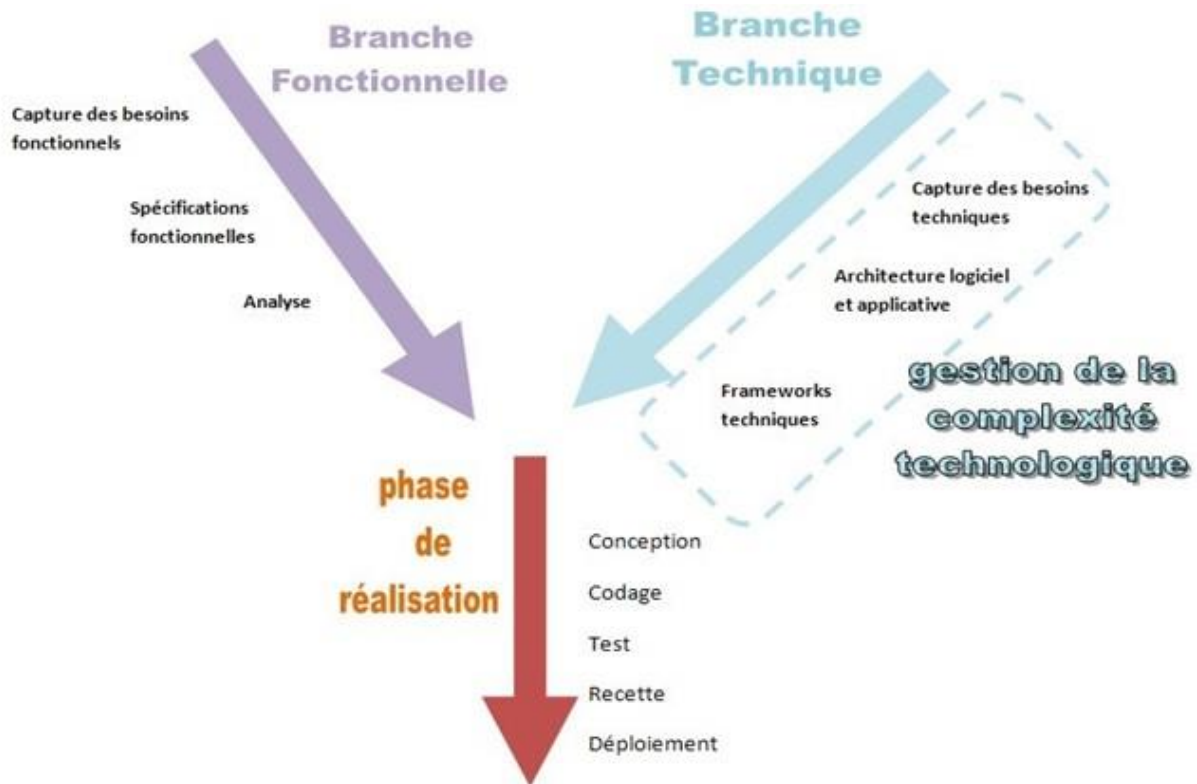


Figure 6 : Illustration du 2TUP

2 Track Unified Process (2TUP), à prononcer "toutiyoupi" est un processus de développement logiciel qui implémente le Processus Unifié. Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système.

Ensuite à produire le cahier des charges et à modéliser le contexte (le système est une boîte) noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens).

Le processus s'articule ensuite autour de 3 phases essentielles :

- Une branche technique ;
- Une branche fonctionnelle ;
- Une phase de réalisation.

2.1.4 Phases du processus 2TUP

- La branche technique (la branche droite) :
 - La capture des besoins techniques qui recense toutes les contraintes sur les choix de dimensionnement et la conception du système, les outils et les matériels sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (prérequis d'architecture technique) ;
 - La conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle a pour objectif d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système, son importance est telle qu'il est conseillé de réaliser un prototype.
- La branche fonctionnelle (la branche gauche) :
 - Elle capture des besoins fonctionnels qui produisent le modèle des besoins focalisés sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs ;
 - L'analyse qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier.
- La phase de réalisation (milieu) :

Elle correspond à :

 - une conception préliminaire qui représente une étape délicate car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer ;

- une conception détaillée qui étudie comment réaliser chaque composant ;
- une étape de codage qui produit ses composants et teste au fur et à mesure les unités de code réalisées ;
- une étape de recette, qui consiste à valider les fonctionnalités du système développé.

2.2 CHOIX DE L'OUTIL DE MODELISATION ET JUSTIFICATION

Dans la modélisation, nous pouvons nous contenter d'utiliser uniquement UML, mais sur quels supports ? De nos jours, nous ne pouvons pas nous contenter des dessins faits sur papier ni au tableau ; c'est là qu'intervient les logiciels de modélisation qui permettent la réutilisation des modèles, leur portabilité (réalisées une fois et utilisables partout), etc.

L'outil retenu pour la modélisation est : Sybase PowerAMC version 15.1.



Figure 7 : Logo PowerAMC

PowerAMC est un environnement graphique de modélisation d'entreprise très simple d'emploi. Il a été créé par la société SDP sous le nom de AMC*Designer, racheté par Power Soft qui lui-même a été racheté par Sybase en 1995. PowerAMC prend en compte plusieurs outils de modélisation tels UML et MERISE.

PowerAMC permet d'effectuer les tâches suivantes :

- Modélisation intégrée via l'utilisation de méthodologies et de notation standards :
 - Données (E/R, Merise) ;
 - Métiers (BPMN, BPEL, ebXML) ;

- Application (UML).
- Génération automatique de code via des templates personnalisables :
 - SQL (avec plus de 50 SGBD) ;
 - Java ;
 - .NET
- Fonctionnalités de reverse engineering pour documenter et mettre à jour des systèmes existants ;
- Une solution de référentiel d'entreprise avec des fonctionnalités de sécurité et de gestion des versions très complètes pour permettre un développement multiutilisateur ;
- Fonctionnalités de génération et de gestion de rapports automatisés et personnalisables ;
- Un environnement extensible, qui vous permet d'ajouter des règles, des commandes, des concepts et des attributs à vos méthodologies de modélisation et de codage.

PowerAMC fournit un jeu unique d'outils de modélisation professionnels qui associent les techniques et notations standard de la modélisation de processus métiers, de la modélisation des données et de la modélisation des diagrammes UML et d'autres fonctionnalités puissantes afin d'aider à analyser, concevoir, construire et maintenir des applications, en utilisant les techniques les plus élaborées d'ingénierie logicielle.

La solution de modélisation PowerAMC permet d'intégrer étroitement la conception et la maintenance des couches de données centrales de l'application et exigences de projet, processus métiers, code orienté objet, vocabulaires XML et informations de réplique de base de données.

2.3 ETUDE DETAILLEE DE LA SOLUTION

Dans cette partie, nous explorerons l'application d'un point de vue interne et externe. Les éléments des spécifications fonctionnelles y seront détaillés afin d'avoir une vision claire et non ambiguë de l'application.

Une façon de mettre en œuvre UML est de considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système.

2.3.1 Etude préliminaire

Nous détaillerons dans cette partie les différents acteurs qui interagiront avec notre système et nous en ferons une représentation à l'aide de diagramme de contexte statique.

2.3.2 Capture des besoins fonctionnels

2.3.2.1 Diagramme des cas d'utilisation

Un diagramme de cas d'utilisation traduit tout ce que l'utilisateur exprime comme action sur le logiciel ou le système à modéliser. C'est une représentation faisant intervenir les acteurs et les cas d'utilisation (ovales). Il traduit les besoins des utilisateurs vis-à-vis du système développé.

2.3.2.1.1 Les acteurs

Un acteur est une personne ou un système qui interagit avec le système en échangeant des informations en entrée comme en sortie. Dans notre cas ici nous déterminons trois acteurs qui sont :

- la secrétaire ;
- le médecin ;
- le patient.

2.3.2.1.2 Les cas d'utilisation

Le cas d'utilisation correspond à un ensemble d'actions ou interactions entre les acteurs et le système en vue d'une finalité. Il représente un des éléments déterminant des fonctionnalités à implémenter dans notre application.

L'étude faite nous a permis de mettre en exergue certains cas d'utilisation que nous avons présentés dans le tableau ci-dessous :

CAS D'UTILISATIONS	ACTEURS
S'authentifier	Patient
S'inscrire	Patient

CAS D'UTILISATIONS	ACTEURS
Gérer les rendez-vous	Tous les utilisateurs
Suivre la consultation	Patient
Consulter les horaires des médecins	Secrétaire
Gérer les patients	Médecin

2.3.2.1.3 Modélisation du diagramme de cas d'utilisation

➤ Secrétaire

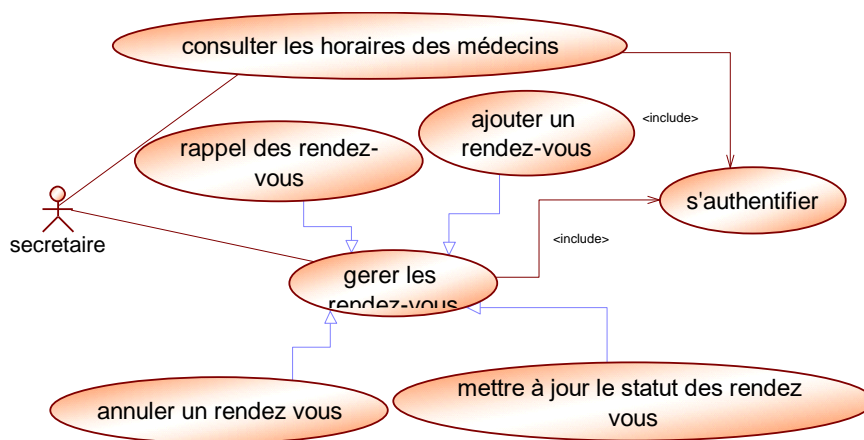


Figure 8 : Diagramme de cas d'utilisation de « Secrétaire »

➤ Médecin

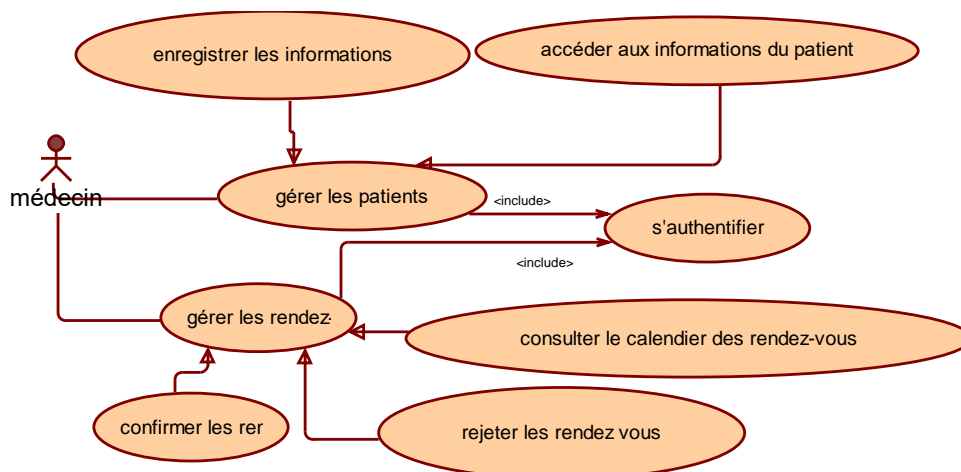


Figure 9 : Diagramme de cas d'utilisation de « Médecin »

➤ Patient

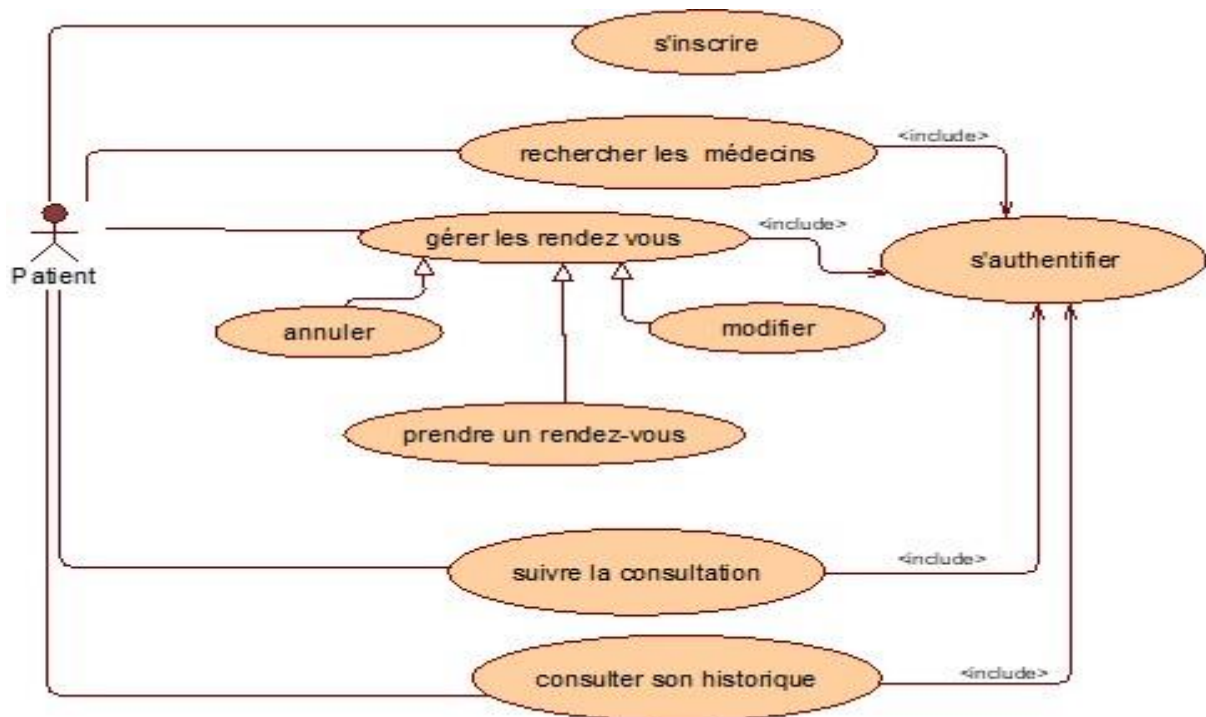


Figure 10 : Diagramme de cas d'utilisation de « Patient »

Le diagramme de cas d'utilisation décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation. Il va falloir décrire de façon littéraire notre diagramme des cas d'utilisation pour une meilleure compréhension. Pour cela, nous allons faire une description textuelle des cas d'utilisation.

2.3.2.1.4 Description textuelle de certains cas d'utilisation

Pour décrire la dynamique du cas d'utilisation, le plus naturel consiste à recenser toutes les interactions de façon textuelle. Cette description permet de clarifier le déroulement de la fonctionnalité et de décrire la chronologie des actions qui devront être réalisées. Nous ferons les descriptions textuelles de : « s'authentifier », « demander rendez-vous ».

➤ Cas d'utilisation « s'authentifier »

SOMMAIRE D'IDENTIFICATION
<p>Titre : S'authentifier</p> <p>Résumé : Ce cas d'utilisation permet à un utilisateur d'être reconnu par le système en saisissant les paramètres de connexion afin d'effectuer des actions sur le système</p> <p>Acteurs principaux : Patient</p> <p>Date de création : 06/06/2023</p> <p>Version : 1.0</p> <p>Responsable : ADZRAKOU Sergine Kossiwa</p>
DESCRIPTION DES SCENARII
Préconditions
<ol style="list-style-type: none"> 1. Disposer d'un smartphone 2. Être connecté à internet 3. Système opérationnel 4. Disposer de l'application sur son smartphone
Scénario nominal
<ol style="list-style-type: none"> 1. L'utilisateur démarre l'application 2. Le formulaire de connexion s'affiche 3. L'utilisateur saisit le mail et le mot de passe (E1) 4. L'utilisateur valide les informations saisies (A1) 5. Le système vérifie le mail et le mot de passe saisis (A2), (E2) 6. Le système informe du succès de connexion et affiche le menu principal
Scénario alternatif
<p>➤ A1 : L'utilisateur ne remplit pas certains champs</p> <ul style="list-style-type: none"> • Ce scénario débute au point 4 du scénario nominal quand l'utilisateur ne renseigne pas les champs obligatoires. • Le système informe l'utilisateur des champs vides <p>Le scénario reprend au point 3 du scénario nominal</p>

- A2 : Les paramètres de connexion sont incorrects moins de 03 fois
 - Ce scénario débute au point 5 du scénario nominal quand les paramètres de connexion saisis ne sont pas justes une (1) fois et (2) fois
 - L'utilisateur est informé que les paramètres saisis ne sont pas corrects
 - Le compteur d'essai est réinitialisé

Scénario d'exception

- E1 : L'utilisateur annule l'opération
 - Ce scénario débute au point 3 du scénario nominal quand l'utilisateur quitte la page
 - Le scénario nominal est interrompu
 - E2 : Les paramètres de connexion sont incorrects pour la 3^e fois
 - Ce scénario débute au point 5 du scénario nominal quand les paramètres de connexions sont incorrects pour la 3^e fois
 - L'utilisateur est informé et le login de l'utilisateur est verrouillé
- Le scénario nominal se termine par un échec

Postconditions

- L'utilisateur se connecte et a accès aux fonctionnalités de l'application.
- Le système est toujours fonctionnel

➤ Cas d'utilisation « s'inscrire »

SOMMAIRE S'INSCRIRE

Titre : S'inscrire

Résumé : Ce cas d'utilisation permet à un utilisateur de s'enregistrer dans le système en saisissant ses informations afin de disposer de paramètres pour la connexion

Acteurs principaux : Patients

Date de création : 06/06/2023

Version : 1.0

Responsable : ADZRAKOU Sergine Kossiwa

DESCRIPTION DES SCENARII
Préconditions
<ol style="list-style-type: none"> 1. Disposer d'un smartphone 2. Être connecté à internet 3. Système opérationnel 4. Disposer de l'application sur son smartphone
Scénario nominal
<ol style="list-style-type: none"> 1. L'utilisateur démarre l'application 2. L'utilisateur choisit de s'inscrire 3. Le formulaire de saisie d'informations s'affiche (E1) 4. L'utilisateur saisit les informations d'inscription 5. L'utilisateur valide les informations saisies 6. Le système vérifie la conformité des informations saisies (A1) (A2) 7. Le système enregistre les informations saisies 8. Le système confirme l'inscription à l'utilisateur
Scénario alternatif
<p>➤ A1 : Les paramètres saisies ne sont pas conformes aux exigences du système</p> <ul style="list-style-type: none"> • Ce scénario débute au point 6 du scénario nominal quand les paramètres saisis au point 4 ne sont pas conformes • L'utilisateur est informé <p>Le scénario reprend au point 4 du scénario nominal.</p>
Scénario d'exception
<p>➤ E1 : L'utilisateur décide de ne plus s'inscrire et annule l'opération</p> <ul style="list-style-type: none"> • Ce scénario débute au point 3 du scénario nominal • Ce scénario arrête le processus d'inscription
Postconditions
<p>➤ L'utilisateur dispose d'un mail et mot de passe lui permettant de se connecter.</p> <p>➤ Le système est toujours fonctionnel</p>

➤ **Cas d'utilisation « prendre un rendez-vous »**

SOMMAIRE PRENDRE UN RENDEZ-VOUS
<p>Titre : Prendre un rendez-vous</p> <p>Résumé : Ce cas d'utilisation permet à un utilisateur authentifié de prendre un rendez-vous avec un professionnel de la santé disponible dans l'application.</p> <p>Acteurs principaux : Patients</p> <p>Date de création : 07/06/2023</p> <p>Version : 1.0</p> <p>Responsable : ADZRAKOU Sergine Kossiwa</p>
DESCRIPTION DES SCENARII
Préconditions
<ol style="list-style-type: none">1. Être authentifié sur l'application en utilisant le cas d'utilisation "S'authentifier".2. Disposer d'une connexion internet active.
Scénario nominal
<ol style="list-style-type: none">1. L'utilisateur accède à l'application et se connecte avec succès en utilisant le cas d'utilisation "S'authentifier".2. L'utilisateur accède à la section "Prendre rendez-vous" dans le menu principal de l'application.3. Le système affiche la liste des professionnels de la santé disponibles pour les rendez-vous, avec leurs spécialités respectives.4. L'utilisateur sélectionne un professionnel de la santé et choisit la date et l'heure souhaitées pour le rendez-vous.5. Le système vérifie la disponibilité du professionnel et de la plage horaire choisie.6. Si la date et l'heure sont disponibles, le système confirme la prise de rendez-vous et envoie une notification de confirmation à l'utilisateur.7. Le rendez-vous est enregistré dans le système et ajouté au calendrier de l'utilisateur.
Scénario alternatif

- A5 : La date ou l'heure choisie n'est pas disponible.
 - Ce scénario débute au point 5 du scénario nominal lorsque la date ou l'heure choisie n'est pas disponible.
 - Le système informe l'utilisateur que la date ou l'heure n'est pas disponible et propose des dates/heures alternatives.
 - L'utilisateur peut choisir une nouvelle date/heure parmi les options proposées.

Postconditions

- L'utilisateur a pris rendez-vous avec succès avec le professionnel de la santé choisi.
- Le rendez-vous est enregistré dans le système et visible dans le calendrier de l'utilisateur.
- Le système est toujours fonctionnel.

2.3.2.2 Diagramme de séquences

Un diagramme de séquence est un type de diagramme d'interaction, car il décrit comment et dans quel ordre plusieurs objets fonctionnent ensemble. Il représente la séquence de messages entre les objets au cours d'une interaction. Il permet aussi de :

- représenter les détails d'un cas d'utilisation UML ;
- voir comment les objets et les composants interagissent entre eux pour effectuer un processus ;
- décliner le déroulement logique d'une procédure, fonction ou opération complexe ;
- schématiser et comprendre le fonctionnement détaillé d'un scénario existant ou à venir

Voici ci-dessous les différents diagrammes de séquence de notre projet :

➤ Diagramme de séquences du cas d'utilisation « s'authentifier »

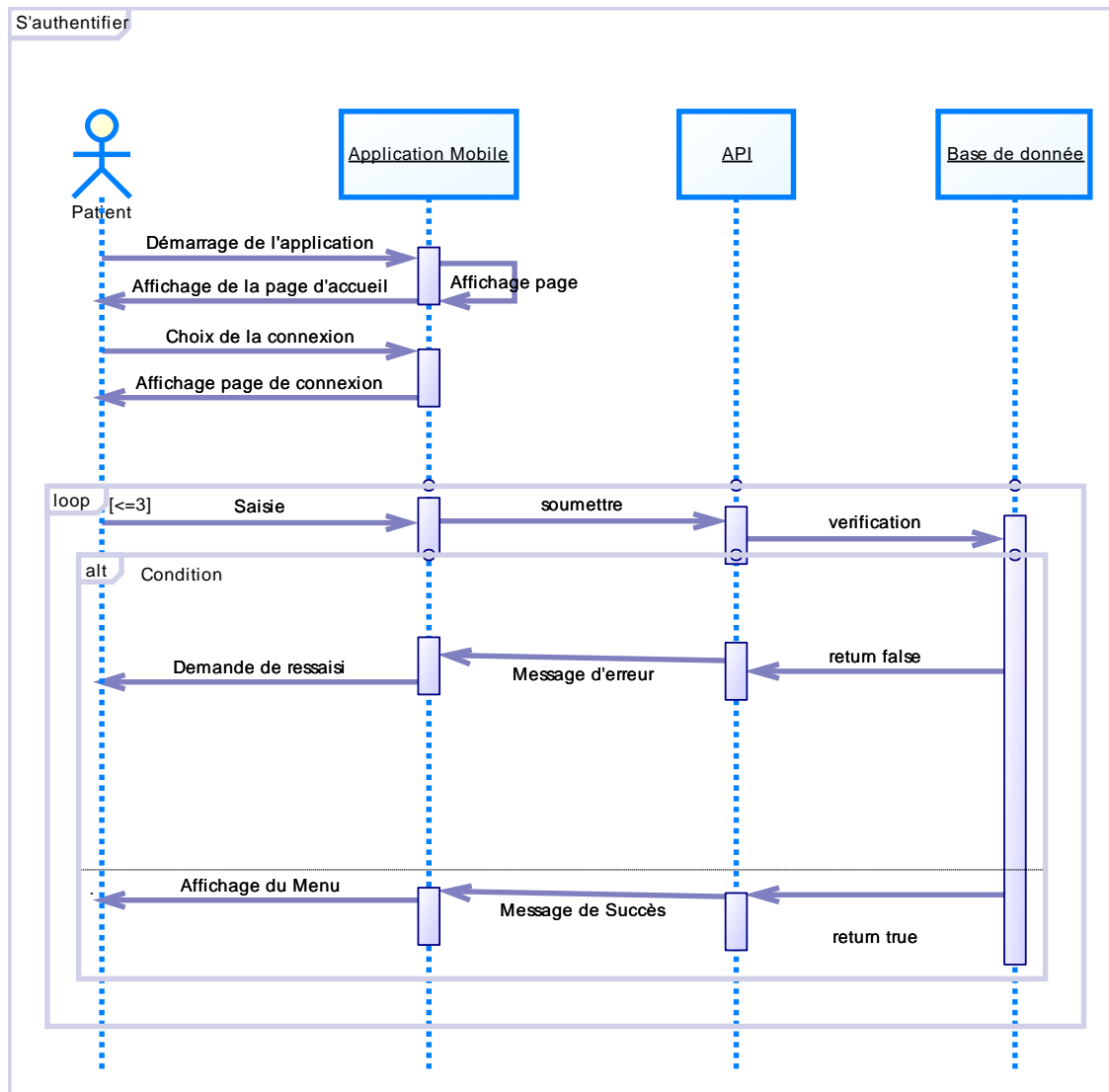


Figure 11 : Diagramme de séquences du cas « s'authentifier »

➤ **Diagramme de séquences du cas d'utilisation « créer un compte »**

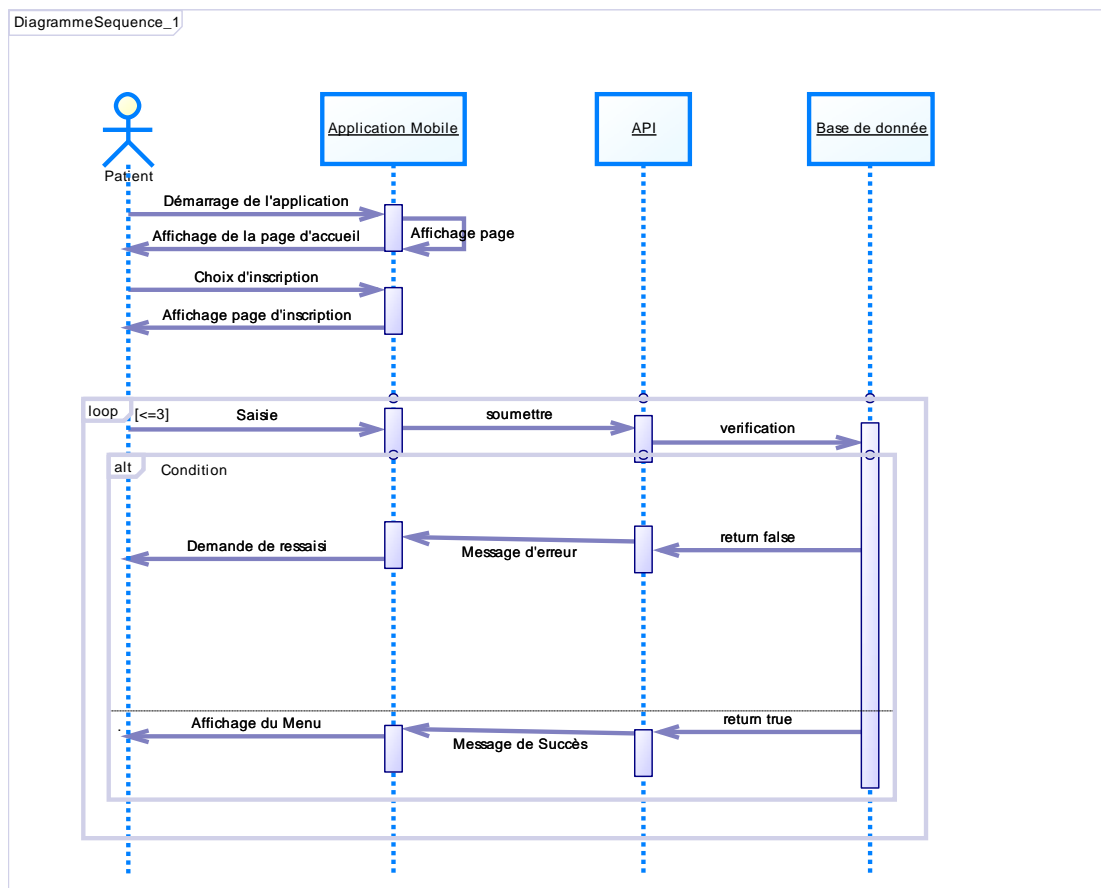


Figure 12 : Diagramme de séquences du cas « créer un compte »

➤ **Diagramme de séquences du cas « prendre un rendez-vous »**

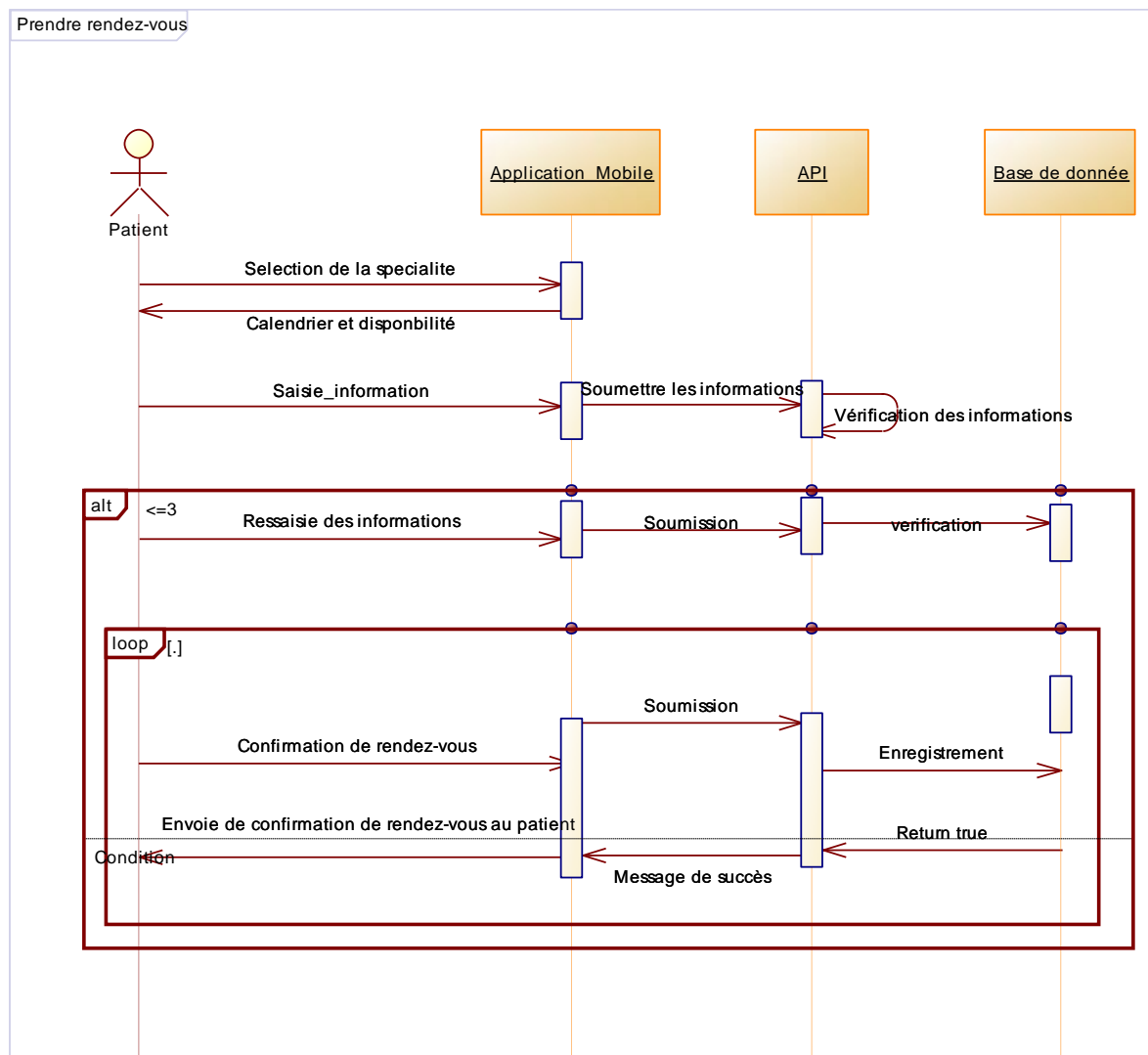


Figure 13 : Diagramme de séquences du cas « prendre un rendez-vous »

2.3.2.3 Diagramme d'activités

Le diagramme d'activités se présente comme un organisme qui décrit les différents enchainements ou étapes du déroulement d'un cas d'utilisation ou groupe de cas d'utilisation. Le diagramme d'activité est une représentation proche de l'organigramme ; la description d'un cas d'utilisation par un diagramme d'activité correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.

Certains des composants les plus courants d'un diagramme d'activités sont :

- **Actions** : étape dans l'activité où les utilisateurs ou le logiciel exécutent une tâche donnée.

- Nœud de décision : embranchement conditionnel dans le flux qui est représenté par un losange. Il comporte une seule entrée et deux sorties ou plus.
- Flux de contrôle : autre nom des connecteurs qui illustrent le flux entre les étapes du diagramme.
- Nœud de départ : symbolise le début de l'activité. Cet élément est représenté par un cercle noir.
- Nœud de fin : représente l'étape finale de l'activité. Il est représenté par un cercle noir avec un contour.

Le diagramme d'activités présente uniquement les actions du système. Voici quelques diagrammes d'activité de notre système :

➤ **Diagramme d'activités du cas d'utilisation « s'authentifier »**

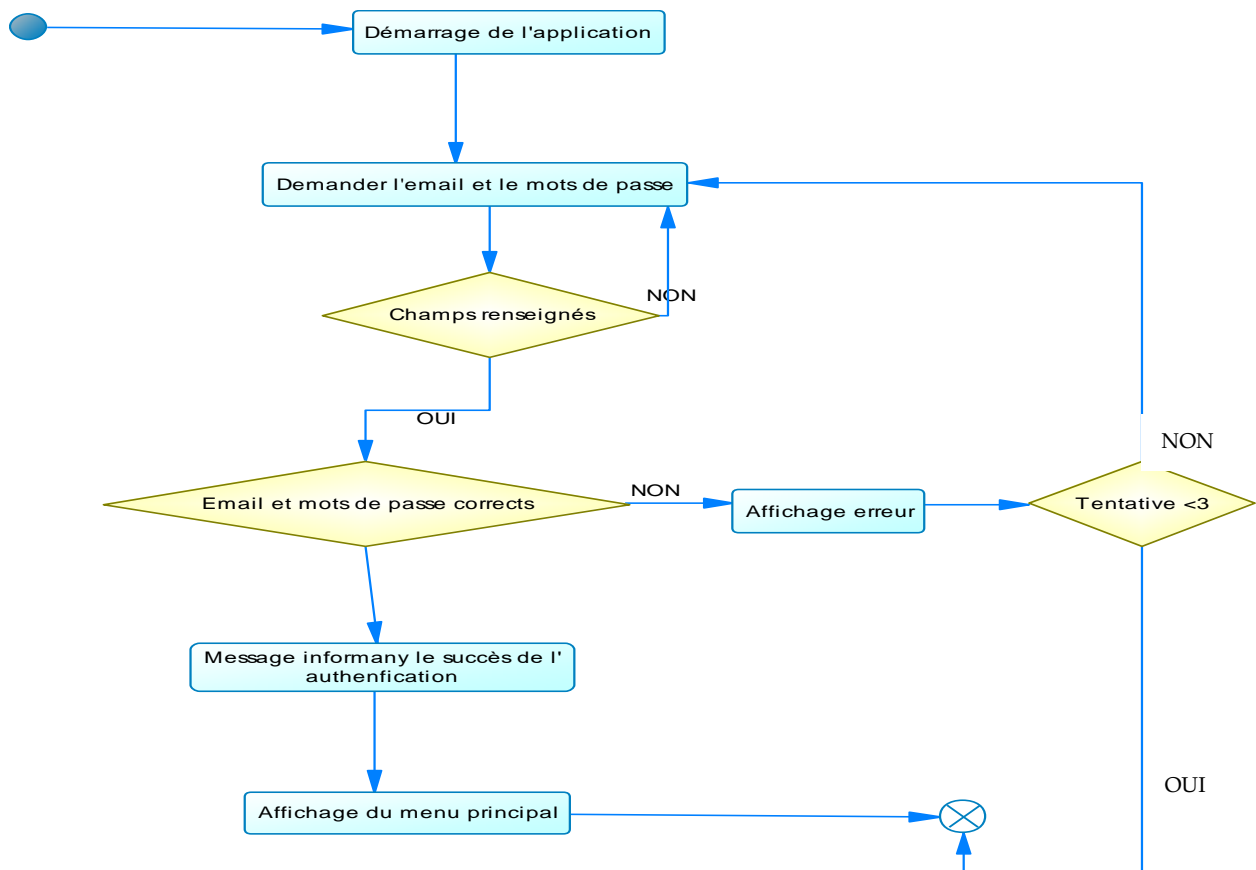


Figure 14 : Diagramme d'activités du cas « s'authentifier »

➤ Diagramme d'activités du cas d'utilisation « créer un compte »

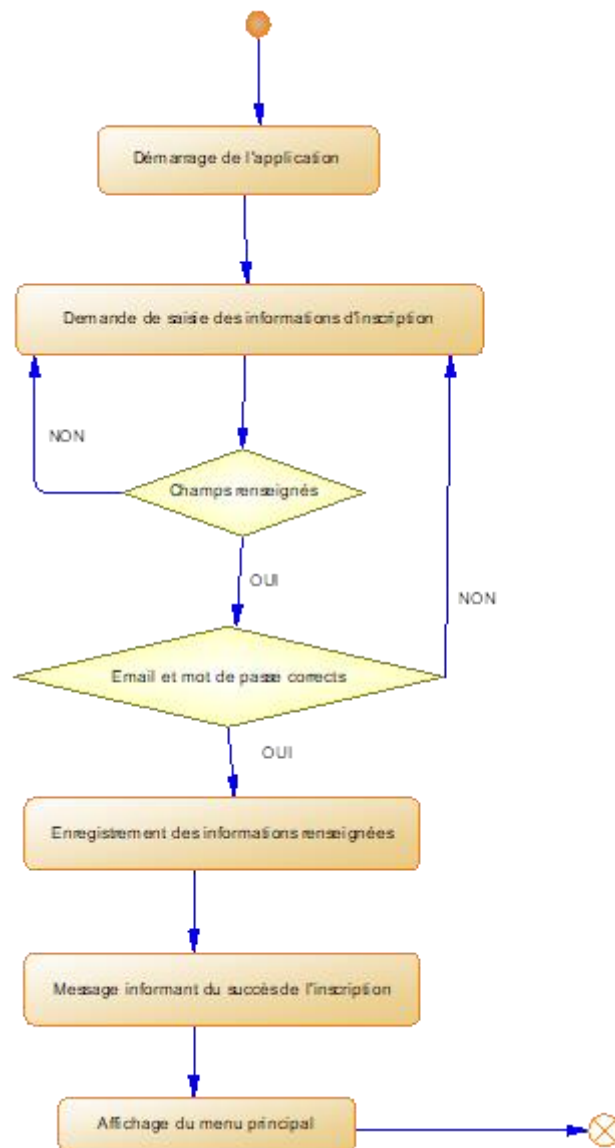


Figure 15 : Diagramme d'activités du cas « créer un compte »

➤ Diagramme d'activités du cas d'utilisation « prendre rendez-vous »

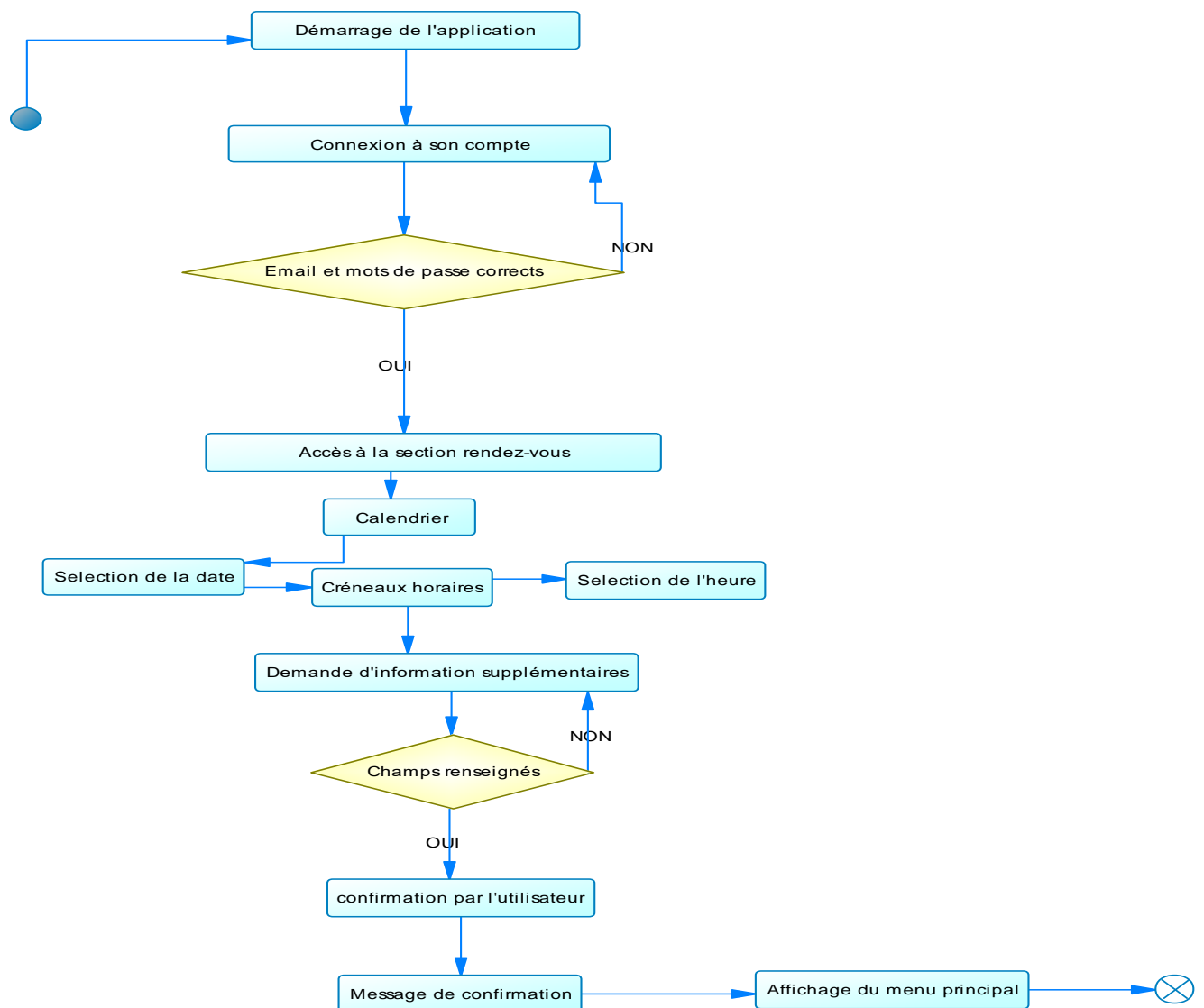


Figure 15 : Diagramme d'activités du cas « prendre un rendez-vous »

2.3.3 Capture des besoins techniques

2.3.4 Conception

2.3.4.1 Diagramme de déploiement

Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. Chaque ressource étant matérialisée par un nœud, le diagramme de déploiement précise comment les composants sont repartis sur les nœuds et quelles sont les connexions entre les composants ou les nœuds.

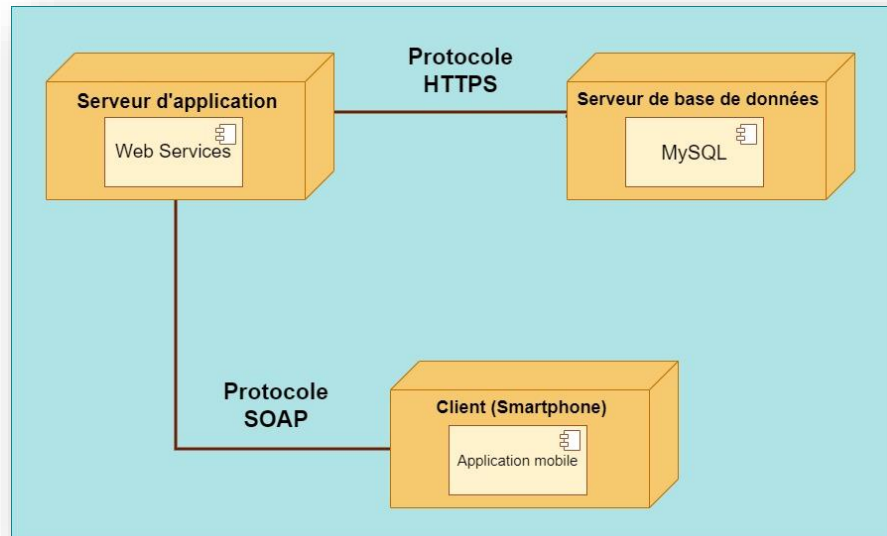


Figure 16 : Diagramme de déploiement

2.3.4.2 Diagramme de classes

Le diagramme de classe exprime la structure statique du système en termes de classes et de relations entre ces classes. L'intérêt du diagramme de classe est de modéliser les entités du système d'informations. Le diagramme de classe permet de représenter l'ensemble des informations finalisées qui sont gérées par le domaine. Ces informations sont structurées, c'est-à-dire qu'elles sont regroupées dans des classes. Le diagramme des classes met en évidence d'éventuelles relations entre ces classes. Le diagramme des classes est le diagramme le plus répandu dans les spécifications d'UML. Le diagramme de classes comporte 6 concepts qui sont : classe, attribut, identification, relation, opération, généralisation / spécialisation.

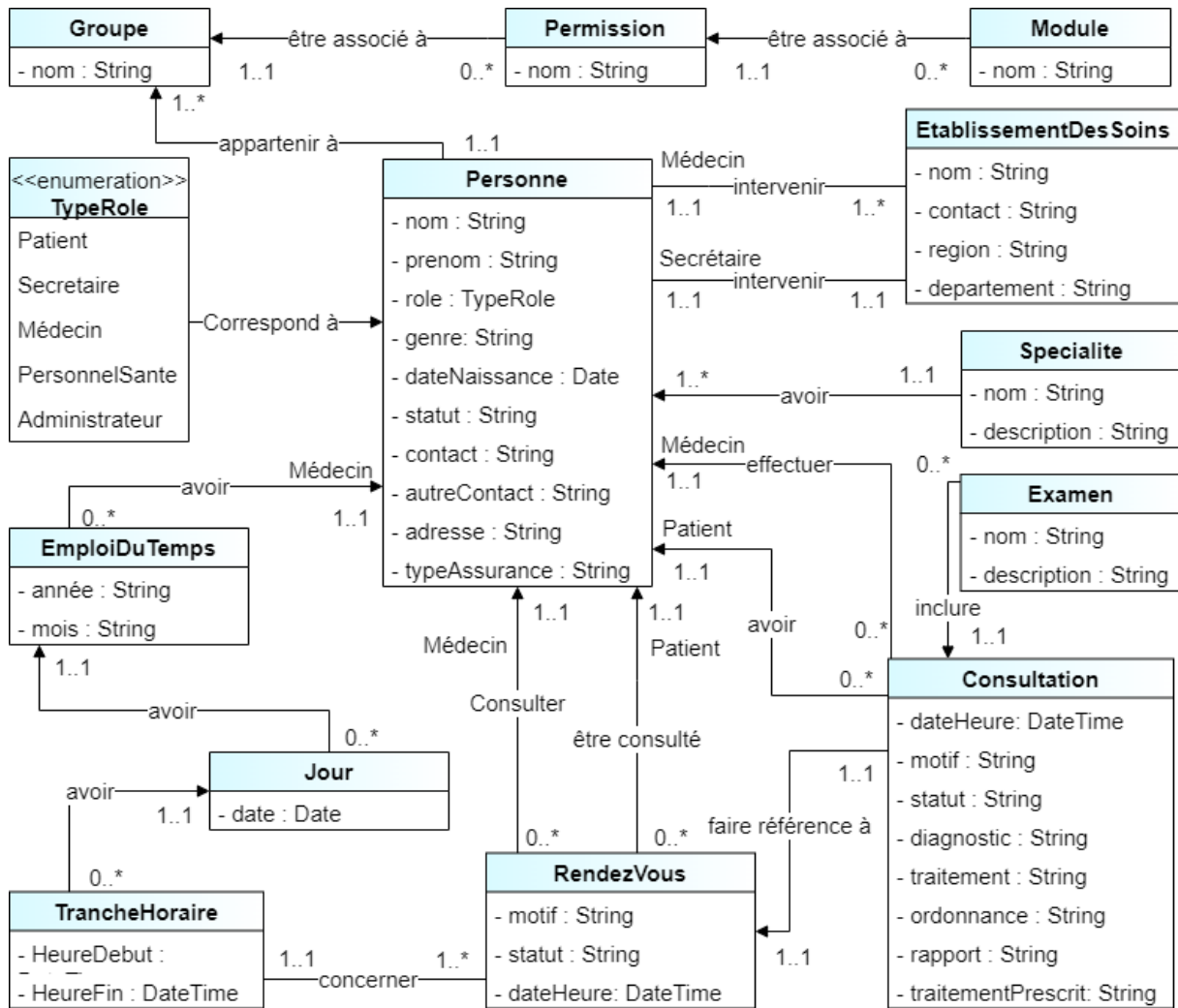


Figure 17 : Diagramme de classes du système

2.3.4.3 Diagramme de composants

Un composant doit fournir un service donné. Les fonctionnalités qu'il encapsule doivent être cohérentes entre elles et génériques étant donné que sa vocation est d'être réutilisable. Un composant est une unité autonome représentée par un classeur structuré, stéréotypé « *component* », comportant une ou plusieurs interfaces requises ou offertes. Son comportement interne, généralement réalisé par un ensemble de classes, est totalement masqué : seules ses interfaces sont visibles.

La relation de dépendance est utilisée dans le diagramme de composants pour indiquer qu'un élément de l'implémentation d'un composant fait appel aux services offerts par les éléments d'implémentation d'un autre composant. Ci-dessous figure notre diagramme de composants.

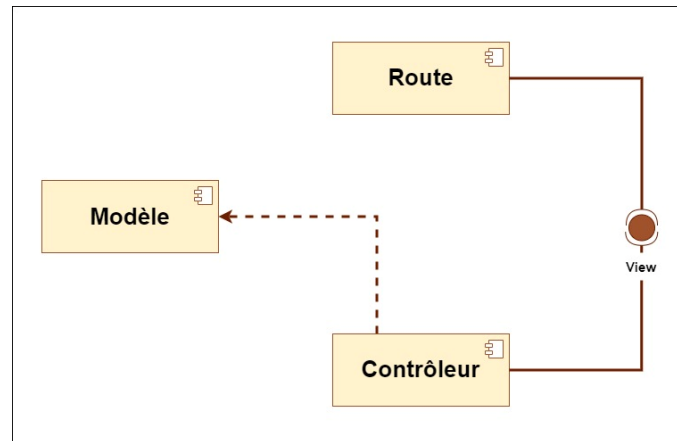


Figure 18 : Diagramme de composants

PARTIE 3 :
REALISATION ET MISE EN ŒUVRE

3 REALISATION ET MISE EN ŒUVRE

Après étude détaillée de notre projet, nous devons implémenter la solution choisie qui consiste à développer une application mobile. Ce document nous permettra de décrire la mise en œuvre de l'application, ensuite de présenter l'application elle-même et enfin d'exposer le planning de réalisation.

3.1 MATERIELS ET LOGICIELS UTILISES

3.1.1 Matériels utilisés

Tableau 9 : Tableau des matériels utilisés

DESIGNATION	DESCRIPTION	SYSTEME D'EXPLOITATION
Ordinateur portable	Marque Dell Processeur Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz, 1190 MHz, 4 cœur(s), 8 processeur(s) logique(s)	Microsoft Windows 10 Famille
Téléphone portable	Pixel 3a (Taille de l'écran : 5.6in - Mémoire interne : 64 Go – Processeur : Snapdragon 670 - Capacité de la batterie - 3000 mAh - Système d'exploitation : Android - Version du système d'exploitation : Android 9.0 - Nombre de cœurs CPU : Octa-Core - Fréquence CPU : 2GHz – GPU : Adreno 615 - Mémoire vive (RAM) : 4 Go)	Android 9.0

3.1.2 Logiciels utilisés

La phase de codage est la phase déterminante dans la réalisation d'un projet informatique et pour implémenter notre solution nous utiliserons divers outils.

Notre projet se subdivise en deux (2) parties à savoir une application mobile et une API (Application Programming Interface) hébergée sur le web. Au vu de la subdivision, nous utilisons différents langages pour développer chacune de ces parties.

➤ API

Nous utiliserons pour notre application Django Rest Framework car ce framework inclut :

- la prise en charge des méthodes http ;
- la gestion des routes d'URL ;
- la sérialisation des données ;
- l'authentification et l'autorisation ;
- la pagination, la validation des données, le filtrage et la recherche, la gestion des erreurs et des exceptions, la documentation automatique et l'extensibilité.



Figure 19 : Logo Django Rest Framework

Pour gérer la partie base de données, nous utiliserons le SGBD (Système de Gestion de Base de Données) MySQL avec le langage de requête SQL (Structured Query Language).

Nous disposons d'une multitude d'autres SGBD tels que SQL Server, PostgreSQL, Oracle, Maria DB, que nous pourrions utiliser mais nous avons choisi MySQL notamment pour :

- la facilité d'installation, d'utilisation et de prise en main ce qui n'est pas le cas des autres comme SQL Server, Oracle ou PostgreSQL qui nécessitent des configurations de base avant toute utilisation ;
- la légèreté, rapidité et fluidité ce que l'on ne note pas chez les autres SGBD ;

- la gratuité : parmi les autres SGBD cités, nous en distinguons des payants et des gratuits comme PostgreSQL mais MySQL possède une configuration plus simple ;

Nous utiliserons pour notre projet la version 2016 de SQL.

- **Outils et logiciels utilisés**

La phase de codage est la phase déterminante dans la réalisation d'un projet informatique et pour implémenter notre solution nous utiliserons divers outils. Parmi ces outils, nous distinguons les **IDE** (Environnement de Développement Intégré), les environnements de serveurs, etc.

- Environnement de développement intégré (EDI) : Visual Studio Code

Visual Studio Code est un IDE (Integrated Development Environment) léger mais puissant qui est disponible pour Windows, MacOS et Linux. Il est un éditeur développé par Microsoft et supportant un très grand nombre de langages. Il supporte l'auto complétion, la coloration syntaxique, le débogage, et les commandes git.

En dépit de toutes les librairies citées plus haut, il dispose énormément d'extensions et fonctionnalités qui aident au développement rapide d'applications dans plusieurs langages de programmation. Il existe une multitude d'autres éditeurs de code tels que Sublime Text, Notepad++, Atom, Brackets, mais notre choix s'est porté sur ce Visual Studio Code car :

- interface utilisateur légère et conviviale : VS Code offre une interface utilisateur minimaliste et intuitive qui met l'accent sur la productivité du développeur.
- prise en charge multiplateforme : Il est disponible pour Windows, macOS et Linux, ce qui permet aux développeurs de travailler sur différentes plateformes.
- extensibilité : VS Code prend en charge une large gamme d'extensions qui permettent d'ajouter des fonctionnalités supplémentaires et d'adapter l'environnement de développement aux besoins spécifiques du développeur.
- intégration avec Git : Il dispose d'une intégration Git native, ce qui facilite le suivi des modifications du code source et la gestion des versions.

- débogage intégré : Il propose des fonctionnalités de débogage intégrées qui permettent aux développeurs de mettre en place des points d'arrêt, d'exécuter le code pas à pas et de trouver et résoudre les erreurs plus facilement.
- prise en charge de nombreux langages de programmation : VS Code prend en charge de nombreux langages de programmation populaires, tels que JavaScript, Python, Java, C++, Go, et bien d'autres encore.
- intégration avec les outils de construction et de test : Il offre une intégration avec des outils couramment utilisés pour la construction et le test des applications, tels que npm, Maven, Gradle, et bien d'autres encore.

Ainsi, pour notre développement de l'application mobile ainsi que de l'API, notre choix s'est porté sur la dernière version de Visual Studio Code.



Figure 20 : Logo Visual Studio Code

Serveur d'application

Comme serveur d'application, nous utiliserons Xampp(version 3.3.0) . nous aurons pu choisir Wamp Server, Easy PHP ou encore Laragon. Pourquoi Xampp ?

XAMPP est un environnement de développement web comprenant Apache (serveur web), MySQL/MariaDB (gestionnaire de bases de données), PHP/Perl (langages de script), et d'autres outils. Il permet de créer un serveur web local pour développer et tester des sites web et des applications. XAMPP offre une compatibilité multiplateforme, facilite la gestion des bases de données via phpMyAdmin, et permet le développement local sans connexion internet. Cependant, il n'est pas sécurisé pour une utilisation en production. Utilisé principalement à des fins de développement et de test.



Figure 21 : Logo XAMPP

➤ L'outil GIT

GIT est un outil de versionnage de code source qui permet une intégration continue. Il suit l'évolution des fichiers sources et garde les anciennes versions de chacun d'eux sans les écraser. Il se distingue par sa rapidité et sa gestion des branches. Notre choix s'est porté sur lui car il est le plus utilisé donc le plus documenté même s'il n'est pas le plus complet comparé à Kiuwan qui lui, possède un examinateur de code. Nous utiliserons la version 2.3.3.0.



Figure 22 : Logo git

➤ L'outil Postman

Postman est un outil populaire utilisé pour tester et développer des API. Il offre une interface conviviale qui permet aux développeurs de créer, envoyer et recevoir des requêtes HTTP, d'inspecter et de manipuler les réponses, ainsi que de collaborer avec d'autres membres de l'équipe. Postman prend en charge diverses fonctionnalités telles que l'envoi de requêtes GET, POST, PUT, DELETE, la gestion des paramètres et des en-têtes, la visualisation et la validation des réponses, la génération de collections de requêtes pour l'automatisation des tests, l'exportation et l'importation de données, et bien plus encore. Il facilite le processus de développement et de débogage des API en offrant une interface intuitive et des fonctionnalités avancées pour simplifier les workflows liés aux API.



Figure 23 : Logo Postman

➤ Framework Flutter

Flutter est un framework de développement open source créé par Google pour construire des applications mobiles, web et de bureau avec un seul code source. Il se

distingue par son langage de programmation Dart, ses widgets personnalisables, ses performances élevées, le Hot Reload pour les modifications rapides, et sa capacité à fonctionner sur plusieurs plateformes.



Figure 24 : Logo Flutter

3.2 ARCHITECTURES MATERIELLES ET LOGICIELLES DE L'APPLICATION

3.2.1 Architecture matérielle

Notre application présente une architecture trois (03) tiers dont les composants sont :

- un serveur de base de données MySQL ;
- un serveur d'application avec Apache ;
- une application mobile.

La figure ci-dessous illustre les interactions entre ces différents composants :

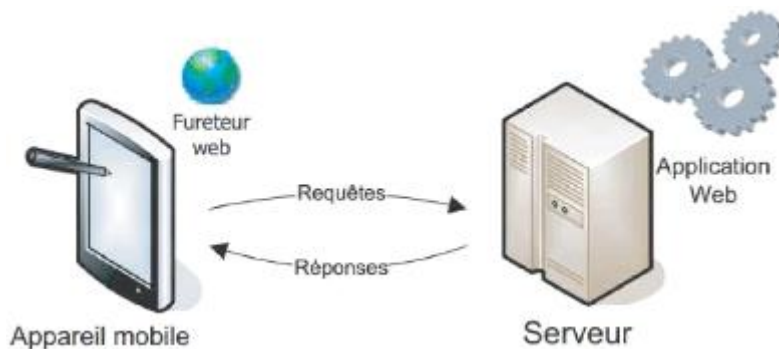


Figure 25: Illustration de l'architecture 3-tiers de notre application

- Serveur d'application : **Apache**

Un serveur d'applications est un logiciel d'infrastructure offrant un [contexte d'exécution](#) pour des composants applicatifs.¹ Il a pour rôle d'écouter les requêtes

¹ Consulter [Serveur d'applications — Wikipédia \(wikipedia.org\)](https://fr.wikipedia.org/wiki/Serveur_d'applications)

émises par les navigateurs, de chercher la page demandée et de la renvoyer. Pour notre projet, nous utiliserons le serveur Apache . Il dispose d'une grande communauté et support disponible en cas de problème. Il est compatible avec tous les systèmes d'exploitation comme Windows, Linux et Mac OSX. Il est léger et rapide.

- Serveur de base de données : **MySQL**

Un serveur de base de données désigne la machine physique sur laquelle tourne le Système de Gestion de Base de Données. Pour notre projet, nous avons utilisé MySQL pour plusieurs raisons que nous avons cités plus haut.

3.2.2 Architecture logicielle

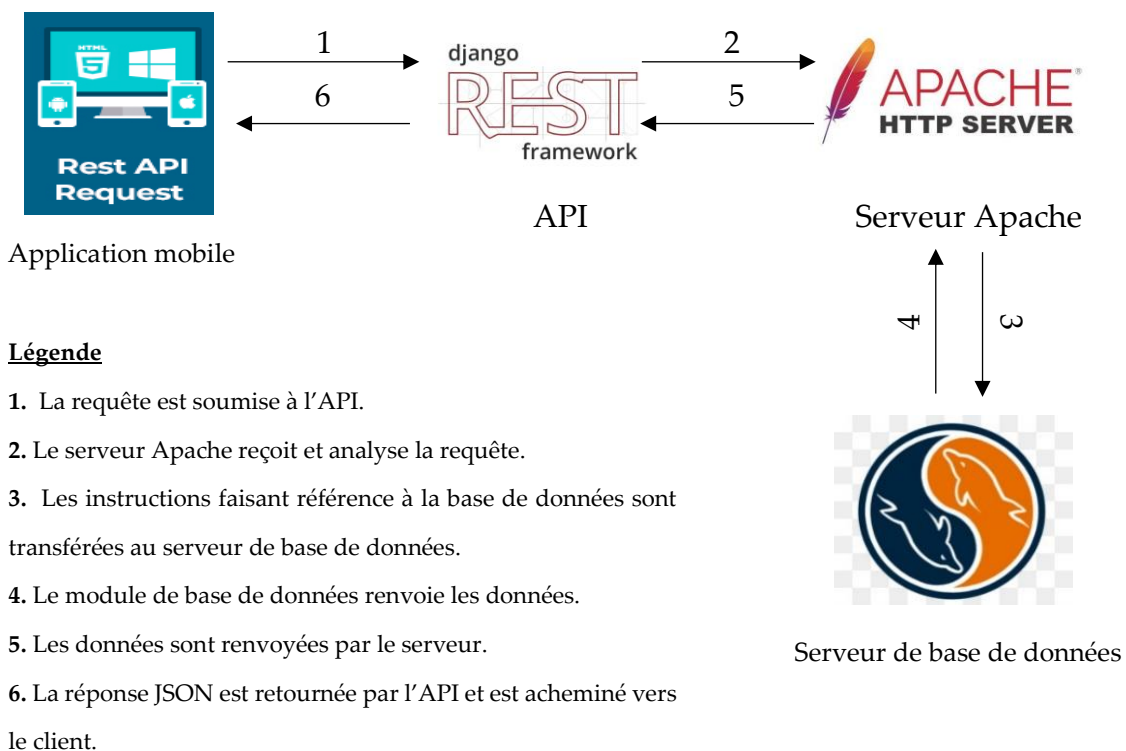


Figure 25 : Architecture logicielle de l'application

3.3 SECURITE DE L'APPLICATION

La sécurité est de mise tant au niveau de l'application mobile que de l'API. Nous vérifions l'authenticité du numéro de téléphone à l'inscription ou à la connexion d'un compte utilisateur. Lors de toute transaction vers l'API, l'application mobile envoie à celui-ci un token² qui lui permet de vérifier son identité. L'hébergeur assure la disponibilité des données servies par l'API. L'hébergeur final sur lequel sera déployée l'application assure la disponibilité et aussi une autre couche de sécurité en plus de celle mise en place pendant la phase de développement.

3.4 MISE EN PLACE DE LA BASE DE DONNEES

Habituellement, le script de création de base de données s'écrit dans le langage SQL. Mais Django propose les migrations pour gérer la création de bases de données. Une migration est une classe qui va contenir la définition d'une table ou altération de la définition de cette table. Elle permet de créer et de mettre à jour un schéma de bases de données. Nous choisissons les migrations plutôt que la création grâce au script SQL car :

- Évolution de la structure : Permet de modifier la base de données au fil du temps sans la recréer manuellement.
- Maintenabilité : Facilite la collaboration d'équipe grâce à des scripts versionnés pour les modifications de base de données.
- Sécurité des données : Réduit les erreurs humaines lors des modifications de schéma complexes.
- Reproductibilité : Assure une cohérence entre les environnements de développement, de test et de production.
- Gestion des versions : Permet de suivre l'historique des changements de la base de données pour le débogage et la compréhension.

² Une chaîne de caractère stockée dans l'application mobile après authentification et qui identifie un utilisateur de façon unique

- Facilité de déploiement : Simplifie les mises à jour de base de données lors du déploiement de nouvelles versions.
- Gestion des données initiales : Peut être utilisé pour insérer des données de démonstration ou des configurations par défaut.

Dans le fichier `settings.py` de notre projet Django, nous avons manipulé les paramètres de la base de données. Vous pouvez choisir parmi plusieurs moteurs de base de données pris en charge, tels que PostgreSQL, MySQL, SQLite, etc. Configurez également les informations de connexion, telles que le nom de la base de données, le nom d'utilisateur et le mot de passe.

Code de quelques migrations

```
class Migration(migrations.Migration):
    dependencies = [
        ("core", "0003_remove_assurance_created_by"),
    ]

    operations = [
        migrations.AlterField(
            model_name="assurance",
            name="deleted",
            field=models.BooleanField(
                db_index=True,
                default=False,
                help_text="Designates if object is deleted.",
                verbose_name="deleted status",
            ),
        ),
        migrations.AlterField(
            model_name="consultation",
            name="deleted",
            field=models.BooleanField(
                db_index=True,
                default=False,
                help_text="Designates if object is deleted.",
                verbose_name="deleted status",
            ),
        ),
    ]
```

```
),
migrations.AlterField(
    model_name="emploitemps",
    name="deleted",
    field=models.BooleanField(
        db_index=True,
        default=False,
        help_text="Designates if object is deleted.",
        verbose_name="deleted status",
    ),
),
migrations.AlterField(
    model_name="etablissementsoins",
    name="deleted",
    field=models.BooleanField(
        db_index=True,
        default=False,
        help_text="Designates if object is deleted.",
        verbose_name="deleted status",
    ),
),
migrations.AlterField(
    model_name="examen",
    name="deleted",
    field=models.BooleanField(
        db_index=True,
        default=False,
        help_text="Designates if object is deleted.",
        verbose_name="deleted status",
    ),
),
migrations.AlterField(
    model_name="jour",
    name="deleted",
    field=models.BooleanField(
        db_index=True,
        default=False,
        help_text="Designates if object is deleted.",
        verbose_name="deleted status",
    ),
),
migrations.AlterField(
    model_name="rendezvous",
```

```
name="deleted",
field=models.BooleanField(
    db_index=True,
    default=False,
    help_text="Designates if object is deleted.",
    verbose_name="deleted status",
),
),
```

PARTIE 4 :
EXPLOITATION

4 EXPLOITATION (GUIDE D'EXPLOITATION)

Le guide d'exploitation est un document qui aidera le patient lors de la configuration de l'application. Il décrit les configurations matérielles et logicielles de l'application ainsi que son déploiement et suivi.

4.1 CONFIGURATION LOGICIELLE ET MATERIELLE

4.1.1 Configuration logicielle

4.1.1.1 Configuration logicielle du serveur

Pour le déploiement du système sur le serveur, ce dernier doit posséder une certaine configuration nécessaire au bon fonctionnement du système. Voici ci-dessous les configurations nécessaires.

- Le serveur web Apache 2 ;
- MySQL installé.

4.1.1.2 Configuration logicielle de l'appareil mobile

L'application mobile pourra être installée sur les téléphones Ios, Windows Phone et Android ayant les versions minimales que voici : 4.0 pour Android, Ios 10 pour les iPhones et la version 8 pour les Windows Phone.

4.1.2 Configuration matérielle

4.1.2.1 Configuration matérielle du serveur

Le bon fonctionnement du système requiert pour le serveur, au minimum les caractéristiques suivantes :

- **Processeur** : 2.2 GHz
- **Mémoire RAM** : 8 Go
- **Disque dur** : 500 Go.

En outre, le serveur doit être connecté à internet, et donc disponible à tout moment.

4.1.2.2 Configuration matérielle du téléphone portable

Du côté matériel, l'appareil mobile (téléphone ou tablette Android), doit avoir, pour un fonctionnement optimal, au minimum les spécifications suivantes :

- **Processeur** : Dual Core
- **Mémoire RAM** : 1.5 Go
- **Mémoire interne** : Au moins 1 Go de libre

4.2 DEPLOIEMENT ET SUIVI

4.2.1 Déploiement de l'application

- Création d'un Compte Développeur Google Play :

Si vous n'avez pas encore de compte développeur Google Play, créez-en un en vous rendant sur la [Console Play](#).

Vous devrez payer des frais uniques pour créer un compte développeur.

- Préparation de l'APK :

Générez un fichier APK signé pour votre application. Assurez-vous d'utiliser le même certificat que celui que vous avez utilisé pendant le développement.

- Accès à la Console Play :

Connectez-vous à la [Console Play](#) en utilisant votre compte développeur Google Play.

- Création d'une Fiche de l'Application :

- Cliquez sur "Créer une application".
- Choisissez la langue et le nom de l'application. Cela peut être le nom que vous souhaitez afficher sur le Play Store.

- Remplir les Détails de l'Application :

Remplissez les informations requises comme la description, les captures d'écran, l'icône, la catégorie, etc.

- Publication de l'APK :

- Sous la section "Version de production", cliquez sur "Créer une version".
- Téléchargez votre fichier APK signé.

- Remplir les Informations de Version :

Entrez le numéro de version et le code de version de votre application. Ajoutez des notes de version pour informer les utilisateurs des nouvelles fonctionnalités et des corrections.

Autres :

➤ Gestion des Pays et des Appareils :

Vous pouvez choisir dans quels pays votre application sera disponible et quels types d'appareils la prendront en charge.

➤ Vérification des Politiques et des Contenus :

Vérifiez que votre application respecte les [politiques de distribution de Google Play](#) et qu'elle ne viole pas les droits d'auteur ni les autres règles.

➤ Examen de l'APK :

Votre APK sera examiné par Google pour vérifier s'il respecte les directives et les règlements.

➤ Publication :

Une fois l'APK approuvé, cliquez sur "Publier" pour que votre application soit disponible sur le Google Play Store.

Publication des Mises à Jour :

À l'avenir, si vous apportez des mises à jour à votre application, vous devrez créer de nouvelles versions dans la console et suivre des étapes similaires pour les soumettre.

➤ Disponibilité pour les Utilisateurs :

Après la publication, votre application sera disponible sur le Google Play Store pour les utilisateurs à travers les pays et les appareils que vous avez sélectionnés.

4.2.2 Suivi de l'application

Le suivi d'une application après son déploiement permet d'éviter au cours de son utilisation des problèmes qui pourraient entraîner une indisponibilité de l'application. Les principales actions à mener pour le suivi d'une application sont : la sauvegarde de la base de données ainsi que la gestion des erreurs.

- **Sauvegarde de la base de données :** Il faudra dans cette partie définir une politique de sauvegarde des données d'application car son absence pourrait coûter chère à la vie d'une entreprise. Il faudra par conséquent aussi instaurer un moyen de restauration en cas de panne ou d'indisponibilité du système.
- **La gestion des incidents ou encore le cahier d'évènements :** Pendant tout le cycle de vie de l'application, l'administrateur doit s'assurer du bon déroulement de toutes les activités à travers l'application. Il sera donc consigné dans un document les éventuelles erreurs qui pourraient survenir lors de l'utilisation du système ainsi que leur explication afin de mieux s'en sortir lorsqu'on y fera face. En cas de panne ou tout autre problème dans le système, il est prévu un document dans lequel seront mentionnées les erreurs et incidents. Ce recueil servira de support pour une maintenance. La structure du document dans lequel seront mentionnés ce genre d'informations se présente comme suit :

Tableau 10 : Cahier des évènements

DATE ET HEURE	LIBELLE DE L'ERREUR	DESCRIPTION	SOLUTIONS APPORTEES

4.3 MAINTENANCE

Dans cette partie, nous énumérerons les éventuelles erreurs que l'application pourrait générer au cours de son utilisation afin de mener une meilleure action de maintenance.

Tableau 11 : Codes d'erreur, description et actions correctives

CODE D'ERREUR	DESCRIPTION	ACTION A MENER
404 : Not Found	La page demandée est introuvable	Vérifier si le chemin vers la page demandée est correct
401 : Unauthorized	Paramètres de connexion non corrects, donc pas d'accès à la ressource demandée	Vérifier les paramètres de connexion ainsi que les droits
405 : Method Not Allowed	La méthode exécutée n'est pas définie pour la route exécutée	Définir la méthode pour la route en question
500 : Internal Server Error	Erreur interne du serveur	Vérifier si le serveur est bien démarré et vider le cache du système
ERR_INTERNET_DISCONNECTED	Survient quand l'application n'est pas connectée à internet	Vérifier votre connexion internet
Autres		Contactez l'administrateur

PARTIE 5 :
GUIDE D'UTILISATION

5 GUIDE D'UTILISATION

5.1 DESCRIPTION TEXTUELLE DU LOGICIEL

Simplifiez la gestion de vos rendez-vous médicaux avec notre application conviviale. Choisissez parmi une variété de médecins et de spécialités, fixez des rendez-vous en quelques clics et recevez des rappels utiles. Gérez facilement votre calendrier médical, en toute sécurité et confidentialité. Téléchargez maintenant pour prendre le contrôle de votre santé.

5.2 PLAN DE NAVIGATION

Voici l'organigramme qui constitue le plan de navigation détaillé pour notre application, illustrant la hiérarchie des différentes sections et fonctionnalités afin de fournir une expérience utilisateur fluide et intuitive.

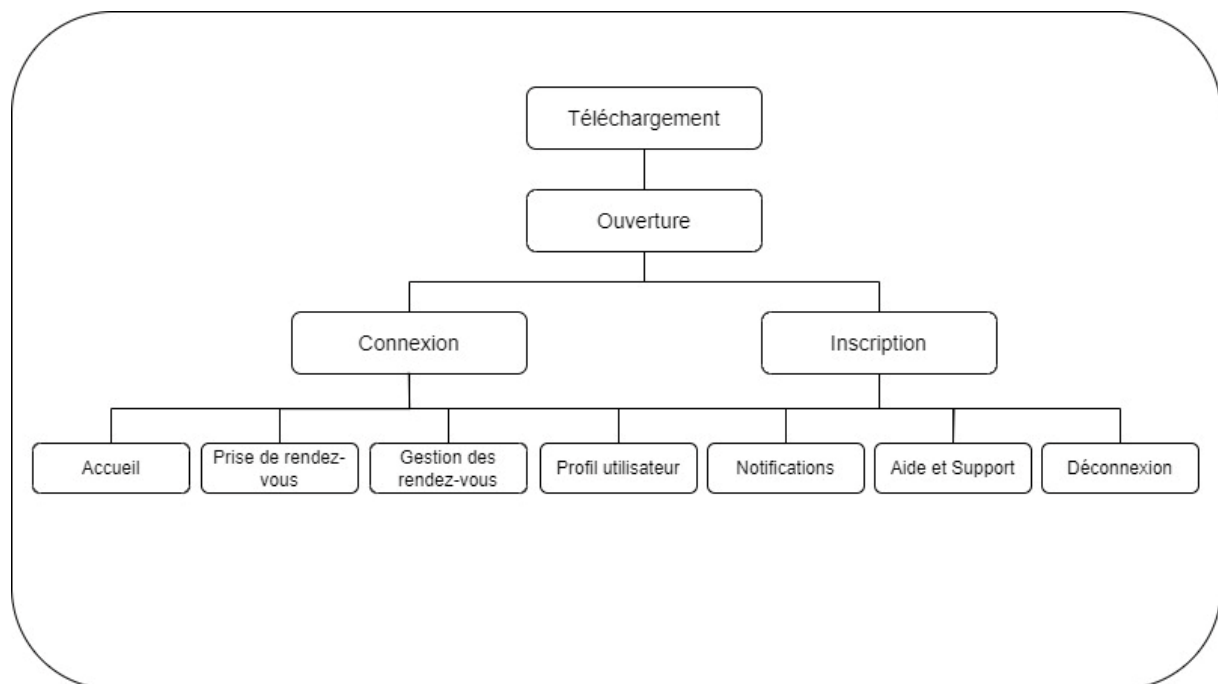


Figure 26 : Plan de navigation de l'application

5.3 PRESENTATION DES DIFFERENTES INTERFACES DE L'APPLICATION

5.3.1 Quelques masques de saisies

- Page d'accueil



Figure 27 : Page d'accueil de l'application mobile

➤ Page de connexion



Figure 28 : Page de connexion de l'application mobile

➤ Page d'inscription

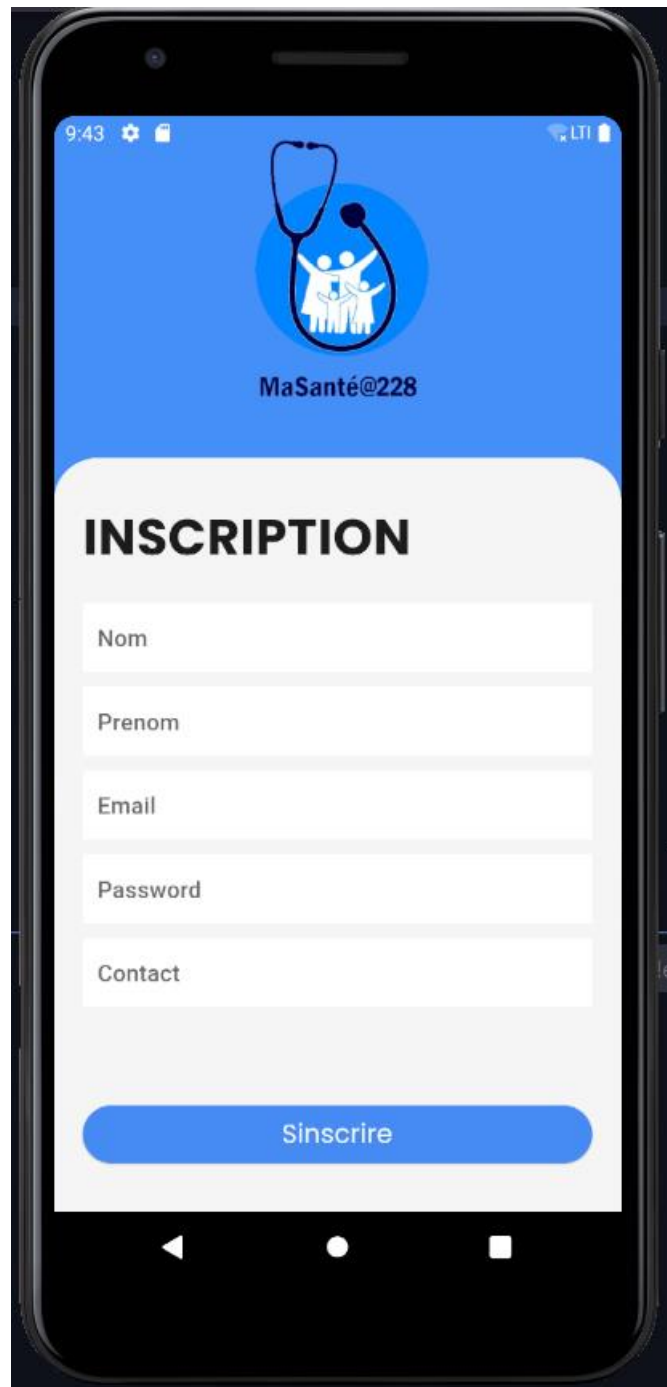


Figure 29 : Page d'inscription à travers l'application mobile

5.3.2 Quelques masques de codes sources

➤ Code de la connexion

```
class SignInPage extends StatefulWidget {  
  const SignInPage({super.key});
```

```
@override  
State<SignInPage> createState() => _SignInPageState();  
}
```

```
class _SignInPageState extends State<SignInPage> {  
  late UserProvider userProvider;  
  String email = "";  
  final key = GlobalKey<FormState>();  
  @override  
  void initState() {  
    userProvider = context.read<UserProvider>();  
    userProvider.addListener(listener);  
    super.initState();  
  }
```

```
@override  
void dispose() {  
  userProvider.removeListener(listener);  
  super.dispose();  
}
```

```
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    backgroundColor: kPrimaryColor,  
    body: SingleChildScrollView(  
      child: Form(  
        key: key,  
        child: Column(  

```



```
children: [
  SizedBox(
    height: kSize(context).height / 3.2,
    child: Center(
      child: Image.asset(
        kImagePath(imageName: "logo.png"),
      ),
    ),
  ),
  Container(
    height: kSize(context).height - kSize(context).height / 3.2,
    padding:
      const EdgeInsets.symmetric(vertical: 30, horizontal: 20),
    width: double.infinity,
    decoration: BoxDecoration(
      color: Colors.grey[100],
      borderRadius: const BorderRadius.only(
        topLeft: Radius.circular(30),
        topRight: Radius.circular(30))),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      mainAxisAlignment: MainAxisAlignment.min,
      children: [
        const TextWidget(
          data: "CONNEXION",
          fontSize: 35,
          fontWeight: FontWeight.bold,
        ),
        TextWidget(
```

```
      data: "Bienvenue",
      color: kGreyColor,
    ),
    const SizedBox(
      height: 10,
    ),
    Padding(
      padding: const EdgeInsets.symmetric(vertical: 10),
      child: InputWidget(
        placeholder: "Email",
        onChanged: (p0) {
          email = p0;
        },
        validator: (p0) {
          if (p0 == null || p0.isEmpty) {
            return "le champ ne peut etre vide";
          }
          if (!p0.contains("@")) {
            return "le mail n'est pas valide";
          }
        },
      ),
    ),
    Padding(
      padding: const EdgeInsets.symmetric(vertical: 25),
      child: Center(
        child: ButtonWidget(
          width: kSize(context).width / 2,
          child: "Se connecter",
```

```
        onPressed: () {  
          if (key.currentState!.validate()) {  
            userProvider.signInUser(email: email);  
          }  
          ;  
        })),  
      ),  
      Center(  
        child: TextWidget(  
          data: "Mot de passe oubli ",  
          color: kGreyColor,  
        )),  
      ],  
    ),  
  )  
  ],  
),  
),  
),  
);  
}
```

```
listener() {  
  final st = userProvider.status;  
  if (st == Status.loading) {  
    showLoaderDialog(context);  
  }  
  if (st == Status.loaded) {  
    Navigator.pop(context);  
  }  
}
```

```
/* Navigator.pushReplacement(
    context,
    MaterialPageRoute(
        builder: (_) => const EmailVerify())) */
}
if (st == Status.error) {
    Navigator.pop(context);
    kSnackBar(context, 'une erreur sest produite', color: Colors.red);
}
}
}
```

➤ Code de l'inscription

```
class SignUpPage extends StatefulWidget {
    const SignUpPage({super.key});

    @override
    State<SignUpPage> createState() => _SignUpPageState();
}
```

```
class _SignUpPageState extends State<SignUpPage> {
```

```
late UserProvider userProvider;

String nom = "";
String prenom = "";
String email = "";
String contact = "";
final key = GlobalKey<FormState>();

@override
void initState() {
```

```
userProvider = context.read<UserProvider>();  
userProvider.addListener(listener);  
super.initState();  
}
```

```
@override  
void dispose() {  
  userProvider.removeListener(listener);  
  super.dispose();  
}
```

```
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    backgroundColor: kPrimaryColor,  
    body: SingleChildScrollView(  
      child: Column(  
        children: [  
          SizedBox(  
            height: kSize(context).height / 3.2,  
            child: Center(  
              child: Image.asset(  
                kImagePath(imageName: "logo.png"),  
              ),  
            ),  
          ),  
          Container(  
            height: kSize(context).height - kSize(context).height / 3.2,  
            padding: const EdgeInsets.symmetric(vertical: 30, horizontal: 20),
```

```
width: double.infinity,
decoration: BoxDecoration(
  color: Colors.grey[100],
  borderRadius: const BorderRadius.only(
    topLeft: Radius.circular(30),
    topRight: Radius.circular(30))),
child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  mainAxisAlignment: MainAxisAlignment.min,
  children: [
    const TextWidget(
      data: "INSCRIPTION",
      fontSize: 35,
      fontWeight: FontWeight.bold,
    ),
    const SizedBox(
      height: 10,
    ),
    Padding(
      padding: const EdgeInsets.symmetric(vertical: 10),
      child: InputWidget(placeholder: "Nom",
        onChanged: (p0) {
          nom = p0;
        },
        validator: (p0) {
          if (p0 == null || p0.isEmpty) {
            return "le champ ne peut etre vide";
          }
        }
      )
    )
  ]
)
```

```
    },  
  ),  
  InputWidget(placeholder: "Prenom",  
    onChanged: (p0) {  
      prenom = p0;  
    },  
    validator: (p0) {  
      if (p0 == null || p0.isEmpty) {  
        return "le champ ne peux etre vide";  
      }  
    },  
  ),
```

```
Padding(  
  padding: const EdgeInsets.symmetric(vertical: 10),  
  child: InputWidget(placeholder: "Email",  
    onChanged: (p0) {  
      email = p0;  
    },  
    validator: (p0) {  
      if (p0 == null || p0.isEmpty) {  
        return "le champ ne peux etre vide";  
      }  
      if (!p0.contains("@")) {  
        return "le mail n'est pas valide";  
      }  
    },  
  ),
```

```
)  
,
```

```
Padding(  
  padding: const EdgeInsets.symmetric(vertical: 10),  
  child: InputWidget(placeholder: "Contact",  
    onChanged: (p0) {  
      contact = p0;  
    },  
    validator: (p0) {  
      if (p0 == null || p0.isEmpty) {  
        return "le champ ne peut etre vide";  
      }  
    },  
  ),  
,
```

```
Expanded(  
  child: Align(  
    alignment: Alignment.bottomCenter,  
    child: ButtonWidget(  
      //width: kSize(context).width / 2,  
      child: "Sinscrire",  
      onPressed: () {  
  
        if (key.currentState!.validate()) {  
          userProvider.signUpUser(nom: nom, contact: 'contact', email:  
'email', prenom: 'prenom');
```



```
        }

        /* Navigator.pushReplacement(
            context,
            MaterialPageRoute(
                builder: (_) => const SignUpPage()) */

        ;

/* Navigator.pushAndRemoveUntil(
    context,
    slidableRoute(nextPage: const HomePage()),
    (route) => false);*/
/* Navigator.push(
    context,
    MaterialPageRoute(
        builder: (_) => const HomePage()));*/

    },
),
),
),
],
),
)
],
),
),
);
}
```

```
listener() {
```

```
final st = userProvider.status;

if (st == Status.loading) {
  showLoaderDialog(context);
}

if (st == Status.loaded) {
  Navigator.pop(context);
  //Navigator.push.....
}

if (st == Status.error) {
  Navigator.pop(context);
  kSnackBar(context, 'une erreur sest produite', color: Colors.red);
}
}
}
```

CONCLUSION

Notre stage de trois (03) mois au sein de **a-ICT Group** était destiné à élaborer et mettre en place une application mobile qui facilitera la gestion des prises de rendez-vous à temps réel. Nous sommes parvenus à implémenter cette application grâce à divers outils et langages tels que UML, 2TUP, les Framework Django et Flutter. Nous avons appliqué en entreprise nos connaissances acquises au cours de notre formation à IAI-TOGO et par la même occasion aiguiser et enrichir ces connaissances. L'application mise en place répondant déjà à un certain nombre de problèmes, elle pourrait être encore étendue à d'autres perspectives telles que la consultation, les résultats d'analyse, les assurances et encore un plus grand périmètre des manquements de ce domaine. Mais étant donné que toute application est destinée à évoluer dans le temps, nous l'améliorerons en tenant compte du retour que feront les utilisateurs sur la première version de notre application.

BIBLIOGRAPHIE INDICATIVE

➤ **Ouvrages**

- ✓ UML 2 par la pratique : Pascal Roques
- ✓ UML en action : Pascal Roques
- ✓ Developper votre première application android :
Openclassclassrooms
- ✓ Introduction à UML 2 : Pierre Géard

➤ **Notes de cours**

- ✓ Introduction au Génie logiciel : **M. SEWAVI**
- ✓ Plateforme et outil de développement : **M. AZIATCHI**
- ✓ Diagramme de cas d'utilisation : **M. AMEVOR**

➤ **Ancien mémoire consulté**

- ✓ « CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION MOBILE
DE COLLECTE DES ORDURES INTEGRANT LES SYSTEMES
D'INFORMATION GEOGRAPHIQUES : CAS DE LA VOIRIE de POUH
Pouwèdéou Crésus

WEBOGRAPHIE INDICATIVE

- [1] «DTCare - Doctor Appointment Flutter App UI KIT Template - Patient App Preview - CodeCanyon,» 14 Août 2023. [En ligne]. Available: http://preview.codecanyon.net/item/dtcare-doctor-appointment-flutter-app-ui-kit-template-patient-app/full_screen_preview/39991031?_ga=2.36282345.81915287.1686226360-738192651.1686226360.
- [2] «ICONS8,» 20 JUILLET 2023. [En ligne]. Available: <https://icons8.com/icons/set/icons8>.
- [3] «flutterawesome,» 20 JUILLET 2023. [En ligne]. Available: <https://flutterawesome.com/building-production-ready-healthcare-doctor-consult-android-and-ios-app-ui-using-flutter/>.
- [4] «Remove,» 10 août 2023. [En ligne]. Available: <https://www.remove.bg/fr>.
- [5] «stackoverflow,» [En ligne]. Available: <https://stackoverflow.com/>.
- [6] «flutter dev,» [En ligne]. Available: <https://flutter.dev/>.
- [7] «flutter pub dev,» [En ligne]. Available: <https://pub.dev/>.

TABLE DES MATIERES

DEDICACES.....	i
REMERCIEMENTS	ii
SOMMAIRE.....	iv
RESUME	vi
ABSTRACT.....	vii
GLOSSAIRE.....	viii
LISTE DES FIGURES.....	ix
LISTE DES TABLEAUX.....	xi
LISTE DES PARTICIPANTS AU PROJET	xii
INTRODUCTION.....	1
1 CAHIER DES CHARGES.....	3
1.1 PRESENTATIONS.....	3
1.1.1 Brève présentation de l'IAI-TOGO	3
1.1.1.1 Formation à l'IAI-TOGO	3
1.1.1.2 Plan de localisation	3
1.1.2 Présentation du cadre de stage.....	4
1.1.2.1 Statut	4
1.1.2.2 Mission.....	4
1.1.2.3 Activités.....	5
1.1.2.4 Quelques réalisations.....	5
1.1.2.5 Service d'accueil	6
1.1.2.6 Plan de localisation	7
1.2 THEME DU STAGE	8
1.2.1 Présentation du sujet.....	8
1.2.2 Problématique du sujet.....	8
1.2.3 Intérêt du sujet	9
1.2.3.1 Objectifs	9

1.2.3.1.1	Objectif général	9
1.2.3.1.2	Objectifs spécifiques	9
1.2.3.2	Résultats	10
1.3	ETUDE DE L'EXISTANT	11
1.4	CRITIQUE DE L'EXISTANT.....	12
1.5	PROPOSITIONS ET CHOIX DE SOLUTIONS	13
1.5.1	Evaluation technique des solutions	13
1.5.1.1	Première solution	13
1.5.1.1.1	Avantages	13
1.5.1.1.2	Inconvénients	13
1.5.1.1.3	Ressources matérielles	14
1.5.1.1.4	Ressources logicielles	14
1.5.1.2	Deuxième solution	15
1.5.1.2.1	Avantages	15
1.5.1.2.2	Inconvénients	15
1.5.1.2.3	Ressources matérielles	16
1.5.1.2.4	Ressources logicielles	16
1.5.2	Evaluation financière des solutions proposées	17
1.5.2.1	Première solution	17
1.5.2.2	Deuxième solution	17
1.5.3	Choix de la solution	18
1.6	PLANNING PREVISIONNEL DE REALISATION.....	18
2	ANALYSE ET CONCEPTION.....	21
2.1	CHOIX DE LA METHODE D'ANALYSE ET JUSTIFICATION	21
2.1.1	Langage UML	21
2.1.2	Diagrammes UML.....	22
2.1.3	Processus 2TUP.....	23
2.1.4	Phases du processus 2TUP.....	24
2.2	CHOIX DE L'OUTIL DE MODELISATION ET JUSTIFICATION.....	25

2.3	ETUDE DETAILLEE DE LA SOLUTION	26
2.3.1	Etude préliminaire	27
2.3.2	Capture des besoins fonctionnels	27
2.3.2.1	Diagramme des cas d'utilisation.....	27
2.3.2.1.1	Les acteurs.....	27
2.3.2.1.2	Les cas d'utilisation	27
2.3.2.1.3	Modélisation du diagramme de cas d'utilisation	28
2.3.2.1.4	Description textuelle de certains cas d'utilisation	29
2.3.2.2	Diagramme de séquences	34
2.3.2.3	Diagramme d'activités	37
2.3.3	Capture des besoins techniques	40
2.3.4	Conception	40
2.3.4.1	Diagramme de déploiement.....	40
2.3.4.2	Diagramme de classes	41
2.3.4.3	Diagramme de composants	42
3	REALISATION ET MISE EN ŒUVRE	45
3.1	MATERIELS ET LOGICIELS UTILISES.....	45
3.1.1	Matériels utilisés.....	45
3.1.2	Logiciels utilisés.....	45
3.2	ARCHITECTURES MATERIELLES ET LOGICIELLES DE L'APPLICATION	
	50	
3.2.1	Architecture matérielle	50
3.2.2	Architecture logicielle.....	51
3.3	SECURITE DE L'APPLICATION	52
3.4	MISE EN PLACE DE LA BASE DE DONNEES.....	52
4	EXPLOITATION (GUIDE D'EXPLOITATION)	57
4.1	CONFIGURATION LOGICIELLE ET MATERIELLE	57
4.1.1	Configuration logicielle	57
4.1.1.1	Configuration logicielle du serveur.....	57

4.1.1.2	Configuration logicielle de l'appareil mobile	57
4.1.2	Configuration matérielle	57
4.1.2.1	Configuration matérielle du serveur.....	57
4.1.2.2	Configuration matérielle du téléphone portable.....	58
4.2	DEPLOIEMENT ET SUIVI.....	58
4.2.1	Déploiement de l'application	58
4.2.2	Suivi de l'application	59
4.3	MAINTENANCE	60
5	GUIDE D'UTILISATION	63
5.1	DESCRIPTION TEXTUELLE DU LOGICIEL.....	63
5.2	PLAN DE NAVIGATION	63
5.3	PRESENTATION DES DIFFERENTES INTERFACES DE L'APPLICATION	
	63	
5.3.1	Quelques masques de saisies.....	64
5.3.2	Quelques masques de codes sources.....	66
	CONCLUSION	78
	BIBLIOGRAPHIE INDICATIVE	79
	WEBOGRAPHIE INDICATIVE	80
	TABLE DES MATIERES.....	81