

Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et informatique
Filière : Informatique

RAPPORT DE MINI-PROJET

Option : **Ingénierie des Systèmes d'Information**

THEME :

**Conception et réalisation d'un système de
gestion des donneurs d'une banque de sang**

Étudiant(e) : **« Latreche Sarra »**

« Ould Mamar Mansour Nasr Allah »

Encadrant(e) : **« Dr. Midoun Mohammed »**

Résumé

Les situations d'urgence, comme les accidents, créent un besoin immédiat et critique en produits sanguins. En plus des besoins urgents, les progrès de la médecine ont de nombreux traitements et chirurgies électives en cours.

Le don de sang est une procédure médicale qui implique la transfusion de sang d'une personne en bonne santé et consentante à une personne malade nécessitant du sang et qui peut sauver des millions de vies chaque année. Le principal défi auquel fait face le Centre de Transfusion Sanguine (CTS) est l'incapacité de fournir du sang sûr et adéquat en temps opportun à tous les patients qui en ont besoin en raison de pénuries constantes de sang. Sur la base de cela, il est logique d'avoir un dossier des donneurs réguliers et en même temps d'explorer le potentiel de l'utilisation de la technologie pour avoir une stratégie de recrutement efficace et de maintenir la fidélité des donneurs. Le principal objectif du projet est de fournir un logiciel qui facilite la gestion des collectes de don de sang en permettant la gestion des bénévoles et des donneurs, la localisation des CTS, l'identification des lieux de collectes ainsi que le contact avec les partenaires, les associations, les donneurs et les CTS. Ceci, sans parler de la gestion des donneurs de sang et de plaquettes et des informations sur les donneurs et leurs disponibilités. Dans le logiciel que nous devons réaliser, les informations peuvent être partagées entre les CTS et les associations et il doit permettre l'importation et l'exportation de tables.

Mots-clés :

Don de sang, UML, transfusion sanguine, gestion de donneurs

الملخص

حالات الطوارئ ، مثل الحوادث ، تخلق حاجة فورية وحاسمة لدم معين. بالإضافة إلى الاحتياجات العاجلة ، فإن التقدم في الطب يتضمن العديد من العلاجات والعمليات الجراحية الاختيارية

يعتبر التبرع بالدم إجراءً طبيًا يتضمن نقل الدم من شخص سليم وموافق إلى شخص مريض يحتاج إلى الدم ويمكن أن ينقذ حياة الملايين كل عام. التحدي الرئيسي الذي يواجهه

CTS

هو عدم القدرة على توفير الدم الآمن والكافي في الوقت المناسب لجميع المرضى الذين يحتاجون إليه بسبب النقص المستمر في الدم. بناءً على ذلك ، من المنطقي أن يكون لديك سجل للمانحين المنتظمين وفي نفس الوقت استكشاف إمكانية استخدام التكنولوجيا كاستراتيجية توظيف واستبقاء للمانحين. الهدف الرئيسي من المشروع هو توفير برنامج يساعد في إدارة مجموعات التبرع بالدم: إدارة المتطوعين والمتبرعين ، وموقع

CTS

بالإضافة إلى إدارة المتبرعين CTS والشركاء ، والجمعيات ، وتسهيل الاتصال مع الشركاء والجمعيات والمتبرعين و بالدم والصفائح الدموية ، ومعلومات عن المتبرعين وتوافرهم. في البرنامج ، يمكن مشاركة المعلومات بين

CTS

والجمعيات ويجب أن تسمح أيضًا باستيراد الجداول وتصديرها

Summary

Emergencies, such as accidents, create an immediate and critical need for specific blood. In addition to urgent needs, advances in medicine have many treatments and elective surgeries going on.

Blood donation is a medical procedure that involves the transfusion of blood from a healthy, consenting person to a sick person in need of blood and which can save millions of lives each year. The main challenge facing the CTS is the inability to provide safe and adequate blood on a timely basis to all patients who need it due to constant blood shortages. Based on this, it makes sense to have a record of regular donors and at the same time explore the potential of using technology as a donor recruitment and retention strategy. The main objective of the project is to provide software that helps in the management of blood donation collections: management of volunteers and donors, location of CTS, collection locations, partners, associations, facilitate contact with partners, associations, donors and CTS. Plus the management of blood and platelet donors, information on donors and their availability. In the software the information can be shared between CTS and Associations and it must also allow the import and export of tables.

Keywords :

Blood donation, UML ,blood transfusion, donor management

Remerciement

Tout d'abord nous rendons grâce à Dieu, lui qui nous a permis d'être bien portant afin d'effectuer ce travail du début jusqu'à la fin.

Nous remercions nos parents respectifs pour leurs soutiens .Nos remerciements vont,à notre encadrant Dr Mohammed Midoun qui nous a guidé avec ses orientations, ses conseils et ses critiques tout au long de ce travail en nous laissant la liberté dont on avait besoins.

Nos remerciements vont aussi au membre du jury, pour l'honneur qu'ils nous ont fait en acceptant d'évaluer ce travail et de participer à la soutenance.

Et enfin, nous sommes reconnaissants envers tous les enseignants de l'Université
Abdelhamid Ibn Badis Mostaganem

Liste d'abréviation :

BDD : Binary Decision Diagram.

CTS : centre de transfusion sanguine.

ID : identifiant.

JPEG : Joint Photographic Experts Group.

PNG : Portable Network Graphic.

PSL : produits sanguins labiles.

RUP : Rational Unified Process.

UML : Unified Modeling Language.

Liste des figures

Figure N	Titre de la figure	Page
Figure 1	L'outil StarUML	09
Figure 2	Diagramme de paquetage	10
Figure 3	Diagramme de cas d'utilisation pour le package « Administrateur »	11
Figure 4	Diagramme de cas d'utilisation pour le paquetage « Consultation »	11
Figure 5	Diagramme de cas d'utilisation pour le paquetage « Gestionnaire des donneurs »	12
Figure 6	Diagramme de cas d'utilisation pour le paquetage « Gestionnaire d'appels	13
Figure 7	Diagramme de cas d'utilisation pour le paquetage « Gestionnaire de la collecte »	13
Figure 8	Diagramme de séquence du cas « S'authentifier »	15
Figure 9	Diagramme de séquence du cas « Gérer donneurs uniquement »	17
Figure 10	Diagramme de séquence du cas : « Ajouter une collecte / donneurs_collecte »	19
Figure 11	Diagramme de classe	21
Figure 12	Les tables de la base de données	24
Figure 13	L'architecture du système	26
Figure 14	La page login	29
Figure 15	La page accueil	30
Figure 16	La page ajout	31
Figure 17	La page ajouter un donneur	31
Figure 18	La liste des donneurs	32

Figure 19	La page ajouter une collecte	33
Figure 20	La page ajouter donneur_collecte	33
Figure 21	La page appels donneur	34

Table des matières

RESUME.....	2
الملخص	3
SUMMARY	4
REMERCIEMENT	5
LISTE D'ABRÉVIATION :.....	6
LISTE DES FIGURES.....	7
TABLE DES MATIERES	1
INTRODUCTION GENERALE.....	4
CHAPITRE 1 : LE DON DE SANG	5
1.1. INTRODUCTION	5
1.2. LE SANG	5
1.3. LES PLAQUETTES.....	5
1.4. LE PLASMA.....	5
1.5. L'IMPORTANCE DU DON SANG	6
1.6. L'IMPORTANCE DU DON DE PLASMA.....	6
1.7. L'IMPORTANCE DU DON DE PLAQUETTES.....	6
1.8. CRITERES POUR FAIRE UN DON DE SANG	6
1.9. CRITERES POUR FAIRE UN DON DE PLAQUETTES.....	6
1.10. CRITERES POUR FAIRE UN DON DE PLASMA.....	7
1.11. LES CENTRES DE TRANSFUSION SANGUINE (CTS).....	7
1.11.1. La banque de sang.....	7
1.11.2. La gestion des donneurs de sang et de plaquettes dans les CTS.....	7
1.12. CONCLUSION.....	7
CHAPITRE 2: ANALYSE DES BESOINS ET CONCEPTION	8
2.1. INTRODUCTION	8
2.2. SPECIFICATION DES BESOINS.....	8
2.2.1. LES BESOINS FONCTIONNELS.....	8
2.2.2. LES BESOINS NON FONCTIONNELS.....	9

2.3. METHODOLOGIE ET APPROCHE.....	9
2.3.1. Le langage UML	9
2.3.2. Démarche suivie.....	10
2.3.3. L'outil StarUML.....	10
2.3.4. La méthode RUP.....	10
2.4. ANALYSE DES BESOINS	11
2.4.1. Les diagrammes de paquetages	11
2.4.2. Les diagrammes de cas d'utilisation.....	11
2.4.3. Diagrammes de séquences.....	15
2.5. LA CONCEPTION.....	21
2.5.1. Diagramme de classe	21
2.5.2. Description textuelle	22
2.6. CONTENU DE LA BD.....	23
2.7. CONCLUSION	25
CHAPITRE 3: IMPLEMENTATION ET EXPERIMENTATION	26
3.1. INTRODUCTION	26
3.2. CONFIGURATION REQUISE :	26
3.3. ARCHITECTURE DU SYSTEME :	26
3.4. OUTILS DE DEVELOPPEMENT	28
3.5. PRESENTATION DES INTERFACES	30
3.5.1. La page de Login.....	30
3.5.2. Page d'accueil (admin)	31
3.5.3. Page Ajout.....	32
3.5.4. Page de la liste des donateurs.....	33
3.5.5. Page collecte.....	33
3.5.6. Ajouter donneur collecte	34
3.5.7. La page appels bénévoles et donateurs.....	35
3.6. CONCLUSION	35
CONCLUSION GENERALE.....	36
ANNEXE 1	38
ANNEXE 2	40
WEBOGRAPHIES	41
BIBLIOGRAPHIE.....	42

Introduction générale

Le sang est nécessaire et vital au fonctionnement du corps humain. Aujourd'hui, il n'existe pas de produit capable de remplacer le sang humain. La générosité des donneurs de sang est donc indispensable. Combien de patients, d'accidentés, d'opérés, de jeunes accouchées ou de nouveau-nés n'auraient pas pu vivre sans transfusion sanguine et vont bien aujourd'hui grâce au sang qu'ils ont reçu ? Le don du sang est donc indispensable à la médecine moderne pour soigner et sauver de nombreuses vies humaines.

Le pourcentage de personnes qui donnent leur sang augmente de jour en jour, car elles sont conscientes de la nécessité de donner leur sang [9]. Le personnel et les infirmières se retrouvent face à une énorme quantité d'informations désordonnées, qui doivent être notées sur un registre, ce qui rend difficile la gestion des donneurs.

Ces problèmes peuvent être résolus en automatisant le système manuel de gestion des banques de sang existant afin de réduire les efforts nécessaires pour rechercher des donneurs de sang. Dans ce rapport nous allons chercher à concevoir un système d'information complet pour la gestion des donneurs de sang et de plaquettes d'une banque de sang. Notre document se compose de trois principaux chapitres : le premier chapitre se focalise sur la présentation du contexte général. Le second chapitre a été consacré à la présentation des besoins, ainsi qu'à l'analyse et à la conception de la base de données. Le dernier chapitre abordera la partie implémentation et expérimentation. Nous concluons ce rapport par une conclusion générale qui résumera le contenu de ce rapport et proposera des perspectives possibles à ce projet.

Chapitre 1 : Le don de sang

1.1. Introduction

En Algérie, toutes les banques de sang reçoivent le sang de divers donneurs, surveillent la base de données des groupes sanguins et envoient le sang nécessaire aux hôpitaux en cas d'urgence. Les donneurs doivent répondre à quelques critères essentiels avant de pouvoir donner leur sang. Il est donc difficile de suivre la base de données des donneurs de sang sans l'utilisation d'un logiciel. Dans ce chapitre, nous allons décrire les concepts inhérents à la problématique du don de sang.

1.2. Le sang

Pulsé par le cœur et circulant dans les vaisseaux à travers l'organisme, le sang oxygène, nourrit, nettoie et défend tous les tissus et les organes de notre corps. Bien que liquide, le sang est un tissu très organisé. Il est composé d'un fluide salé, le plasma, dans lequel circulent trois types de cellules : les globules rouges, les globules blancs et les plaquettes. Pour 1 globule blanc, il y a environ 50 plaquettes et 1000 globules rouges [1].

1.3. Les plaquettes

Les plaquettes sanguines, aussi appelées "thrombocytes", sont issues des mégacaryocytes, grosses cellules produites par la moelle osseuse. En cas de rupture de la paroi d'un vaisseau, ces minuscules cellules se collent dessus et colmatent la brèche. Ensuite, différentes protéines du plasma vont renforcer ce « bouchon » grâce au processus de coagulation [1].

1.4. Le plasma

Le plasma est la partie liquide du sang. Composé à 90 % d'eau salée, il permet aux globules rouges et aux plaquettes de circuler dans le système vasculaire. Il apporte également les substances nutritives aux tissus, dont il recueille en retour les déchets pour les acheminer vers les organes d'élimination [1].

1.5. L'importance du don sang

Il n'existe aucun traitement ni médicament de synthèse capable de se substituer au sang humain et aux produits sanguins labiles (PSL), issus des dons de sang. Cet acte volontaire et bénévole est donc irremplaçable, il est utilisé dans les secours d'urgence, interventions chirurgicales, maladies du sang, cancer, etc. Les besoins en produits sanguins sont quotidiens et ont fortement augmenté au cours des quinze dernières années [11].

1.6. L'importance du don de plasma

Il contient des protéines plasmatiques d'un intérêt thérapeutique majeur. Ces protéines permettent la fabrication de nombreux médicaments (immunoglobulines, facteurs de coagulation, albumine) [2].

1.7. L'importance du don de plaquettes

Les plaquettes aident à soigner les patients atteints de maladies du sang (leucémie, lymphomes...) et subissant des traitements lourds (chimiothérapie, radiothérapie...). Elles permettent d'éviter les risques d'hémorragies mettant en jeu la vie des malades. Le don de plaquettes est donc un acte engagé qui répond à un besoin vital [1].

1.8. Critères pour faire un don de sang

Toute personne âgée de 18 à 60 ans, qui pèse plus de 50 kg et qui est reconnue apte suite à l'entretien, peut donner son sang. Après 50 ans, le premier don est soumis à l'appréciation d'un médecin. Les hommes peuvent donner jusqu'à 4 fois par an et les femmes jusqu'à 3 fois. Entre deux dons de sang, il faut respecter un délai de 12 semaines minimum [1].

1.9. Critères pour faire un don de plaquettes

Il faut avoir entre 18 et 60 ans, peser minimum 50 kilos et avoir un taux de plaquettes suffisamment élevé. Il est alors possible de donner ses plaquettes jusqu'à 6 fois par an, tout en respectant un délai de 8 semaines entre chaque don [1].

1.10. Critères pour faire un don de plasma

Pour faire don de son plasma, il faut avoir entre 18 et 60 ans. Il est possible de donner jusqu'à 24 fois par an, que l'on soit un homme ou une femme, si tant est que l'on respecte un intervalle d'au moins deux semaines entre chaque prélèvement [1].

1.11. Les centres de transfusion sanguine (CTS)

Le Centre de Transfusion sanguine (CTS) est chargé de la collecte, du traitement, de la préparation et de la distribution du sang dans les centres hospitaliers. Ils s'inscrivent dans la politique de développement de la transfusion sanguine dans notre pays [15].

1.11.1. La banque de sang

Une banque du sang est un établissement chargé de la collecte, la préparation, la qualification biologique, la gestion et la distribution du sang et des médicaments dérivés du sang [3].

1.11.2. La gestion des donneurs de sang et de plaquettes dans les CTS

Les CTS rencontrent souvent des problèmes lors de l'organisation des programmes de collecte de sang car dans les hôpitaux la gestion des donneurs de sang se fait sur papier, chaque donneur a un dossier qui contient ses informations. A chaque fois l'administrateur est obligé de les insérer manuellement dans la base de données à partir des fiches de donneurs en raison de l'inexistence d'un logiciel de gestion des donneurs. La probabilité d'erreurs de saisie est donc très élevée [23].

1.12. Conclusion

Dans ce chapitre nous avons présenté les concepts généraux de notre projet (le don de sang et ses composants). Dans le chapitre suivant nous aborderons la partie analyse des besoins et la conception de notre système.

Chapitre 2: Analyse des besoins et conception

2.1. Introduction

Après avoir présenté dans le premier chapitre le contexte général de notre projet, nous allons passer dans ce chapitre à l'étude conceptuelle et logique des données nécessaires pour la réalisation de notre système d'information. Un projet de gestion ne peut être réalisé d'une manière simple et facile sans l'utilisation d'une méthode d'analyse et de conception convenable. En effet, tout travail informatique doit être réalisé avec une méthodologie qui aide les réalisateurs à mieux conduire leurs projets. Plusieurs langages de modélisation existent, parmi lesquels le langage UML (Unified Modeling Language) que nous avons choisi. Nous avons aussi suivi le processus de développement de logiciel RUP (Rational Unified Process) pour réaliser notre projet.

2.2. Spécification des besoins

Des besoins « incomplets » sont une des principales raisons de l'échec des projets de développement. Pour cette épreuve il nous faut décrire et distinguer les besoins fonctionnels et non-fonctionnels du système étudié.

2.2.1. LES BESOINS FONCTIONNELS

Il s'agit des fonctionnalités du système, ce sont des besoins qui répondent aux points précis du cahier des charges, et sont donc requis par le client.

Les besoins fonctionnels recueillis sont les suivants :

1. La gestion des donneurs : Elle nécessite les tâches suivantes :
 - Ajouter certains donneurs ou en supprimer si c'est nécessaire.
 - Consulter les informations sur les donneurs
 - Vérifier leurs disponibilités

2. La gestion des collectes de sang : Elle nécessite les tâches suivantes :
 - La gestion des bénévoles et des donneurs
 - La localisation des CTS, lieux de collectes, partenaires et associations
 - Le contact avec les partenaires, les associations, les donneurs, les CTS
3. Le partage d'information (Les informations peuvent être partagées entre CTS et Associations)
4. La gestion de ses propres collaborateurs, c'est-à-dire, créer ou supprimer d'autres administrateurs pour le logiciel en cas de besoin

2.2.2. LES BESOINS NON FONCTIONNELS

Il s'agit des besoins qui caractérisent le système. Ce sont des contraintes, exigences ou choix techniques. Dans le cadre de ce projet l'application doit assurer :

- L'authentification
- La gestion des champs
- L'utilisation de la base de données pour stocker les données de l'application.
- La gestion des erreurs du système sans perte d'informations
- Le cryptage et hachage des données pour protéger l'ensemble des données dans la BDD ou durant le transfert des fichiers
- L'utilisation d'une architecture client/serveur.

2.3. Méthodologie et approche

2.3.1. Le langage UML

UML (Unified Modeling Language) est basé sur l'approche par objets. Celle-ci vit le jour bien avant UML dans le domaine des langages de programmation Simula, le tout premier langage à objets est né dans les années 1960. Ce langage a connu beaucoup de successeurs comme Small talk, C++, Java ou encore récemment C#. Autrement dit, UML est un Langage de modélisation orientée objet dans le but est de définir la notation standard pour la modélisation des applications construites à l'aide d'objets [13].

2.3.2. Démarche suivie

Puisque le langage UML ne préconise aucune démarche, il y a un avantage à choisir une méthode adaptée au système à réaliser. Dans notre cas, nous avons choisi le processus de développement RUP qui est une méthode qui prend en charge le développement des applications orientés objets durant la partie modélisation du cycle de vie du logiciel [22].

2.3.3. L’outil StarUML

Nous avons utilisé comme outils de modélisation UML, l’outil StarUML qui est un logiciel de modélisation, qui aide à tracer les différents diagrammes UML. Cet outil donne la possibilité d’exporter les diagrammes réalisés en images de format « png » ou « jpeg » afin de les insérer au sein du document. La Figure 1 montre l’interface principale de l’outil de modélisation StarUML version 2.8.0.

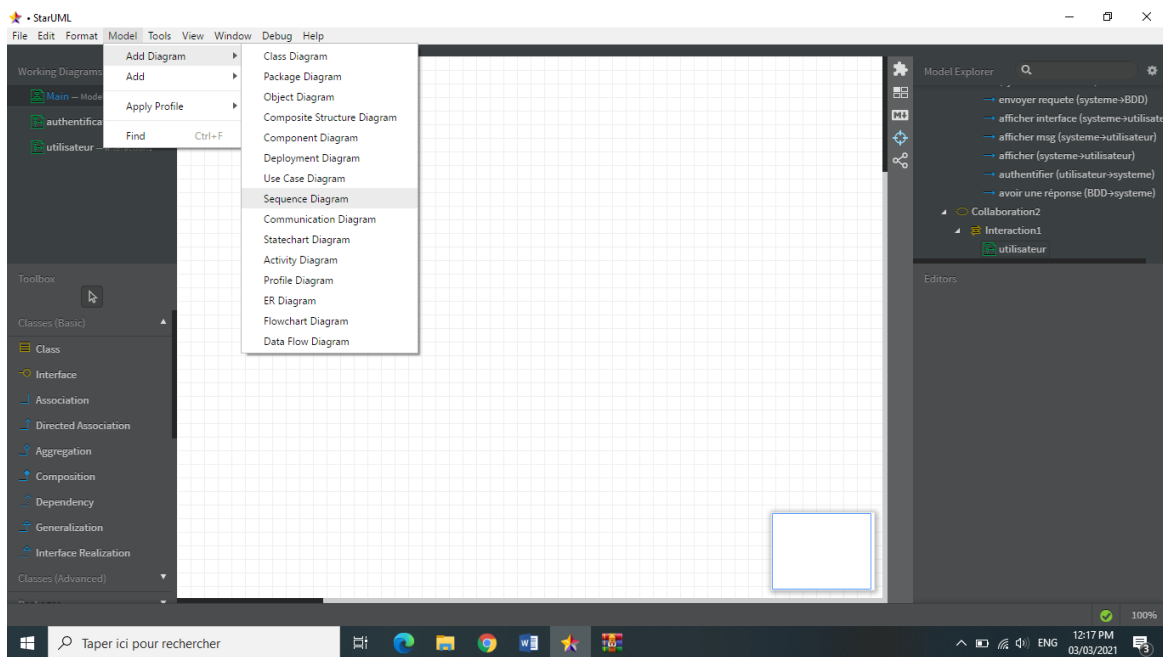


Figure 1 : L’outil STAR UML

2.3.4. La méthode RUP

Rational Unified Process (RUP) est à l’origine du processus unifié. Il est à ce titre l’implémentation la plus connue. Il s’agit d’un produit et d’une marque de Rational Software,

entre-temps acquise par IBM. La méthode est livrée clés en main et est accompagnée d'outils pour guider les équipes dans l'adaptation et exécution du processus

2.4. Analyse des besoins

2.4.1. Les diagrammes de paquetages

Les diagrammes de paquetages sont la représentation graphique des relations existant entre les paquetages composant un système dans le langage Unified Modeling Language.

Nous avons découpé notre application en paquetages selon le point de vue fonctionnel. Chaque paquetage représente un sous-système de l'application

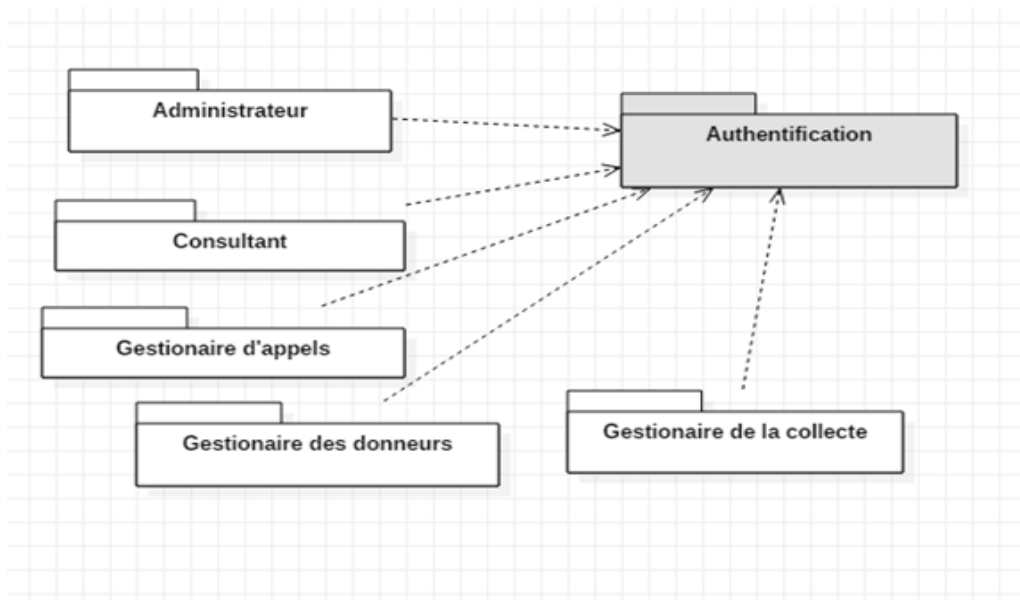


Figure 2 : Le diagramme de paquetages

2.4.2. Les diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour une représentation du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés

- **Le diagramme de cas d'utilisation pour le paquetage « Administrateur »**

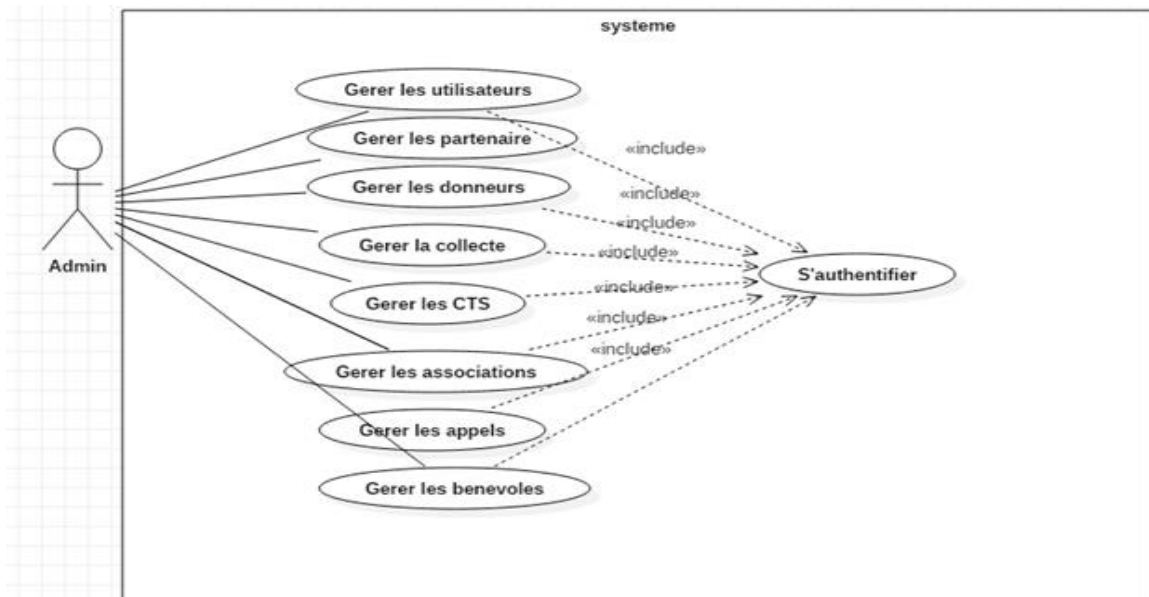


Figure 3 : Diagramme de cas d'utilisation pour le package « Administrateur »

L'administrateur a un accès complet, il peut attribuer les droits d'utilisation, il gère les partenaires, les donneurs, les cts, les collectes, les appels, et les bénévoles (ajout, suppression ou bien mettre à jour)

- **Le diagramme de cas d'utilisation pour le paquetage « Consultation »**

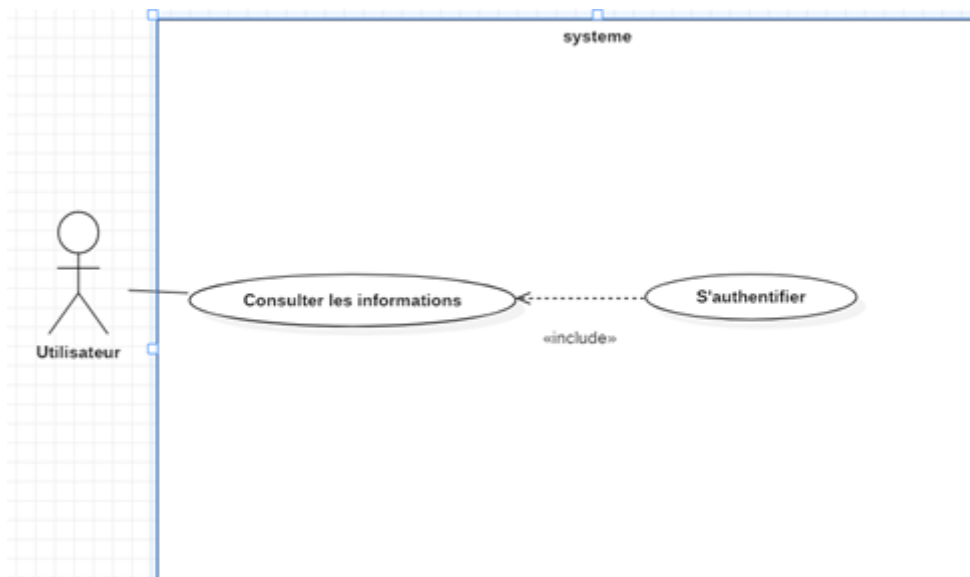


Figure 4 : Diagramme de cas d'utilisation pour le paquetage « Consultation »

Le consultant peut voir toutes les tables sans avoir la capacité de modifier ou de supprimer une information. Son rôle se limite à la consultation des informations. Par exemple, il peut savoir combien la base de données contient de donneurs ou quels sont les partenaires des CTS.

- **Le diagramme de cas d'utilisation pour le paquetage « Gestionnaire des donneurs »**

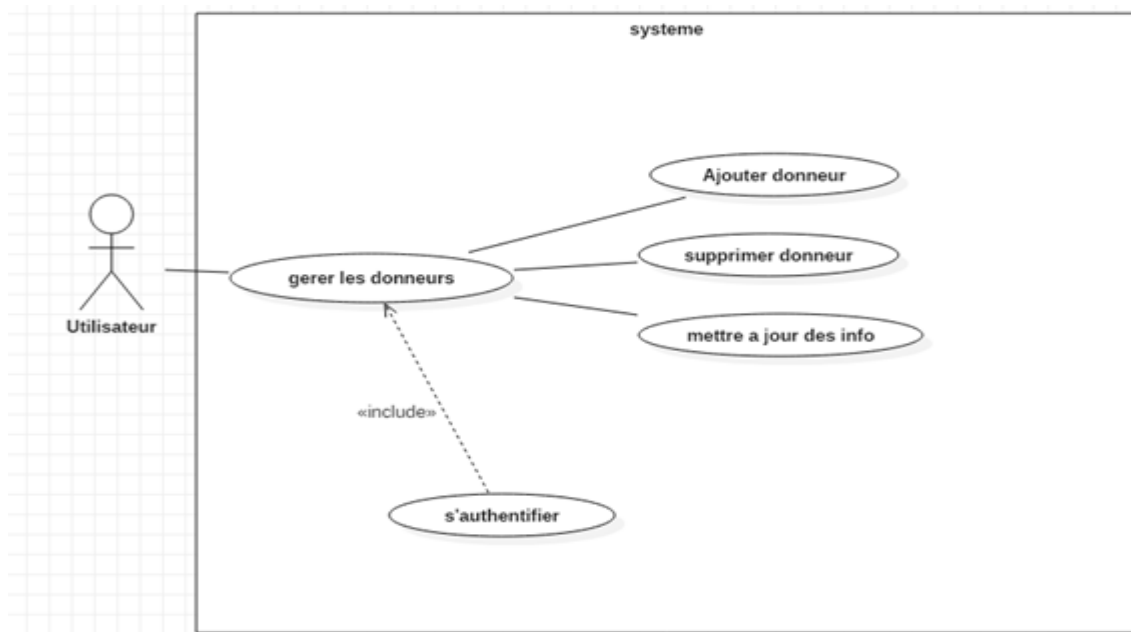


Figure 5 : Diagramme de cas d'utilisation pour le paquetage « Gestionnaire des donneurs »

Cet utilisateur est responsable des donneurs, il peut donc ajouter un nouveau donneur en remplissant toutes les informations sur lui, il peut aussi modifier ou supprimer un donneur déjà existant dans la base de données.

- **Le diagramme de cas d'utilisation pour le paquetage « Gestionnaire d'appels »**

Son rôle est d'appeler les bénévoles et les donneurs avant la préparation de la collecte. Il remplit les tables **Appels bénévoles** et **Appels donneurs**, comme il peut aussi les modifier ou les supprimer.

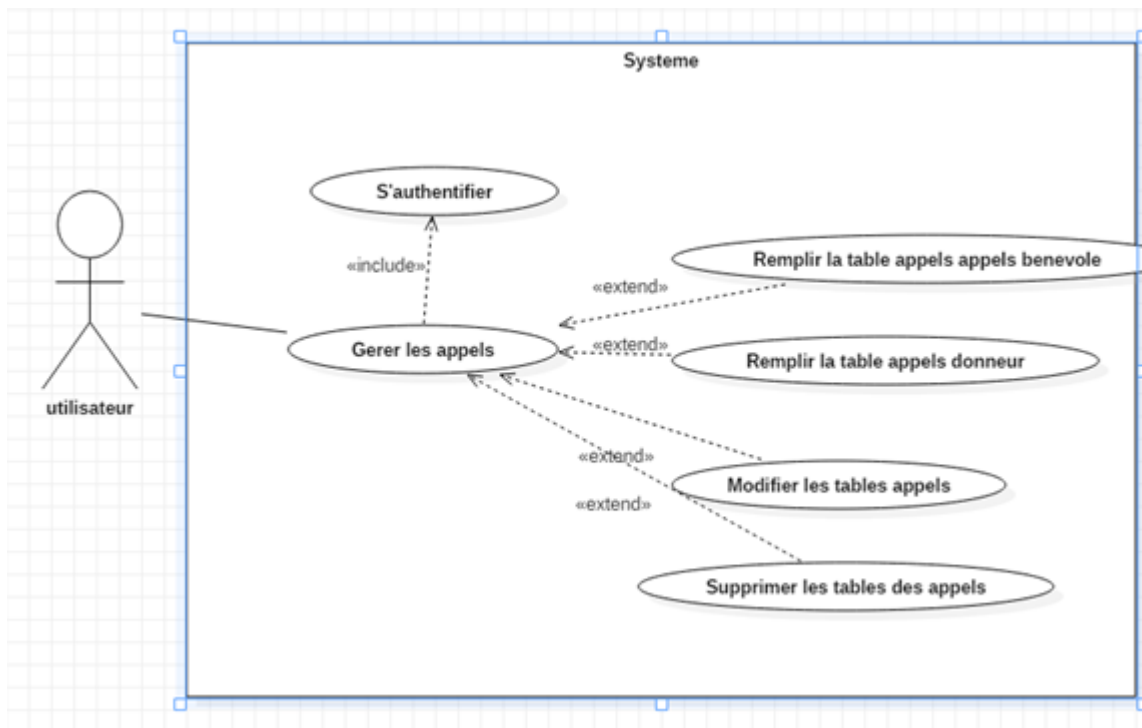


Figure 6 : Diagramme de cas d'utilisation pour le paquetage « Gestionnaire d'appels »

- **Le diagramme de cas d'utilisation pour le paquetage « Gestionnaire de la collecte »**

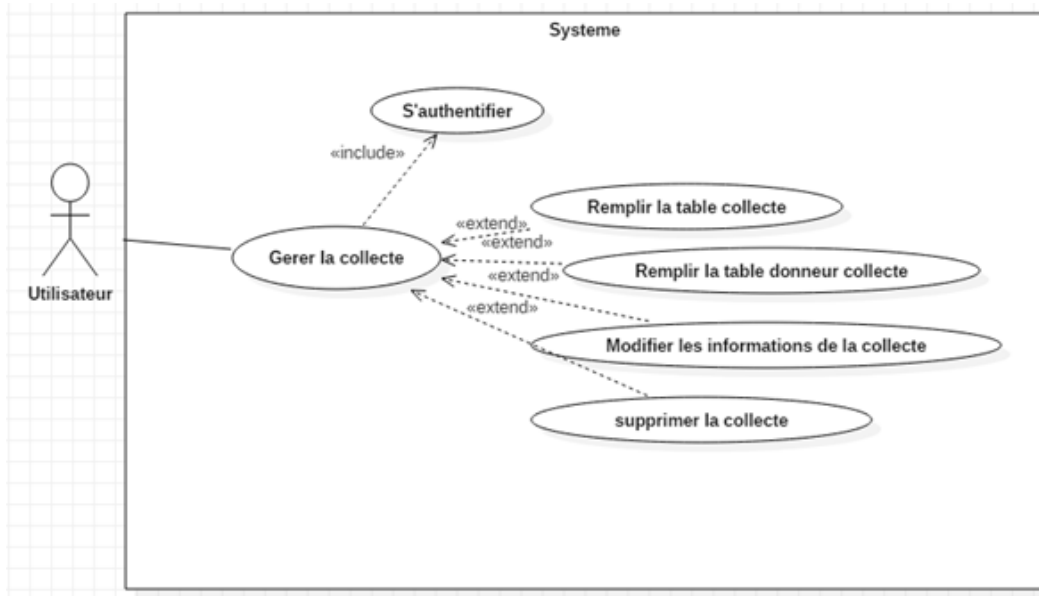


Figure 7 : Diagramme de cas d'utilisation pour le paquetage « Gestionnaire de la collecte »

Son rôle est de gérer la table **Collecte** et également la table **Donneurs_collecte** comme il peut les supprimer ou les mettre à jour.

2.4.3. Diagrammes de séquences

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML [03].

- Description textuelle du cas « S’authentifier » :

Identification :

Nom du cas : « S’authentifier »
But : Décrire les étapes permettant à l’utilisateur de l’application de s’authentifier via l’application.
Acteur principal : Utilisateur (Gérer les donneurs uniquement, Consultation uniquement, Appels uniquement, Collecte Uniquement, Administration)
Acteur secondaire : Néant.
Date de création : 07/05/2021
Date de mise à jour : 25/05/2021
Responsable : Sarra Latreche & Nasr Allah Ould Mamar

Séquencement :

Ce cas commence lorsque l’utilisateur veut accéder à l’application.

Préconditions :

L’utilisateur possède un nom d’utilisateur, un mot de passe et la présence de l’application.

Enchaînement nominal :

- 1) l’utilisateur saisit son nom d’utilisateur et son mot de passe.
- 2) L’utilisateur valide les données.

3) Le système affiche la page correspondante à l'utilisateur.

Enchaînement d'exceptions :

E1 : Le champ du nom d'utilisateur ou du mot de passe est vide

2) L'enchaînement démarre après le point 2 de la séquence nominale.

3) Le système indique que l'un des champs est vide.

4) Retour au point 1 de l'enchaînement nominal.

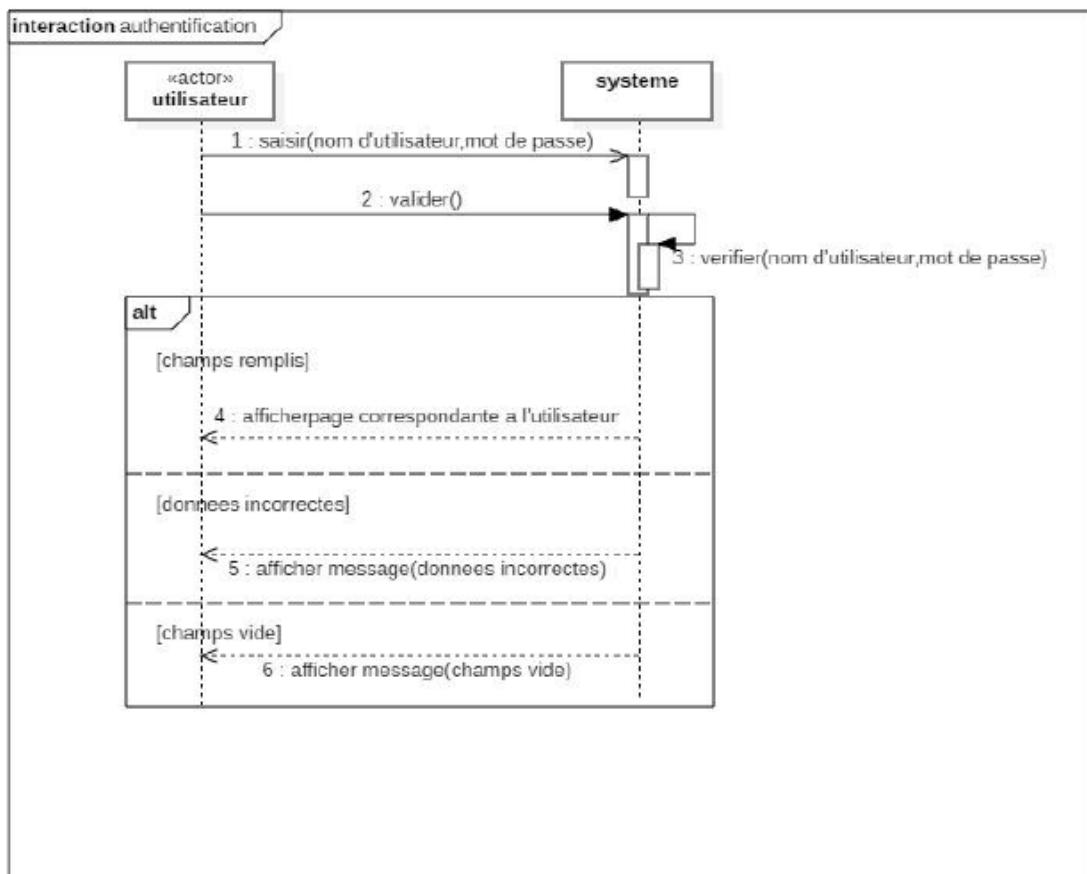


Figure 8 : Diagramme de séquence du cas « S'authentifier »

E2 : Nom d'utilisateur ou mot de passe incorrect

2) L'enchaînement démarre après le point 2 de la séquence nominale.

3) Le système indique que les données sont incorrectes.

4) Retour au point 1 de l'enchaînement nominal.

Post conditions :

Utilisateur authentifié.

- **Description textuelle du cas « Gérer donneurs uniquement » :**

Identification :

Nom du cas : « Gérer les donneurs uniquement »
But : décrire les étapes permettant à l'utilisateur de gérer les donneurs
Acteur principal : Admin
Acteur secondaire : Gestionnaire des donneurs, Collecte Uniquement
Date de création : 07/05/2021
Date de mise à jour : 02/06/2021
Responsable : Sarra Latreche & Nasr Allah Ould Mamar

Séquencement :

Ce cas commence lorsque l'utilisateur veut gérer les donneurs.

Préconditions :

L'utilisateur est déjà connecté

Il possède le droit d'accéder à la page Donneurs

Enchaînement nominal :

1) Après l'authentification, il accède à la page Donneurs

2) Avoir la liste des donneurs, il peut donc :

- Ajouter un nouveau donneur en saisissant les données du donneur comme le nom, prénom, date de naissance...
- Modifier un donneur en choisissant le donneur et modifier un ou plusieurs champ de ses informations.
- Supprimer un donneur.

3) Valider

4) Le système affiche un message qui prouve que l'opération (ajouter/modifier/supprimer) est bien faite.

Enchaînement d'exceptions :

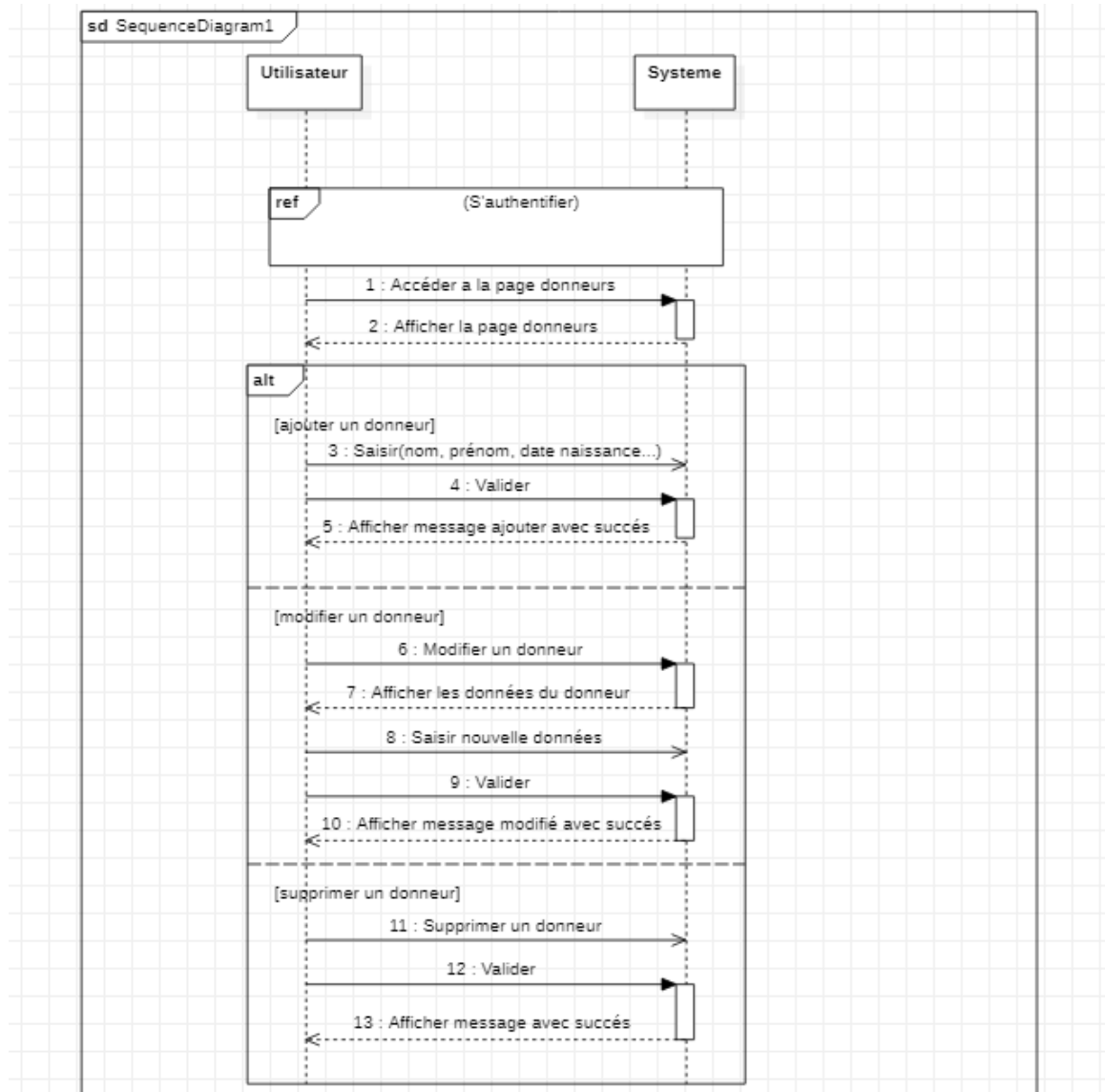


Figure 9 : Diagramme de séquence du cas « Gérer donneurs uniquement »

E1 : Un champ obligatoire est vide

2) L'enchaînement démarre après le point 2 de la séquence nominale.

- 3) Le système indique que l'un des champs est vide.
- 4) Rester dans la même page si le champ est toujours vide
- 5) Retour a le point 1 de la séquence nominale

E2 : Champ incorrecte

- 2) L'enchaînement démarre après le point 2 de la séquence nominale.
- 3) Le système indique que les données sont incorrectes (poids inférieur à 50 kg/ Nombre de dons négatif...)
- 4) Retour au point 1 de l'enchaînement nominal.

Post conditions :

Opération faite avec succès

Nom du cas : « Ajouter une collecte/ Donneurs_collecte»
But : Décrire les étapes permettant à l'utilisateur d'ajouter une nouvelle collecte ou les donneurs assistaient à une collecte.
Acteur principal : Admin
Acteur secondaire : Collecte Uniquement
Date de création : 20/05/2021
Date de mise à jour : 08/06/2021
Responsable : Sarra Latreche & Nasr Allah Ould Mamar

Séquencement :

Ce cas commence lorsque l'utilisateur veut ajouter une nouvelle collecte ou un donneur_collecte.

Préconditions :

L'utilisateur est déjà connecté

Il possède le droit d'ajouter une collecte ou un Donneur_collecte.

Enchaînement nominal :

- 1) Après l'authentification, il accède à la page ajouter une collecte/ajouter un donneur_collecte
- 2) Saisir les informations qui concerne la collecte ou donneur_collecte
- 3) Valider
- 4) Le système affiche un message « ajouté avec succès ».

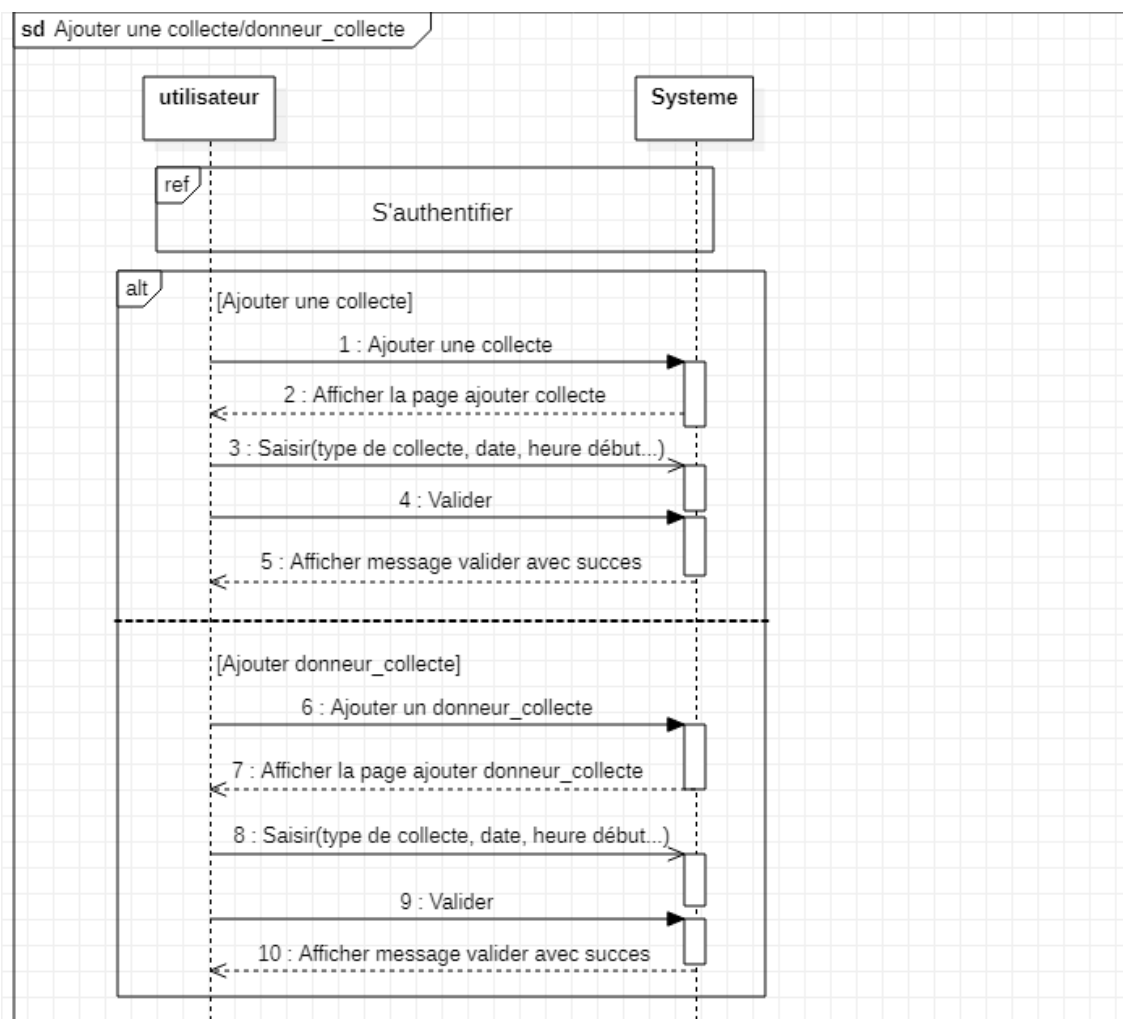


Figure10 : Diagramme de séquence du cas : « Ajouter une collecte/Donneurs_collecte »

Enchaînement d'exceptions :

E1 : Un champ obligatoire est vide

- 2) L'enchaînement démarre après le point 2 de la séquence nominale.
- 3) Le système indique que l'un des champs est vide.
- 4) Rester dans la même page si le champ est toujours vide
- 5) Retour a le point 1 de la séquence nominale

E2 : Champ incorrecte

- 2) L'enchaînement démarre après le point 2 de la séquence nominale.
- 3) Le système indique que les données sont incorrectes (nombre de dons négatif/Nom ou prénom qui contient des chiffres...)
- 4) Retour au point 1 de l'enchaînement nominal.

Post conditions :

Opération faite avec succès

2.5. La conception

2.5.1. Diagramme de classe

Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application [03].

Nous avons utilisé le diagramme de classes pour son importance dans la modélisation orientée objet. Alors que le diagramme de cas d'utilisation montre un système du point de vue de l'interaction entre les utilisateurs et le système, le diagramme de classes en montre la structure interne du système [10]. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il s'agit d'une vue statique, car on ne tient pas compte du facteur de temps dans le comportement du système. La Figure 11 montre le diagramme de classes de notre système.

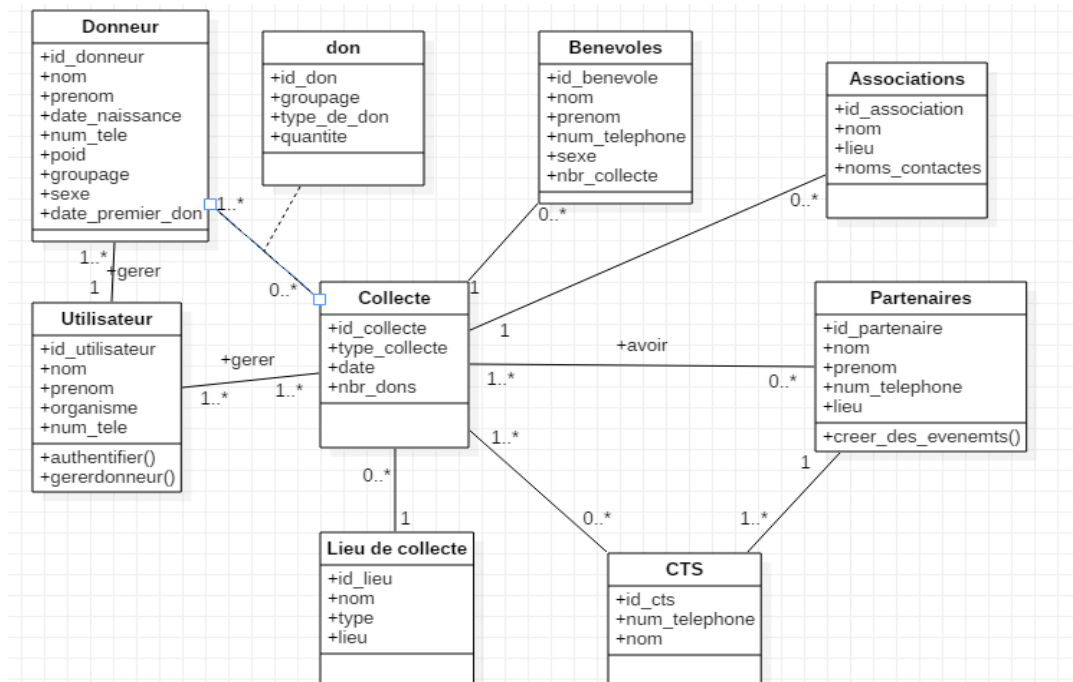


Figure 11 : Diagramme de classe

2.5.2. Description textuelle

Ce diagramme représente les différentes classes de notre système de gestion et les relations entre les classes tel que :

La classe utilisateur : représente l'utilisateur de notre logiciel un utilisateur peut gérer un ou plusieurs donneurs comme il peut gérer aussi un ou plusieurs collectes de sang.

La classe don : Un don peut être effectué par un ou plusieurs donneurs.

La classe donneur représente les donneurs.

La classe collecte : la collecte est effectuée avec les CTS, les associations et les partenaires, et chaque collecte doit avoir un lieu de collecte de sang.

La classe CTS: le cts peut être dans 0 ou plusieurs collecte comme il peut avoir au moins 1 partenaire.

La classe bénévoles : elle représente les bénévoles qui participent a une collecte. Le bénévole peut participer à une seule collecte au même moment.

la classe Associations: c'est la classe des associations qui collabent avec le cts, ses.

la classe partenaires: Un partenaire peut participer à 0 ou plusieurs collectes mais il est partenaire à 1 seul CTS.

la classe lieu de collecte: elle représente le lieu ou la collecte se déroule. La collecte elle a un seul lieu mais le même lieu peut être associé à plusieurs collecte .

2.6. Contenu de la BD

La Figure 12 représente les différentes tables de la base de données utilisées pour le traitement de notre sujet tel que :

- **La table Utilisateurs :** contient l'identifiant de l'Utilisateur comme une clé primaire, Nom, Prénom, Téléphone, Organisme, Type d'organisme (CTS, Association) Fonction, Type d'Accès qui peut être (Gérer le donneurs, Consultation uniquement, Appels uniquement, Collecte Uniquement, Administration), Utilisateur (Login), Mot de Passe et ID Association, ID CTS comme des clés étrangères

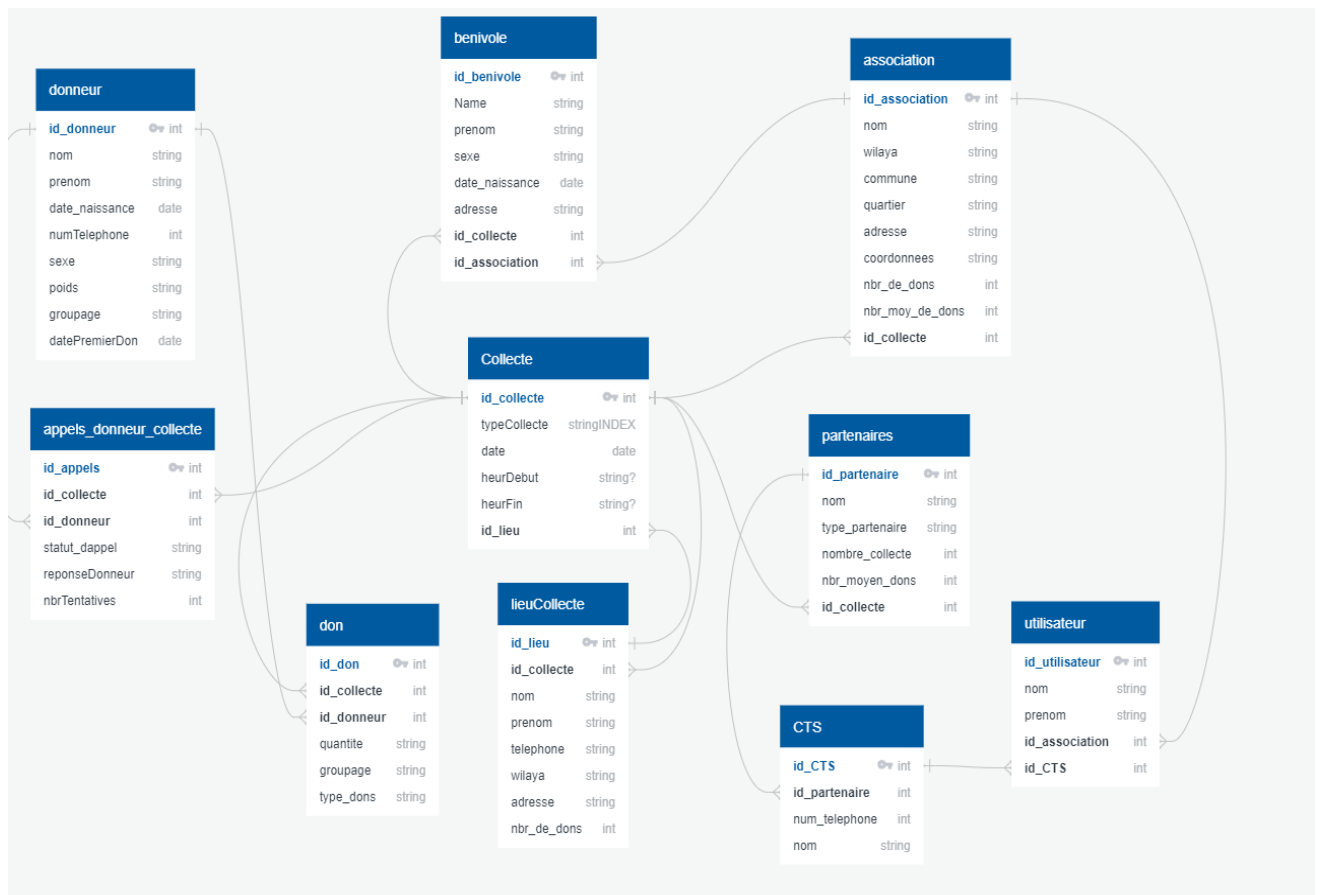


Figure 12 : Les tables de la base de données

- **La table donneur** contient les informations des donneurs (l'identifiant du donneur comme une clé primaire, nom, prénoms, date de naissance, numéro de téléphone, groupage, poids, et date de premier don)
- **La table collecte** contient (l'identifiant de collecte comme une clé primaire, type collecte, la date, l'heure et id lieu comme clé étrangère)
- **La table Bénévole** contient (l'identifiant d'un bénévole comme une clé primaire, son nom, prénom, adresse, sexe, id collecte comme et id association clés étrangères).
- **La table Dons** : elle contient l'ID don comme clé primaire et ID Collecte, ID Donneur comme clés étrangères aussi les Types de don (Sang complet, Don de plaquettes), la Quantité, et le Groupage
- **La table Partenaires** contient l'id Partenaire comme clé primaire , Nom long, Nom court, Type Partenaire, les Noms et les Prénoms des contacts, Wilaya, Commune,

Quartier, Adresse, Coordonnées, Nombre de collecte, Nombre de dons et le Nombre moyen de dons

- **La table Associations** contient l'ID Association comme clé primaire, , Nom long, Nom court, Type Partenaire, les Noms et les Prénoms des contacts, Wilaya, Commune, Quartier, Adresse, Coordonnées, Nombre de collecte, Nombre de dons et le Nombre moyen de dons.
- **La table Lieu de collectes** contient l'Id Lieu comme clé primaire le Nom long, Nom Court, Type (CTS, Entreprise, Mosquée, Place publique, Université, Administration, Siège d'association), la Wilaya, la Commune, le Quartier, les Coordonnées, l'Effectif et le Nombre moyen de dons.

2.7. Conclusion

Dans ce chapitre nous avons représenté les besoins fonctionnels et non fonctionnels de notre système. Nous avons ensuite présenté le langage UML, la méthode de développement RUP, l'outil de modélisation (StarUML), ainsi que les diagrammes qui serviront à réaliser le logiciel de gestion des donateurs d'une banque de sang.

Chapitre 3: Implémentation et expérimentation

3.1. Introduction

Dans ce chapitre nous entamons la présentation de l'environnement de travail, en détaillant les différents outils utilisés. Nous allons aussi présenter les différentes interfaces principales du système.

3.2. Configuration requise :

Certaines conditions doivent être remplies pour que le don de sang atteigne ses objectifs. Les exigences sont classées en exigences côté utilisateur et côté serveur.

- **Exigences côté serveur :**

Le serveur doit être un serveur SQL prenant en charge PHP.

Au niveau du serveur, il doit exister une base de données relationnelle comprenant les différentes tables utilisées.

La base de données doit pouvoir communiquer avec le logiciel côté utilisateur.

- **Exigence côté utilisateur :**

Avoir un PC qui supporte toutes les installations nécessaires.

3.3. Architecture du système :

L'architecture du système est le modèle qui définit la structure et le comportement d'un système. Notre Application de gestion d'une banque de sang se compose d'un logiciel côté utilisateur et de PHP-MySQL côté serveur.

Le logiciel est la partie visible pour l'utilisateur pour l'interaction tandis que le composant côté serveur basé sur PHP/MySQL sert d'intermédiaire entre le logiciel et la base de données sur le serveur.

Il existe 5 types de rôles d'utilisateurs au sein du logiciel :

- **Administration** : Ce rôle permet de gérer tout dans le logiciel, il peut ajouter, modifier ou supprimer n'importe quel utilisateur, il peut aussi gérer les donateurs, les CTS, les associations, les bénévoles, les collectes, les lieux, les partenaires, les appels...
- **Consultation uniquement** : Ce rôle permet uniquement de consulter toutes les tables sans faire des modifications sur les données.
- **Appels uniquement** : Ce rôle est responsable de la gestion des appels des donateurs et des bénévoles.
- **Collecte uniquement** : Ce rôle gère les collectes, il peut ajouter, modifier ou supprimer dans la table **Donneurs**, **Collecte**, **Donneurs_collecte**.
- **Gérer les donateurs uniquement** : Ce rôle permet uniquement d'ajouter, modifier ou supprimer un donneur.

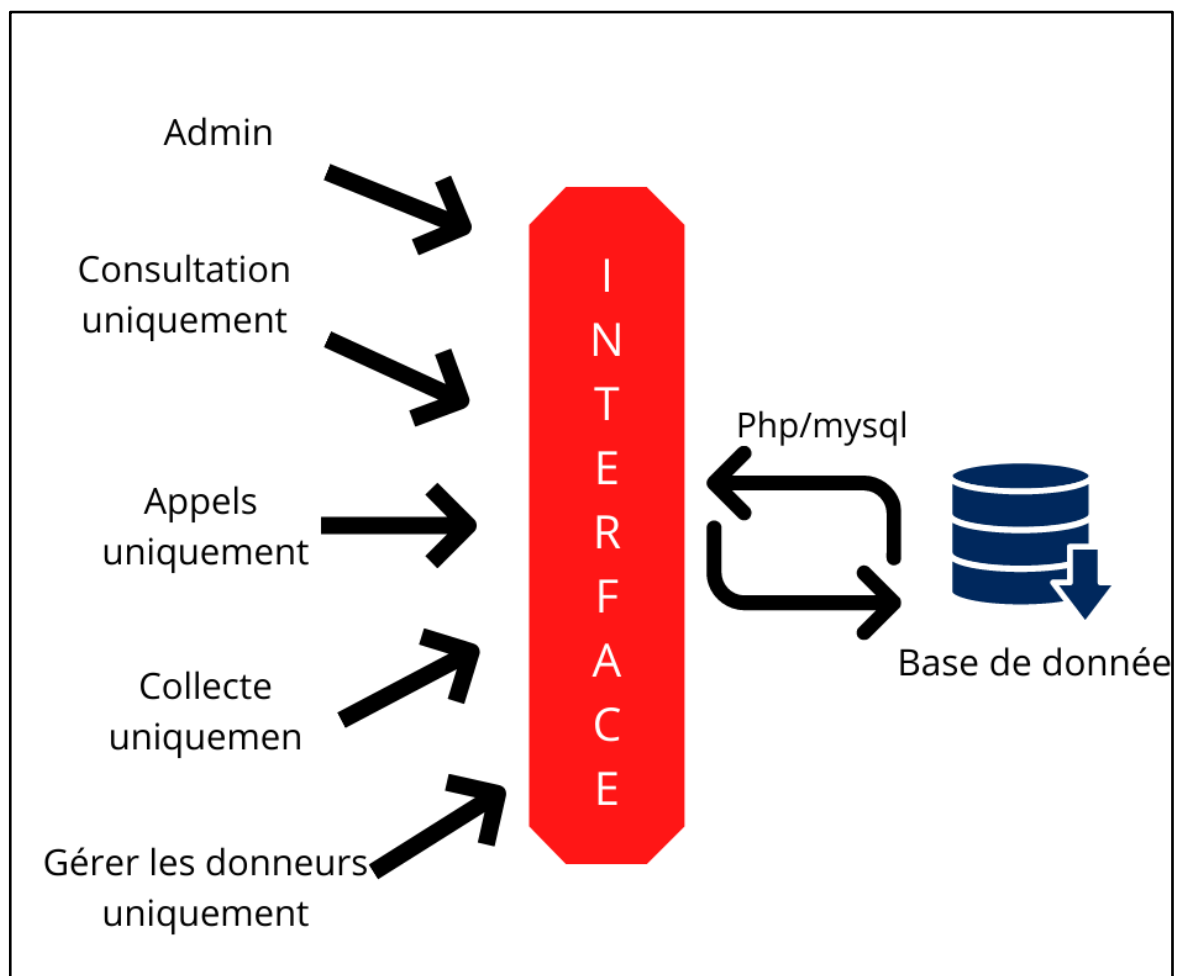


Figure 13 : l'architecture du système

3.4. Outils de développement

Nous avons utilisé un certain nombre d'outils, que nous présentons dans ce qui suit :

- **Xampp**

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide [3].

- **Php**

Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet [21].

- **Php Myadmin**

PhpMyAdmin est un logiciel gratuit écrit en PHP, destiné à gérer l'administration de MySQL sur le Web. Il prend en charge un large éventail d'opérations sur MySQL et MariaDB. Les opérations fréquemment utilisées (gestion des bases de données, des tables, des colonnes, des relations, des index, des utilisateurs, des autorisations, etc.) [14].

- **MySql**

MySQL est un système de gestion de bases de données relationnelles. Il est distribué sous une double licence GPL et propriétaire [21].

- **Sql:**

SQL est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher,

d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles [21].

- **Node js**

Node.js est une plateforme logicielle libre en JavaScript, orientée vers les applications réseau événementielles hautement concurrentes qui doivent pouvoir monter en charge. Elle utilise la machine virtuelle V8, la librairie libuv pour sa boucle d'évènements, et implémente sous licence MIT les spécifications CommonJS [22].

- **Electron js**

Electron.js est un framework d'exécution qui permet à l'utilisateur de créer des applications de suite bureautique avec HTML5, CSS et JavaScript. C'est un projet open source lancé par Cheng Zhao, ingénieur chez GitHub. [16]

- **HTML**

HTML signifie « *HyperText Markup Language* » qu'on peut traduire par « langage de balises pour l'hypertexte ». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure. D'autres technologies sont utilisées avec HTML pour décrire la présentation d'une page (CSS) et/ou ses fonctionnalités interactives. [17]

- **CSS**

Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium [18]

- **JavaScript**

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web [20].

- **Bootstrap :**

Bootstrap est une collection d'outils. Il contient toutes sortes de modèles de conception basés sur HTML et CSS pour diverses fonctions et composants tels que la navigation, le système de grille, les carrousels d'images et les boutons.[24]

- **Adobe Photoshop**

Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur édité par Adobe. Il est principalement utilisé pour le traitement de photographies numériques, mais sert également à la création d'images. [25]

3.5. Présentation des interfaces

Notre logiciel contient plusieurs pages, nous présentons dans ce qui suit les interfaces des différentes pages.

3.5.1. La page de Login

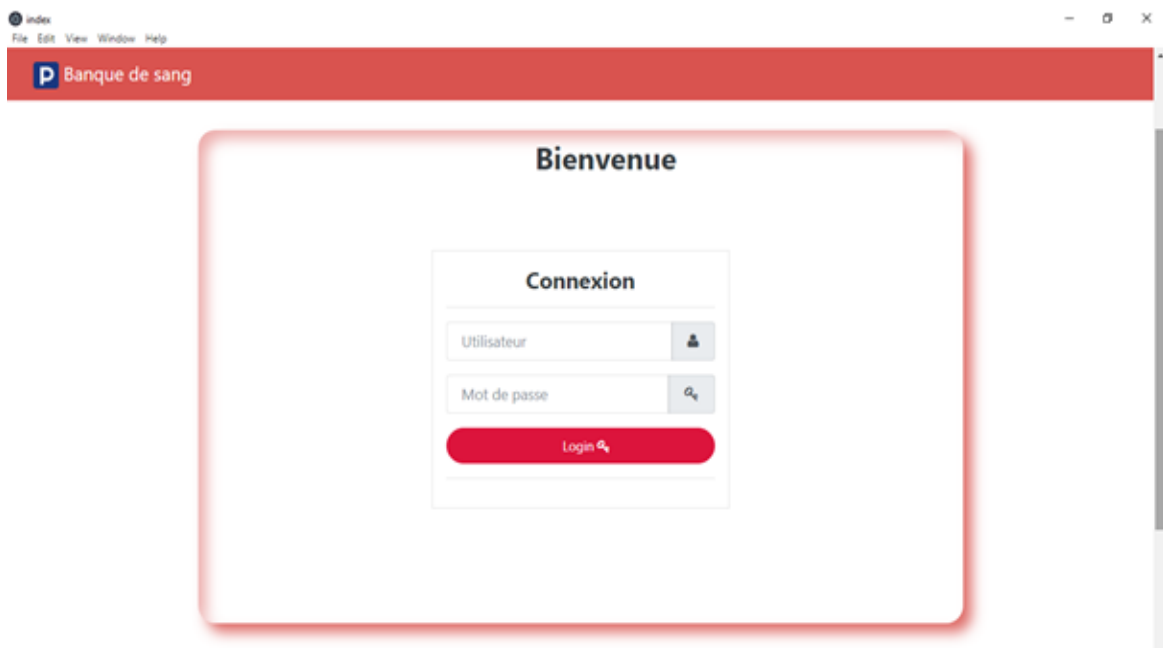


Figure 14 : La page login

Après le lancement du logiciel, cette page apparaît afin de permettre à l'utilisateur de s'identifier. Celui-ci doit entrer son nom d'utilisateur et son mot de passe pour continuer sinon il ne peut pas accéder au menu d'accueil du logiciel. Seul l'administrateur peut inscrire des nouveaux utilisateurs et leur donner leurs noms d'utilisateur et leurs mots de passe.

3.5.2. Page d'accueil (admin)

Après la vérification du nom d'utilisateur et du mot de passe intronisé, l'utilisateur est dirigé vers sa page d'accueil. Ses droits d'accès lui permettent d'accéder à d'autres pages.

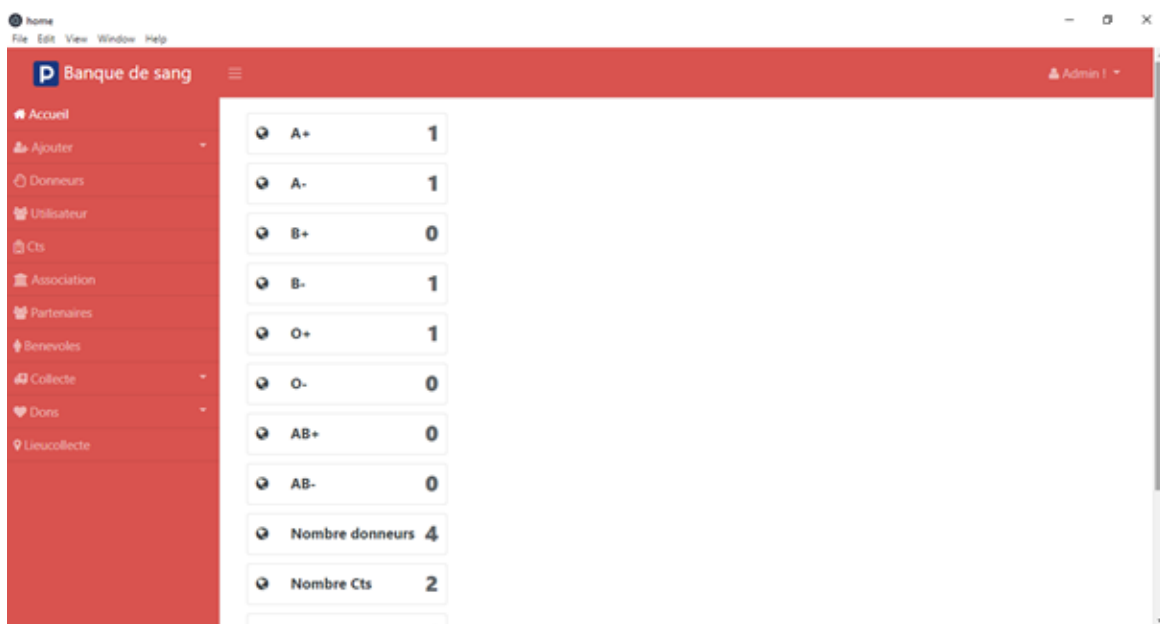


Figure 15 : La page accueil

Dans cet exemple, nous allons prendre la page d'admin qui donne accès à toutes les pages du logiciel.

La page d'accueil est présentée d'une manière simple :

A gauche il y a le panneau de navigation qui permet à l'administrateur de se déplacer d'une page à l'autre (Nous détaillerons par la suite chaque page). Au milieu, il peut voir le stock de sang de chaque groupe sanguin disponible, pour pouvoir en cas d'urgence intervenir au plus vite. L'utilisateur n'est pas obligé d'aller dans la table de donneur et chercher par groupage de sang. Il peut directement avoir toutes ces informations dans la page d'accueil comme il peut avoir aussi le nombre de donneurs, le nombre de CTS et le nombre d'associations.

3.5.3. Page Ajout

Dans cette page l'administrateur peut ajouter un nouvel utilisateur, un donneur, un CTS, une association, un partenaire, un bénévole ou un lieu en remplissant leurs coordonnées et valider l'ajout.

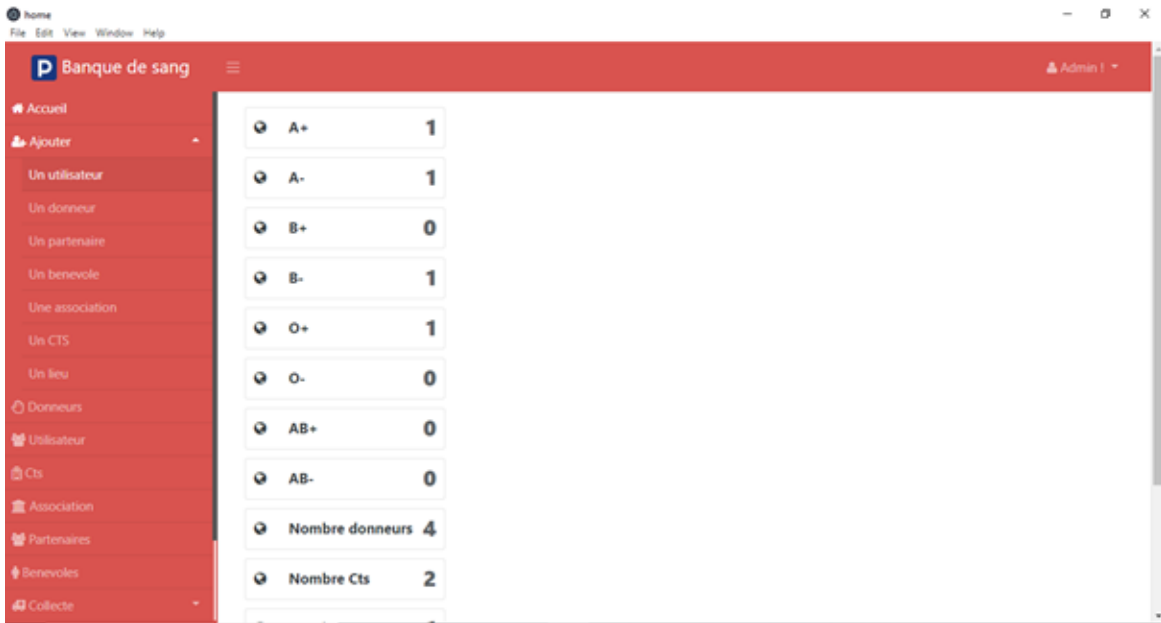


Figure 16 : La page ajout

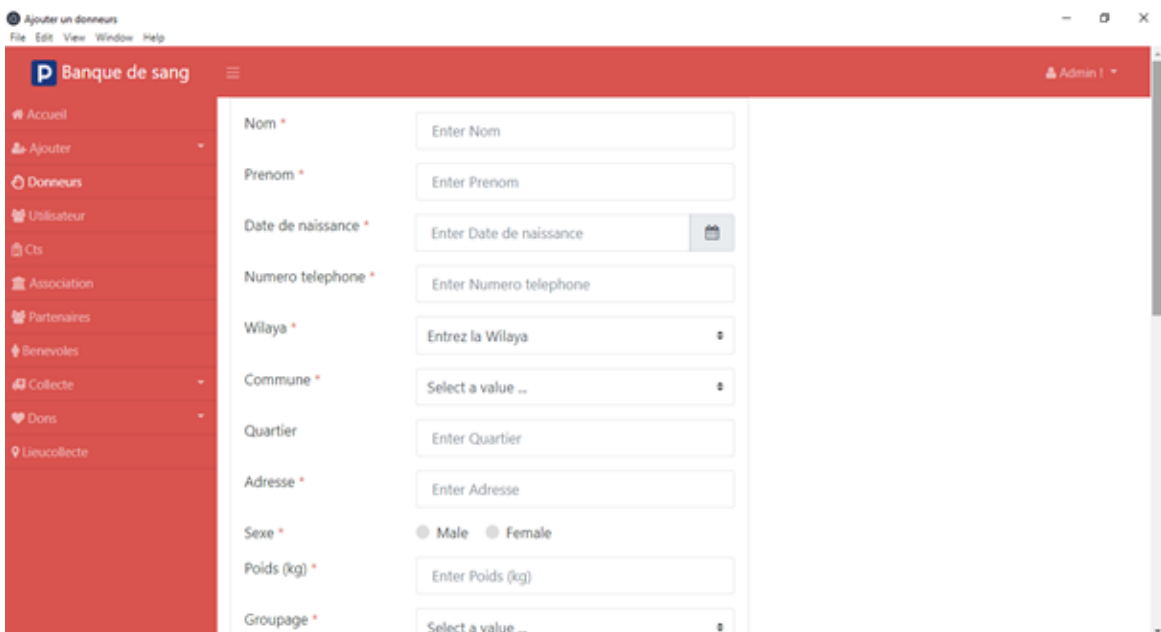


Figure 17 : La page ajouter un donneur

Les champs sont contrôlés. Par exemple le logiciel ne peut pas accepter un nom ou un prénom contenant des chiffres, la date est de type date, la commune est de type liste qui contient toutes les communes de Mostaganem. Le poids est entre 50 et 150 kg, le groupage de sang est une liste qui contient tous les groupes...

3.5.4. Page de la liste des donneurs

Elle contient toutes les informations sur les donneurs où l'administrateur peut ajouter, mettre à jour ou supprimer un donneur. Il peut aussi exporter ou bien importer les tables sous plusieurs formes (Excel, csv, Word...).

The screenshot shows the 'Banque de sang' application interface. The main content area is titled 'Les donneurs' and contains a table of donor records. A context menu is open over the first row, showing options: PRINT, PDF, WORD, CSV, and EXCEL. The table has the following data:

#	Nom	Prenom	Date de naissance	Numero de telephone	Wilaya	Adresse	Sexe	Poids (kg)	Groupage	Date de dernier don	Type dernier don	Nombre de don	Donneur de palaquette
1	Mohamed	Ilyes	0000-00-00	748364712	Mostaganem	aaaa	Male	78	B-	0000-00-00	Sang complet	10	0
			0000-00-00	755263142	Mostaganem	A&E mostaganem	Female	60	A-	0000-00-00	Don de plaquettes	6	0
			0000-00-00	782908970	Mostaganem	aaaa	Male	82	O+	0000-00-00	Sang complet	5	0
			0000-00-00	555698741	mostaganem	aaa	Male	60	A+	0000-00-00	Sang complet	3	0

At the bottom of the table, there are buttons for 'Export' and 'Import Data', and a status indicator 'Records : 4 of 4'.

Figure 18 : La liste des donneurs

3.5.5. Page collecte

Dans cette page l'admin peut gérer la collecte et les appels des bénévoles et des donneurs.

- **Ajouter une collecte**

L'administrateur peut ajouter une nouvelle collecte en remplissant le type de collecte, la date l'heure du début et de la fin de la collecte. Il doit choisir le lieu de la collecte via la liste des lieux qui est dans la base de données. Il choisit aussi, le CTS et les associations qui participent à la collecte.

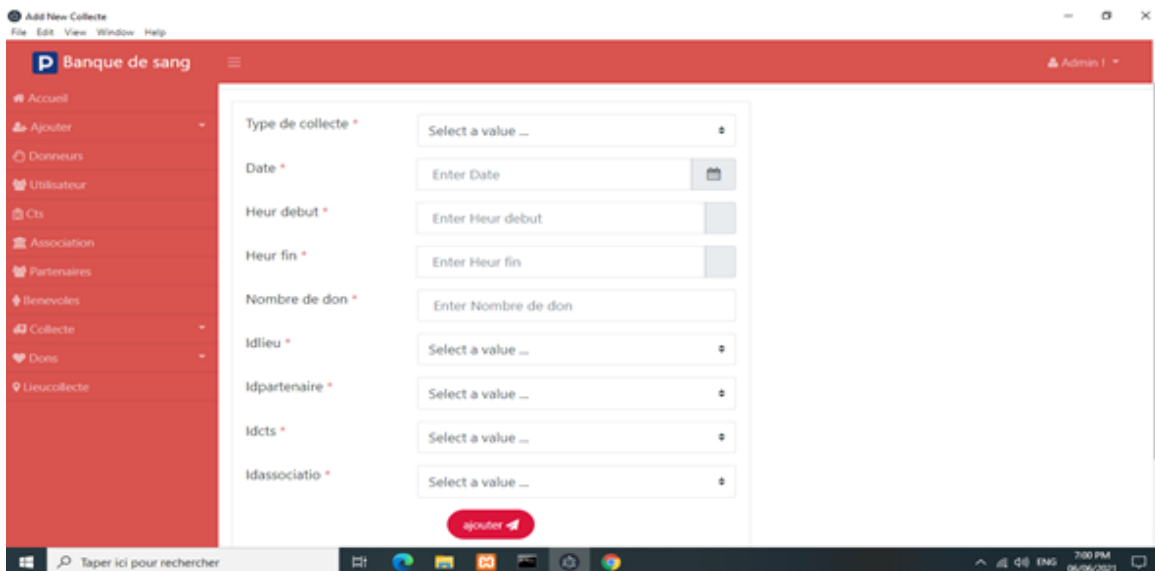


Figure 19 : La page ajouter une collecte

3.5.6. Ajouter donneur collecte

L'administrateur peut ajouter les donneurs qui ont assistés à la collecte en sélectionnant l'identifiant de la collecte et le numéro de la carte du donneur. Il peut aussi ajouter une remarque (Lors du questionnaire de don de sang ou lors des appels téléphoniques)

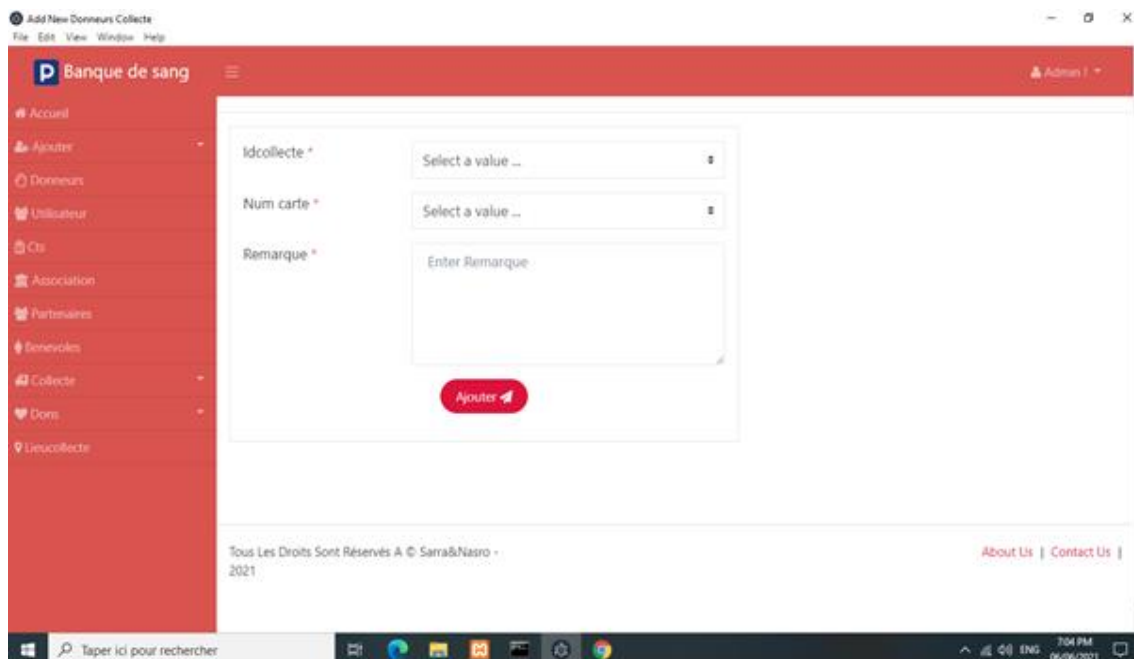
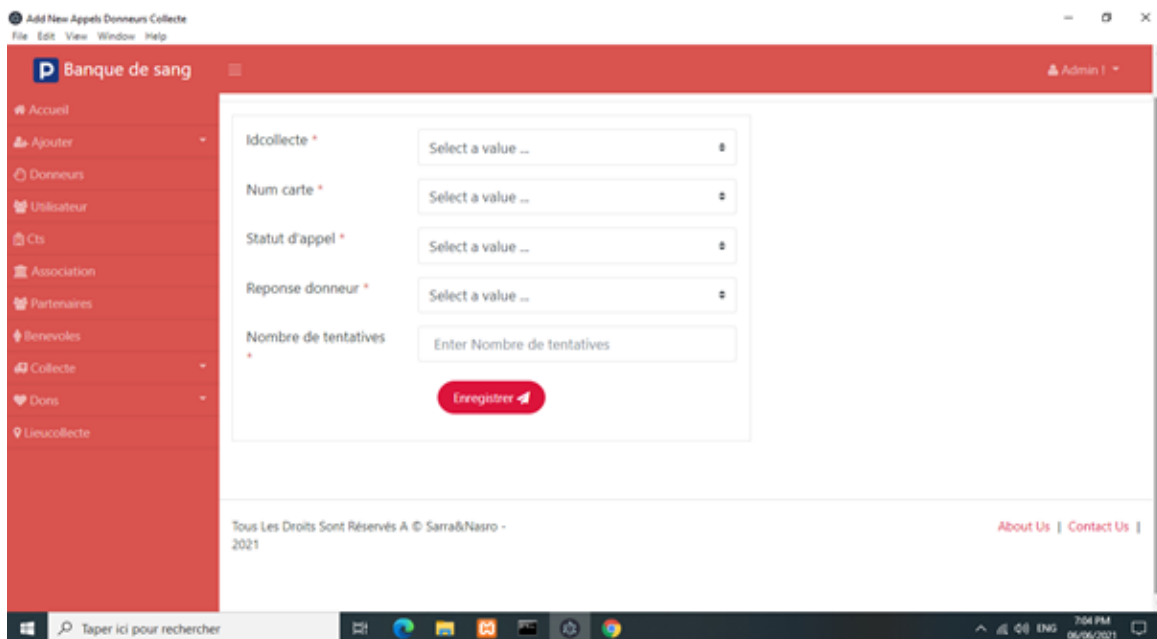


Figure 20 : La page ajouter donneur collecte

3.5.7. La page appels bénévoles et donneurs

Dans cette page l'administrateur doit remplir le formulaire en renseignant l'identifiant de la collecte, le numéro de la carte d'un donneur/l'identifiant du bénévole, le statut de l'appel (Bénévole/Donneur contacté, Pas de réponse, Faux numéro), la réponse du donneur/bénévole (Sera présent, ne pourra pas être présent, n'est pas sûr de venir, Aucune réponse) et le nombre de tentatives d'appels



The screenshot shows a web browser window titled 'Add New Appels Donneurs Collecte'. The page has a red header with the logo 'Banque de sang' and a user profile 'Admin'. A left sidebar contains navigation links: Accueil, Ajouter, Donneurs, Utilisateur, Cts, Association, Partenaires, Bénévoles, Collecte, Dons, and Lieucollecte. The main content area is a form with the following fields: 'Idcollecte' (dropdown menu), 'Num carte' (dropdown menu), 'Statut d'appel' (dropdown menu), 'Reponse donneur' (dropdown menu), and 'Nombre de tentatives' (text input). A red 'Enregistrer' button is at the bottom of the form. The footer contains the text 'Tous Les Droits Sont Réservés A © Sarra&Nasro - 2021' and links for 'About Us' and 'Contact Us'. The Windows taskbar at the bottom shows the search bar with the text 'Taper ici pour rechercher', several application icons, and the system tray with the date '09/06/2021' and time '7:04 PM'.

Figure 21 : La page appels donneur

3.6. Conclusion

Dans ce troisième et dernier chapitre, nous avons représenté les différents langages et logiciels que nous avons utilisés dans l'implémentation de notre application. Nous avons aussi montré certaines interfaces qui représentent les différents cas d'utilisation de notre application.

Conclusion Générale

La technologie apporte chaque jour de nouvelles innovations, réduisant ainsi le temps nécessaire pour faire les choses. Les CTS en Algérie n'utilisent pas un logiciel de gestion de banque de sang. Ils collectent les informations des donneurs à travers un registre écrit manuellement et vu que le nombre de donneurs augmente constamment cela rend la gestion très difficile.

Nous avons donc décidé de réaliser un logiciel afin d'améliorer la gestion des collectes de sang. Le système proposé peut être utilisé pour réduire le temps nécessaire pour fournir le sang requis aux personnes dans le besoin en cas d'urgence. Ce logiciel se charge de la gestion des bénévoles et donneurs, des partenaires, des associations, de la localisation des CTS et des lieux de collectes. Il facilite le contact avec les partenaires, les associations, les donneurs et les CTS.

Le logiciel sera utilisé aussi pour la gestion des donneurs de sang et de plaquettes, il présente les informations sur les donneurs de sang et de plaquettes et leurs disponibilités.

De plus, les informations contenues dans la base de données peuvent être partagées entre les CTS et les associations. Enfin, le logiciel devra aussi permettre l'importation et l'exportation de tables.

Dans ce rapport nous avons expliqué dans le premier chapitre le don de sang, ses conditions et ses types. Dans le deuxième chapitre nous avons abordé la conception du logiciel en utilisant UML.

Enfin, le troisième chapitre présente la partie pratique, nous avons abordé les différents outils de développement que nous avons utilisés, tels que les programmes front-end, la base de données, et nous avons expliqué en détail comment utiliser le logiciel et présenté ses différentes interfaces.

Les avantages apportés par notre solution sont que notre logiciel réponde aux besoins des CTS et améliore la qualité du service fourni par celui-ci.

Plusieurs perspectives peuvent être envisagées dans le prolongement de ce travail. Nous pouvons citer certaines perspectives dans ces quelques lignes :

En premier lieu, nous proposons l'utilisation de la géolocalisation des centres de transfusion ou bien les collectes mobiles. Afin de faciliter la présentation de leurs emplacements.

En deuxième lieu, Nous proposons aussi de développer une application mobile de gestion dans le but de lier les utilisateurs et faciliter le partage des données. Cette application, pourra contenir aussi une section spéciale pour les donneurs où ils pourront s'inscrire et accéder à l'application.

Enfin, avec le développement massif des produits RFID, il est important d'utiliser cette technologie sous forme de cartes. Les donneurs pourront disposer d'une carte de donneur dématérialisée à présenter lors de chaque don. Cette version permet également de mettre à jour les données personnelles, trouver plus facilement toutes les collectes, et de consulter et partager des témoignages de donneurs et de patients.

Annexe 1

Le modèle logique de données :

Donneurs : ID Donneur, Nom Donneur, Prénom Donneur, Date de naissance, Numéro de téléphone, Wilaya, Commune, Quartier, Adresse , Sexe, poids, Groupage, Date Premier Don, Date dernier don, Type du dernier don (Sang complet, Don de plaquettes), Nombre de dons, Actif (Oui/Non), Numéro de carte, QR Code, Donneur de plaquette (Oui/Non), Remarque (Lors du questionnaire de don de sang ou lors des appels téléphoniques)

Dons : ID don, ID Collecte, ID Donneur, Type de don (Sang complet, Don de plaquettes), Quantité, Groupage

Collecte : ID Collecte, Type collecte (Fixe/mobile), Date, Heure début, Heure fin, Nombre de dons, ID Lieu, Id Partenaire 1, Id Partenaire 2, Id Partenaire 3, Id Partenaire 4, ID CTS, ID Association1, ID Association2, ID Association3

nbrCTS: ID CTS, Nom Long, Nom Court, Téléphone, Coordonnées X, Coordonnée Y

Partenaires: Id Partenaire, Nom long, Nom court, Type Partenaire, Nom Contact 1, Prénom Contact1, Téléphone Contact 1, Nom Contact 2, Prénom Contact2, Téléphone Contact 2, Wilaya, Commune, Quartier, Adresse, Coordonnées X, Coordonnée Y, Nombre de collecte, Nombre de dons, Nombre moyen de dons

Associations: ID Association, Nom long, Nom court, Nom Contact 1, Prénom Contact1, Téléphone Contact 1, Nom Contact 2, Prénom Contact2, Téléphone Contact 2, Wilaya, Commune, Quartier, Adresse, Coordonnées X, Coordonnée Y, Nombre de collecte, Nombre de dons, Nombre moyen de dons

Lieu de collectes: Id Lieu, Nom long, Nom Court, Type (CTS, Entreprise, Mosquée, Place publique, Université, Administration, Siège d'association), Wilaya, Commune, Quartier, Coordonnées X, Coordonnée Y, Effectif, Nombre moyen de dons, Nom Contact 1, Prénom Contact1, Téléphone Contact 1, Nom Contact 2, Prénom Contact2, Téléphone Contact 2

Appels_Donneurs_Collecte: ID Collecte, Id Donneur, Statut de l'appel (Donneur contacté, Pas de réponse, Faux numéro) , Réponse du donneur (Sera présent, ne pourra pas être présent, n'est pas sûr de venir, Aucune réponse), Nombre de tentatives d'appels

Appels_Bénévoles_Collecte: ID Bénévole , Id Donneur, Statut de l'appel (Bénévole contacté, Pas de réponse, Faux numéro), Réponse du bénévole (Sera présent, ne pourra pas être présent, n'est pas sûr de venir, Aucune réponse), Nombre de tentatives d'appels

Bénévoles: Id Bénévole, Nom, Prénom, Date de naissance, Numéro de téléphone, Wilaya, Commune, Quartier, Adresse , Sexe, Nombre de collecte, Date première participation, Disponibilité, ID Association

Donneurs_Collecte: Id Collecte, ID Donneurs, Nom Donneur, Prénom Donneur , Date de naissance, Numéro de téléphone, Wilaya, Commune, Quartier, Adresse , Sexe, poids, Groupage, Date Premier Don, Date dernier don, Type du dernier don (Sang complet, Don de plaquettes), Nombre de dons, Actif (Oui/Non), Numéro de carte, QR Code, Donneur de plaquette (Oui/Non), Remarque (Lors du questionnaire de don de sang ou lors des appels téléphoniques)

Utilisateurs : ID Utilisateur, Nom, Prénom, Téléphone, Organisme, Type d'organisme (CTS, Association) Fonction, Type d'Accès (Complet, Consultation uniquement, Appels uniquement, Collecte Uniquement, Administration) , Utilisateur (Login), Mot de Passe, ID Association, ID CTS)

Annexe 2

Mise en service du logiciel

Avant de commencer il faut installer le GIT via le site <https://git-scm.com/> en suite le node js via le site <https://nodejs.org/fr/>.

Déplacez-vous vers le dossier du node js dans Terminal installé Électron avec la commande suivante :

```
npm install g electron
```

Exécutez ensuite les commandes suivantes afin de lancer le projet :

```
# Clonez le dépôt Quick Start
$ git clone https://github.com/electron/electron-quick-start

# Allez dans le dépôt
$ cd electron-quick-start

# Installez les dépendances et lancez l'app
$ npm install && npm start
```


Webographies

- [1] www.dondesang.efs.sante.fr
- [2] www.associationcassandra.org/don-de-plasma
- [3] www.fr.wikipedia.org/wiki/Banque_du_sang
- [4] www.dspace.univ-tlemcen.dz/bitstream/112/14674/1/Ms.GBM.Benchikh.pdf
- [5] www.fr.wikipedia.org/wiki/Processus_unifi%C3%A9
- [6] www.hopital-dz.com/upload/09-2017/article/TransfSang2016.pdf
- [7] www.academia.edu/37498927/Blood_Bank_Management_System
- [8] www.iraj.in/journal/journal_file/journal_pdf/3-299-147711892527-31.pdf
- [9] www.irjet.net/archives/V7/i6/IRJET-V7I6240.pdf
- [10] www.fr.wikipedia.org/wiki/Diagramme_de_classes
- [11] www.ehuoran.dz/dondusang.html
- [12] www.donneurdesang.be/
- [13] [www.fr.wikipedia.org/wiki/UML_\(informatique\)](http://www.fr.wikipedia.org/wiki/UML_(informatique))
- [14] www.phpmyadmin.net/
- [17] <https://developer.mozilla.org/fr/docs/Web/HTML>
- [18] https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade
- [24] <https://www.hostinger.fr/tutoriels/cest-quoi-bootstrap/>
- [25] <https://www.techno-science.net/glossaire-definition/Adobe-Photoshop.html>

Bibliographie

- [15] Harmening, D. M. (2018). *Modern blood banking & transfusion practices*. FA Davis.
- [16] Rudmann, S. V. (2005). *Textbook of blood banking and transfusion medicine*. Elsevier Health Sciences.
- [19] Kruchten, P. (2004). *The rational unified process: an introduction*. Addison-Wesley Professional.
- [20] Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, 14(6), 80-83.
- [21] Welling, L., & Thomson, L. (2003). *PHP and MySQL Web development*. Sams Publishing.
- [22] UML, O., & MOF, I. (2011). *The Unified Modeling Language UML*.
- [23] Bouali, S., Ghozali, A., Deba, T., Ayad, S., & Noubadji, S. (2017). Promotion du don de sang total au centre de transfusion sanguine CHU d'Oran-Algérie, années 2012–2016. *Transfusion Clinique et Biologique*, 24(3), 325.