

## *Table de matière*

Introduction.....	1
Chapitre1 : Etat de lieu .....	2
Introduction .....	2
1. Cadre du projet.....	2
1.1. Etude de l'existant.....	2
1.2. Problématique .....	2
1.3. Solution proposée.....	3
1.4. Sujet du projet.....	3
2. Spécification des besoins .....	3
2.1. Besoin Fonctionnel : .....	4
2.1.1. Description textuelle : .....	4
2.1.2. Les diagrammes de cas d'utilisation .....	5
2.1.3. Les interfaces : .....	6
2.2. Identification des besoins non fonctionnels : .....	6
3. Méthodologie de travail adoptée : .....	7
Chapitre 2: Analyse et Conception .....	8
Introduction .....	8
1. Diagramme de cas d'utilisation détaillés .....	8
1.1. Le diagramme de cas d'utilisation partie web« consulter les donateurs» .....	8
1.2. Le diagramme de cas d'utilisation partie mobile« crée un rendez-vous» .....	9
2. Diagramme de classe : .....	10
3. Diagramme de séquence: .....	12
3.1. Diagrammes des séquences détaillé « partie application web ».....	12
3.1.1. Diagramme de séquence détaillé « Authentification».....	12
3.1.2. Diagramme de séquence détaillé « gestion des analyses » .....	13
3.1.3. Diagramme de séquence détaillé « ajouter des analyses ».....	14
3.1.4. Diagramme de séquence détaillé « supprimer des analyses ».....	15
Conclusion.....	15
Chapitre 4 : réalisation .....	16
Introduction .....	16
1. Choix techniques.....	16
1.1. Environnement logiciel.....	16
1.2. Choix des frameworks .....	18
2. Présentation des interfaces de l'application.....	21

2.1.	Application Web .....	21
2.1.1.	Partie Administrateur .....	21
2.1.1.1.	Authentification .....	21
2.1.1.2.	Gestion des utilisateurs .....	22
2.1.2.	Partie Infirmier .....	23
2.1.2.1.	Consultation de la liste des donateurs.....	23
2.2.	Application mobile : .....	23
2.2.1.	Espace d'inscription .....	23
2.2.2.	Authentification.....	24
2.2.3.	Liste des demandes.....	25
2.2.4.	Créer un rendez-vous .....	25
	Conclusion.....	26
	Conclusion générale.....	27

## Table de figure

Figure I.2: interface Web «authentication » .....	6
Figure I.3: modèle en V .....	7
Figure II.1: diagramme de cas d'utilisation « consulter liste des donateurs» .....	8
Figure II.2: diagramme de cas d'utilisation « créer un rendez-vous» .....	9
Figure II.3: diagramme classe .....	11
Figure II.4 : diagramme de séquence « Authentification» .....	12
Figure II.5 : diagramme de séquence « Gestion des analyses» .....	13
Figure II.6: diagramme de séquence « ajouter des analyses» .....	14
Figure II.7 : diagramme de séquence « supprimer des analyses» .....	15
Figure IV.1 : PowerDesigner .....	16
Figure IV.2 : Eclipse IDE.....	16
Figure IV.3 : Apache Tomcat.....	17
Figure IV.4 : Apache Tomcat.....	17
Figure IV.5 : Apache Tomcat.....	17
Figure IV.6 : WampServer .....	17
Figure IV.7 : Adobe Photoshop.....	18
Figure IV.8 : logo JavaJEE .....	18
Figure IV.9: logo JavaJEE .....	19
.....	19
Figure IV.10:Hibernate .....	19
Figure IV.11 – Logo Spring Boot .....	20
.....	20
Figure IV.12: logo angular .....	20
Figure IV13. : Logo flutter.....	21
Figure IV14 : Interface d'authentification .....	21
Figure IV. 15 : Interface de gestion de la liste des utilisateurs .....	22
Figure IV16 : Interface de modifier un employé existant .....	22

Figure IV17 : Interface de consultation des donateurs.....	23
Figure IV18 : Interface d'inscription .....	24
Figure IV19 : Interface d'authentification .....	24
Figure IV19 : Interface de liste des demandes .....	25
Figure IV.20 : Interface Pour créer un rendez-vous.....	25

## **Introduction**

Le sang est une composante déterminante et vitale pour l'être humain. Son manque est très périlleux pour la vie des personnes en nécessité. Le sang, tissu vivant irrigue tous les organes, leur apporte oxygène et éléments nutritifs et les débarrasse de leurs déchets.

Le don du sang est un geste généreux. Il représente à ce jour la seule chance de soins pour les malades qui ils en ont besoin.

A cet égard, vu les victimes d'accidents du travail ou des routes ainsi que les maladies qui nécessitent l'intervention chirurgicale, la demande du sang n'a pas cessé d'augmenter.

Nous nous intéresserons dans ce projet à définir une méthodologie pour que le malade contacte directement les gens pour lui faire un don du sang. C'est ce qui constitue notre projet de fin d'années.

On commence par situer le projet dans son contexte, puis définir ses objectifs et ensuite spécifier les besoins. Dans le deuxième chapitre, on va traiter la partie de la conception. Alors que dans le troisième chapitre, On va présenter l'environnement de développement ainsi que la réalisation et en fin on terminera par la conclusion générale.

# **Chapitre1 : Etat de lieu**

## **Introduction**

Ce chapitre concerne la présentation générale du projet. On s'intéresse premièrement à la présentation du cadre de projet. Deuxièmement, on passe à la présentation des solutions proposées. Troisièmement, on va indiquer les besoins fonctionnels et non fonctionnels.

## **1. Cadre du projet**

### **1.1. Etude de l'existant**

Afin de comprendre le projet, on va présenter dans cette section une étude de l'existant dans la Tunisie, puis on va présenter notre problématique. Ensuite on va proposer une solution à cette problématique. Enfin, on va déduire le sujet de notre projet.

Vue la pandémie de covid 19 qui a commencer en mars 2020, les donateurs de sang ont évité par peur d'être contaminer

Après un cri d'alarme lancé lors du premier confinement en mars 2020 dû à la pandémie covid 19 et attirer l'attention des citoyens sur l'importance de donner leur sang en cette période qui a engendré un grand besoin de dons du sang dû à la peur de donateur d'être contaminer .

Le Centre national de transfusion sanguine (CNTS) se mobilise, face à l'urgence de renflouer les réserves nationales de sang.

### **1.2. Problématique**

D'après le professeur Hmida slama Directeur Général du Centre National de Transfusion Sanguine, le nombre de donateurs n'augmente pas mais au contraire le nombre de personnes ayant besoin de sang, à l'échelle nationale et internationale, a considérablement augmenté, alors que, le nombre de donateurs a enregistré une baisse remarquable pendant l'apparition de cette épidémie.

Le professeur Slama, a également déclaré que le produit sanguin a une durée de vie limitée dans le temps. A titre d'exemple, les concentrés de globules rouges sont conservés jusqu'à 42 jours seulement, d'où la nécessité de renouveler les stocks d'une manière permanente et continue et l'obligation de continuer de donner le sang de manière régulière.

Il est important ici de souligner que le produit sanguin labile (produit issu d'un don de sang et destiné à être transfusé à un patient) provient exclusivement de donateurs de sang bénévoles. Il ne s'agit pas d'un produit qu'on peut fabriquer. Et donc, il n'a pas d'autres sources que l'Homme.

### **1.3. Solution proposée**

Suite à ces problèmes on devra créer une application qui aide le Centre national de transfusion sanguine .Cet application servira à augmenter le nombre des donateurs de sang.

Nous sommes amenées à réaliser une application mobile qui permettra d'accéder rapidement à toutes les informations et de créer un espace personnel à chacun.

L'application permettra aussi de gagner le temps dans les cas urgents pour intervenir le plus vite possible et sauver ainsi plus de vies. Elle donnera en plus des conseils pour les donateurs (les principales informations sur le don de sang).

Cette application va aussi créer un historique de don pour chaque donateur (sauvegarde des données personnelles et la date du dernier don).D'un autre côté, on devra réaliser une application web qui donne une vision sur les informations qui concerne les donateurs, les demandeurs du sang et également les analyses.

### **1.4. Sujet du projet**

Notre projet s'intitule : application web et mobile pour gérer les processus de don du sang.

En effet, cet application sera un outil intermédiaire entre le demandeur qui a besoin du sang et le donateur .la partie web que nous allons créer facilitera la procédure de don du sang.

## **2. Spécification des besoins**

La spécification des besoins consiste à comprendre le contexte du système. Il s'agit de spécifier ses acteurs qui interagissent avec lui ainsi que les besoins fonctionnels et non-fonctionnels. Ces deux paliers seront le sujet de cette partie.

## **2.1. Besoin Fonctionnel :**

Dans cette partie on va présenter tout d'abord notre besoin par description textuelle. Ensuite, on va présenter un diagramme de use case. Enfin, on va exposer les interfaces graphiques.

### **2.1.1. Description textuelle :**

Nous exposons dans cette sous-section les différents besoins fonctionnels auxquels doit répondre notre application. En effet, on doit créer une application web permettant de réaliser les fonctionnalités suivantes pour l'infirmier ainsi que pour l'administrateur qui consiste à :

#### **Infirmier**

- gérer les donateurs pour consulter leurs liste;
- gérer les analyses pour être sûr que le donneur est prêt pour le don ou non ;
- gérer les utilisateurs afin de consulter, modifier, supprimer et rechercher une personne.

#### **Administrateur**

- gérer les utilisateurs ce qui consiste à consulter, modifier, supprimer et rechercher une personne ;
- gérer les infirmiers ce qui consiste à consulter, modifier, supprimer et rechercher un infirmier.

On doit également créer une application mobile permettant de réaliser les fonctionnalités suivantes selon le type de l'utilisateur.

#### **Demandeur**

Le Demandeur va pouvoir de demander du sang en cas d'urgence afin que le donateur intervient le plus tôt possible pour faire le don.

#### **Donateur**

Le donateur il va consulter la liste des demandes dans le but de sauver les malades.

D'une autre part il va Créer un rendez-vous afin de se déplacer à l'association pour faire les analyses et faire le don.



### 2.1.2. Les diagrammes de cas d'utilisation

Un cas d'utilisation définit l'interaction du système et sa manière d'utilisation. Il permet de décrire les exigences fonctionnelles de l'application.

Le diagramme de cas d'utilisation général présente et définit les fonctionnalités dont dispose l'utilisateur dans notre application :

- gérer les donateurs pour assurer les dons du sang ;
- gérer les analyses afin de garantir que les donateurs sont en bon conditions pour faire les dons ;
- gérer les utilisateurs inscrits dans l'application ;
- gérer les infirmiers ;
- gréer des demandes dans le but de résoudre un cas d'urgence qui aura besoin du sang ;
- consulter les demandes afin de sauver les vies des autres personnes ;
- créer un rendez-vous pour aller à l'association pour faire le don.

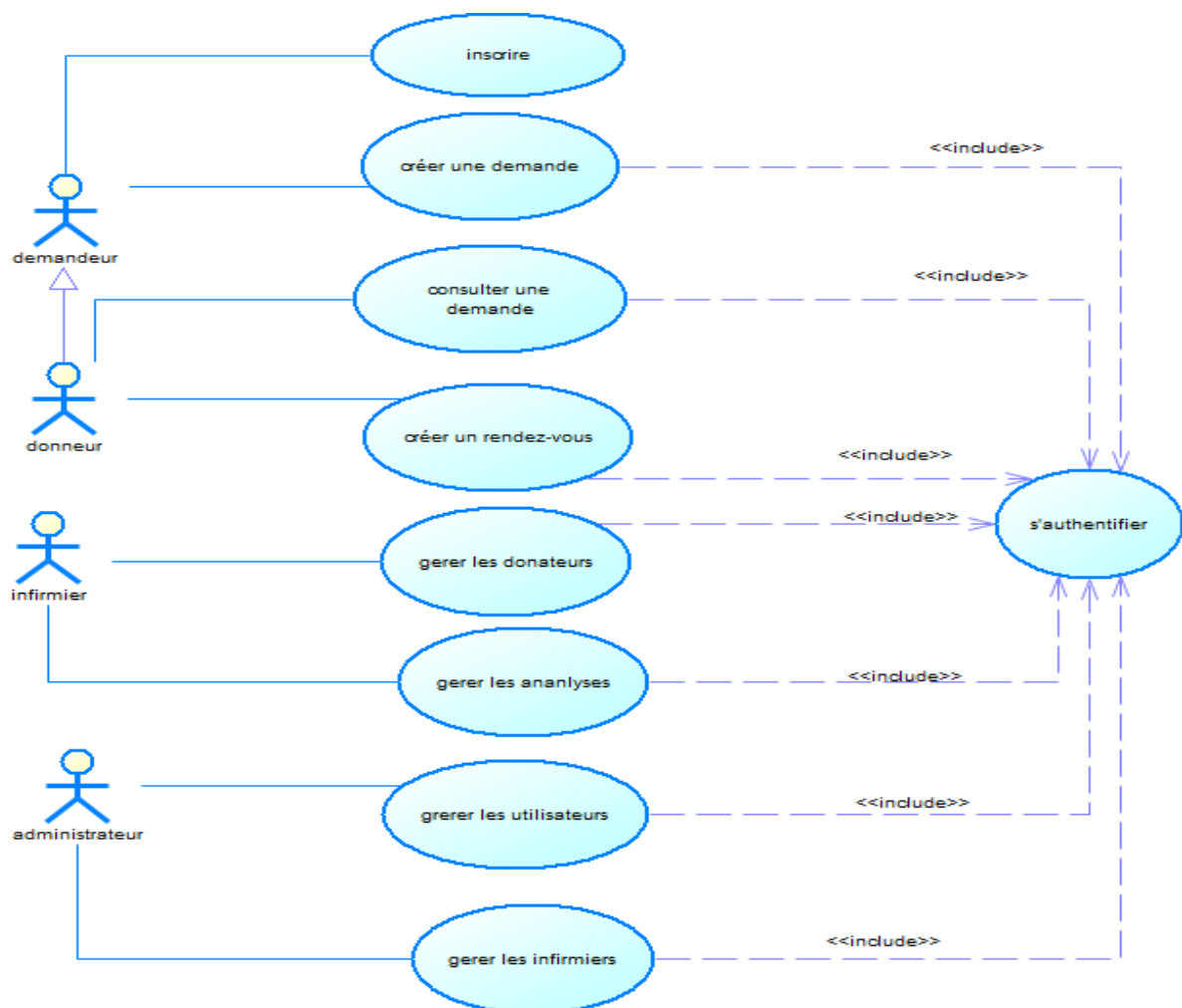


Figure I.1 : diagramme de cas d'utilisation général<<don du sang>>

### 2.1.3. Les interfaces :

Afin de maîtriser nos besoins fonctionnels, nous avons opté par les exprimer sous forme d'interfaces graphiques. Dans cette section, nous allons présenter cinq interfaces pour notre application web et mobile.

Dans l'application web, la première interface indique l'authentification, la deuxième la gestion des utilisateurs, la troisième exprime la gestion des analyses.

Dans l'application mobile, la première interface indique l'inscription, la deuxième l'authentification, la troisième exprime la liste des demandes, la quatrième l'ajout d'une demande et la cinquième est une interface pour prendre un rendez-vous.

#### Application web

##### - authentification :

Cette interface est dédiée aux infirmiers et aux administrateurs. Ces derniers écrivent les informations nécessaires pour accéder à la plateforme.

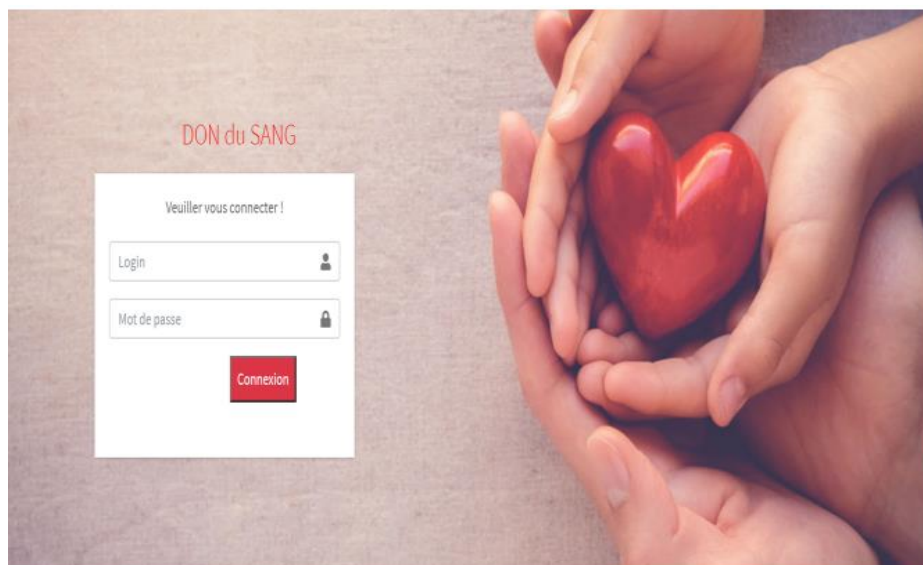


Figure I.2: interface Web «authentication »

### 2.2. Identification des besoins non fonctionnels :

Les besoins non fonctionnels décrivent toutes les contraintes auxquelles est soumis le système pour sa réalisation et son bon fonctionnement. En effet, notre application doit couvrir les besoins non fonctionnels suivants :

- **Facilité d'emploi:** faciliter l'apprentissage, d'utilisation, la préparation des données, l'interprétation des erreurs et le rattrapage en cas d'erreur d'utilisation ;
- **Extensibilité:** facilité avec laquelle un logiciel se prête à une modification ou à une extension des fonctions qui lui sont demandée ;
- **Fiabilité ou robustesse:** aptitude d'un produit logiciel à fonctionner dans des conditions normales.

### 3. Méthodologie de travail adoptée :

Le modèle du cycle de vie: La méthodologie est l'ensemble des techniques et méthodes qui servent de guide à l'élaboration des recherches ainsi qu'à l'orientation la démarche scientifique. C'est la mise en forme des données afin de les rendre analysables. En fait, on a adapté une des méthodologies classiques qui est le cycle en V, car il est le plus adéquat avec notre projet.

#### Modèle en V

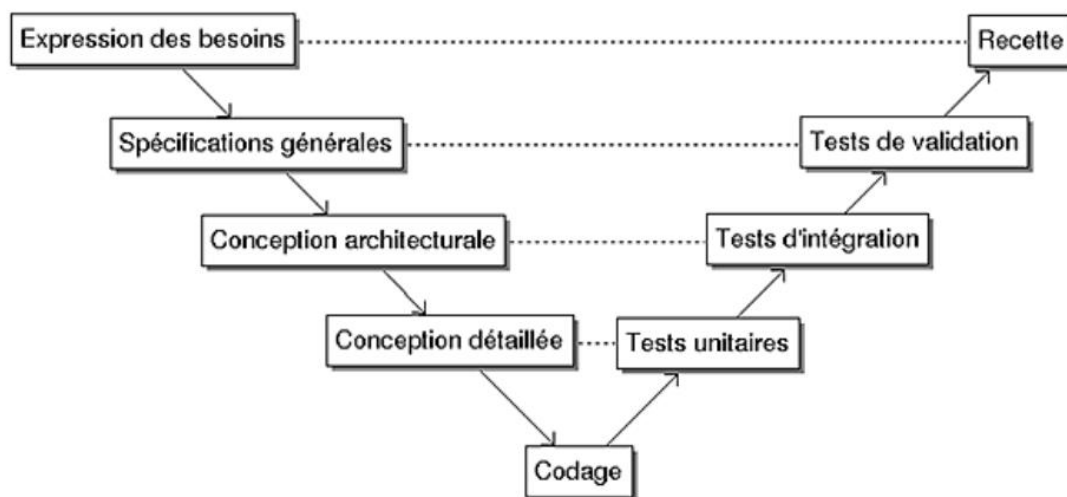


Figure I.3: modèle en V

### Conclusion :

Dans ce chapitre on a exposé la problématique traitée dans ce rapport. Elle nous permet de spécifier des besoins afin de déterminer le problème et la solution proposée jusqu'à atteindre l'objectif. Ce qui nous emmène à présenter la conception dans le chapitre suivant.

## Chapitre 2: Analyse et Conception

### Introduction

L'objectif du chapitre est de définir la phase de conception. Au début, on va présenter les diagrammes de cas d'utilisation détaillés. Ensuite, on va présenter l'application modélisation statique dans un diagramme de classe. Enfin, on va présenter une vue dynamique à travers la modélisation de diagramme de séquence.

### 1. Diagramme de cas d'utilisation détaillés

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale 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. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système.

#### 1.1. Le diagramme de cas d'utilisation partie web« consulter les donateurs»

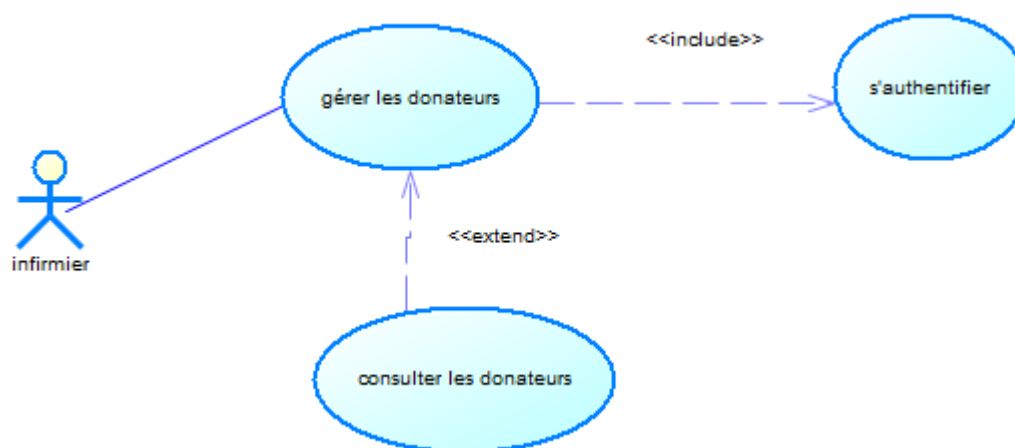


Figure II.1: diagramme de cas d'utilisation « consulter liste des donateurs»

Tableau 1 : Description de cas d'utilisation « consultation des donateurs »

Intitulé	Consulter les informations des donneurs
Acteur Principal	Infirmier
Acteur Secondaire	

Description	L'infirmier peut consulter les données des donateurs afin de déterminer leurs coordonnées.
Pré-Action	S'authentifie
Post-Action	L'infirmier consulte la liste des donateurs
Scénario nominal	1- L'infirmier s'authentifie 2-L'infirmier clique sur le champ de consulter donateurs. 3-L'infirmier consulte la liste des donateurs
Scénario d'exception	3-Erreur au niveau de l'Interface web

## 1.2. Le diagramme de cas d'utilisation partie mobile« crée un rendez-vous»

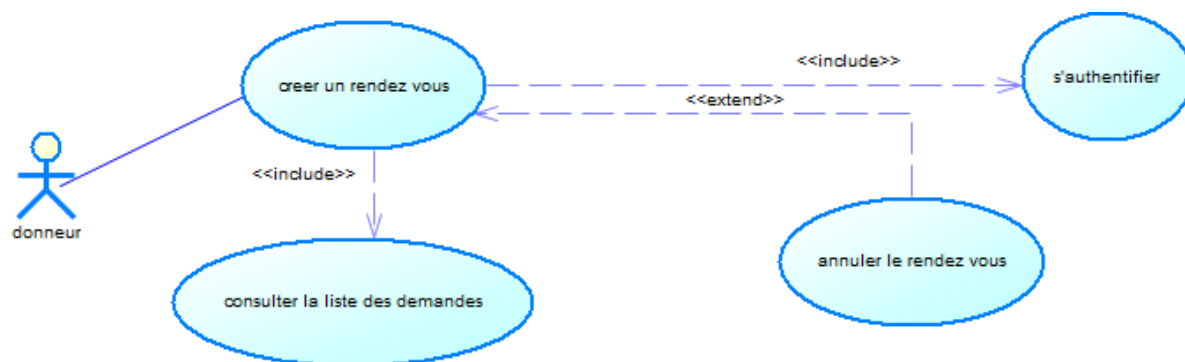


Figure II.2: diagramme de cas d'utilisation « créer un rendez-vous»  
Tableau 3 : Description de cas d'utilisation "Créer un rendez-vous "

Intitulé	Créer un rendez-vous
Acteur Principal	Un donateur
Acteur Secondaire	
Description	donateur peut prendre un rendez-vous pour faire le don du sang

Pré-Action	S'authentifie
Post-Action	donneur peut prendre un rendez-vous
Scénario nominal	1- Le donateur s'authentifie 2- Le donateur consulte la liste des demandes 3- Le donateur clique sur le bouton créer un rendez-vous devant l'annonce de demandeur 4- Le donateur choisir la date de rendez vous
Scénario d'exception	5-Erreur au niveau de l'Interface web

## 2. Diagramme de classe :

La figure II.4 présente le diagramme de classe qui constitue le processus de don du sang avec des relations existantes entre les différentes entités. On va présenter une description textuelle pour mieux comprendre le rôle des entités présentées dans le diagramme de classe .

- La classe personne : il présente les coordonnées des personnes. Ces dernières sont inscrites dans l'application (cin, nom, prénom, date de naissance, groupe sanguin, email, mot de passe, numéro téléphone, sexe) ;
- La classe demande : cette classe présente les personnes qui ajoutent une demande ;
- La classe response : cette classe représente les personnes qui répondent aux demandes des utilisateurs ;
- La classe analyse : cette classe contient les dates des analyses, la description et le résultat des analyses ;
- La classe don : cette classe contient la date de don du donateur ;
- La classe infirmier : cette classe représente les informations qui correspondent aux infirmiers (cin, nom, prénom, date de naissance, groupe sanguin, email, mot de passe, numéro téléphone, sexe, diplôme).

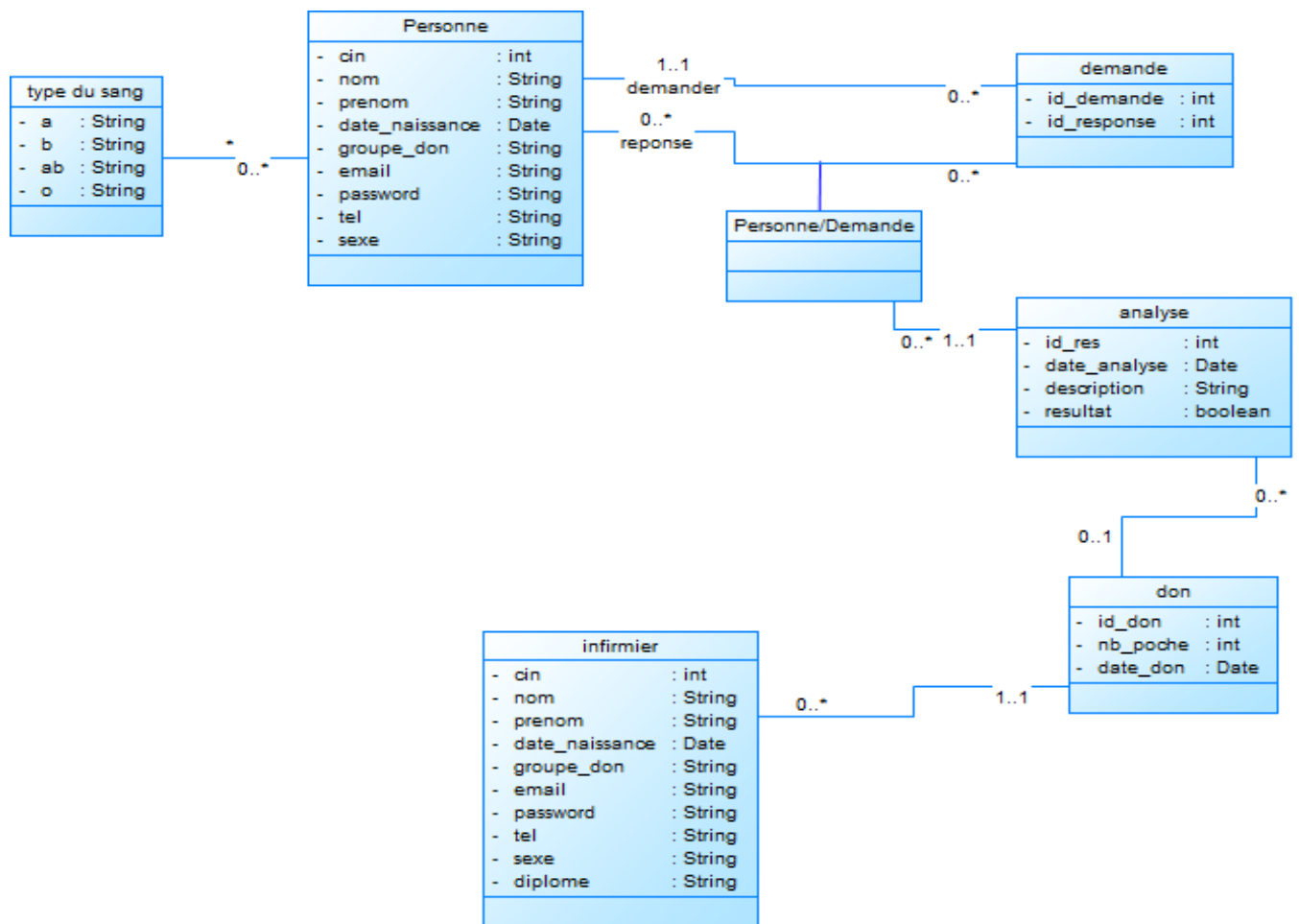


Figure II.3: diagramme classe

### 3. Diagramme de séquence:

Le rôle de diagramme de séquence est de représenter les vues dynamiques de l'application. En effet, il montre les coopérations entre les objets dans un intervalle de temps bien déterminé.

#### 3.1. Diagrammes des séquences détaillé « partie application web »

Cette partie concerne la partie web afin de bien gérer le processus de don.

##### 3.1.1. Diagramme de séquence détaillé « Authentification »

La figure ci-dessous montre la règle d'authentification

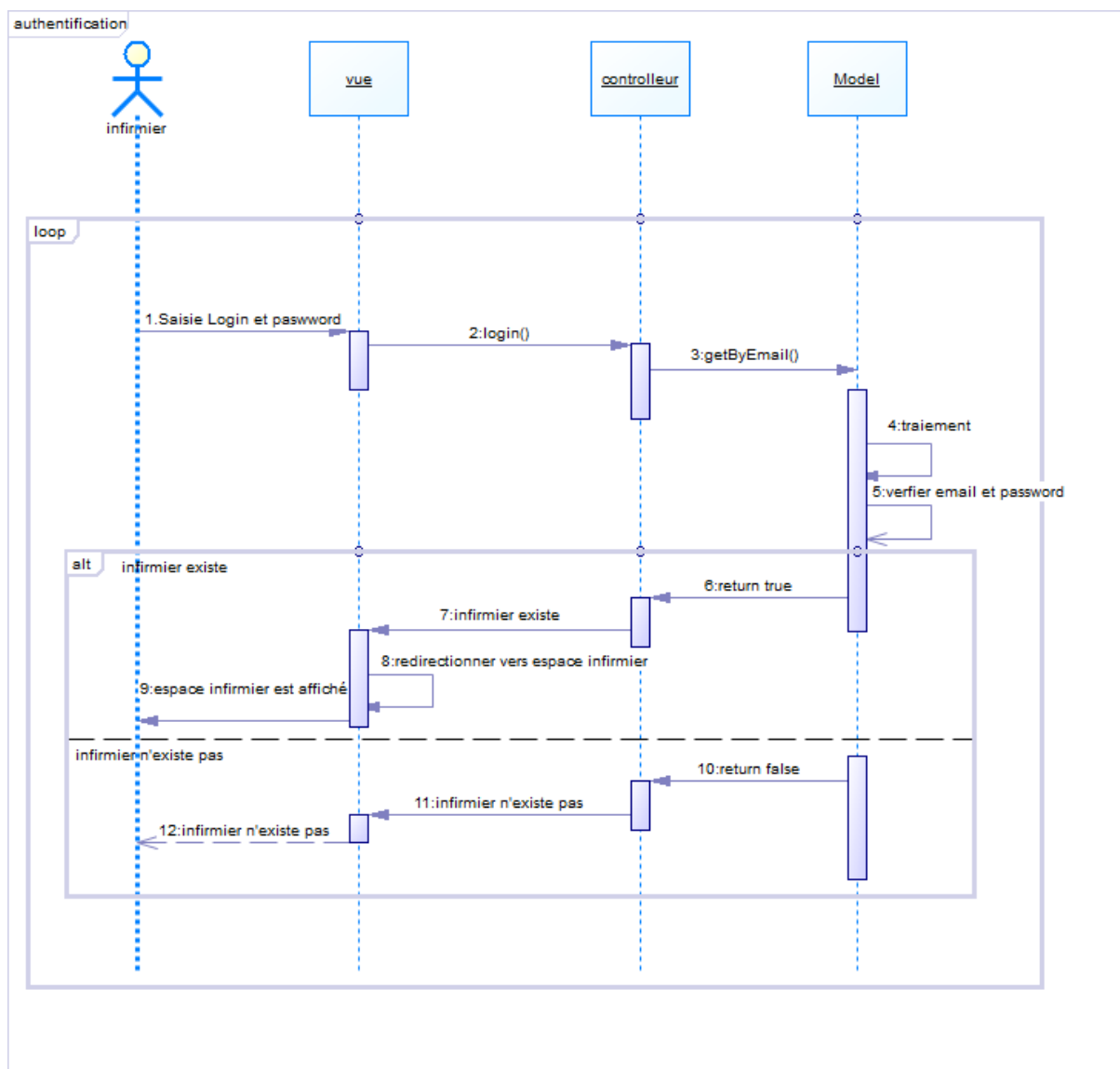


Figure II.4 : diagramme de séquence « Authentification »



### 3.1.2. Diagramme de séquence détaillé « gestion des analyses »

La figure ci-dessous montre l'enchaînement de l'application pour faire la consultation d'une liste des analyses, leur suppression et leur l'ajout.

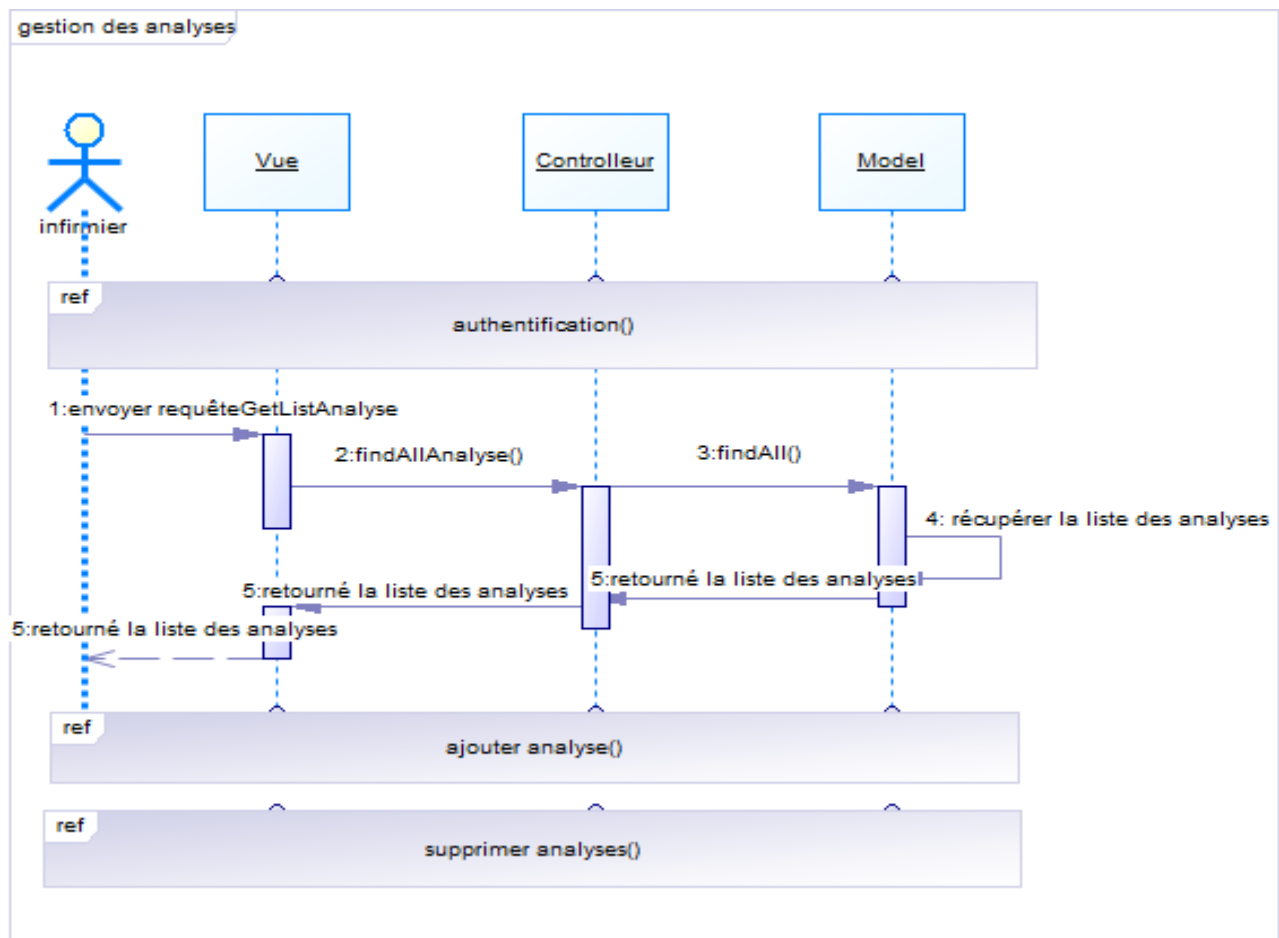


Figure II.5 : diagramme de séquence « Gestion des analyses»

### 3.1.3. Diagramme de séquence détaillé « ajouter des analyses »

Cette figure représente le scénario d'ajout d'une analyse

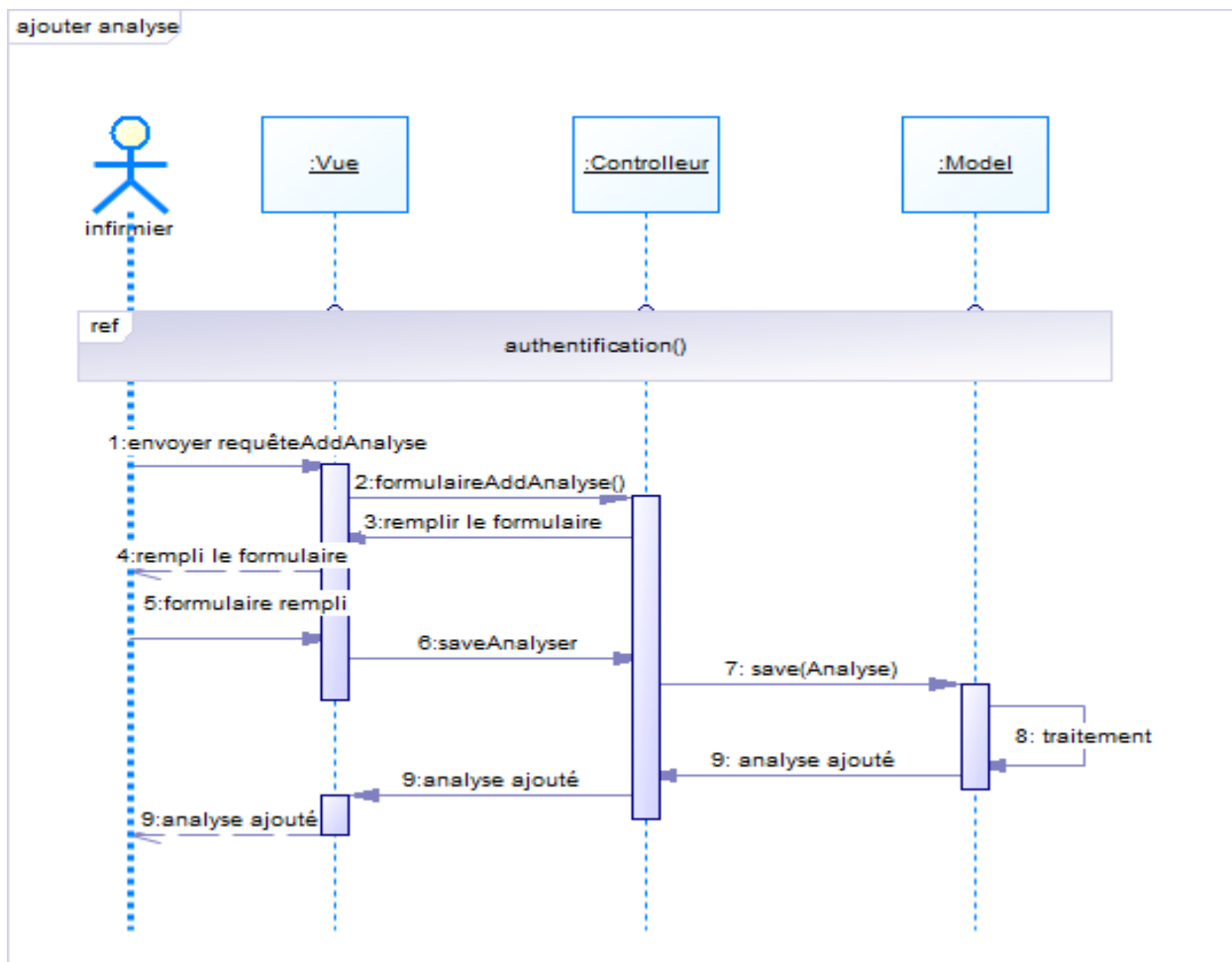


Figure II.6: diagramme de séquence « ajouter des analyses »

### 3.1.4. Diagramme de séquence détaillé « supprimer des analyses »

Cette figure comporte le traitement pour la suppression

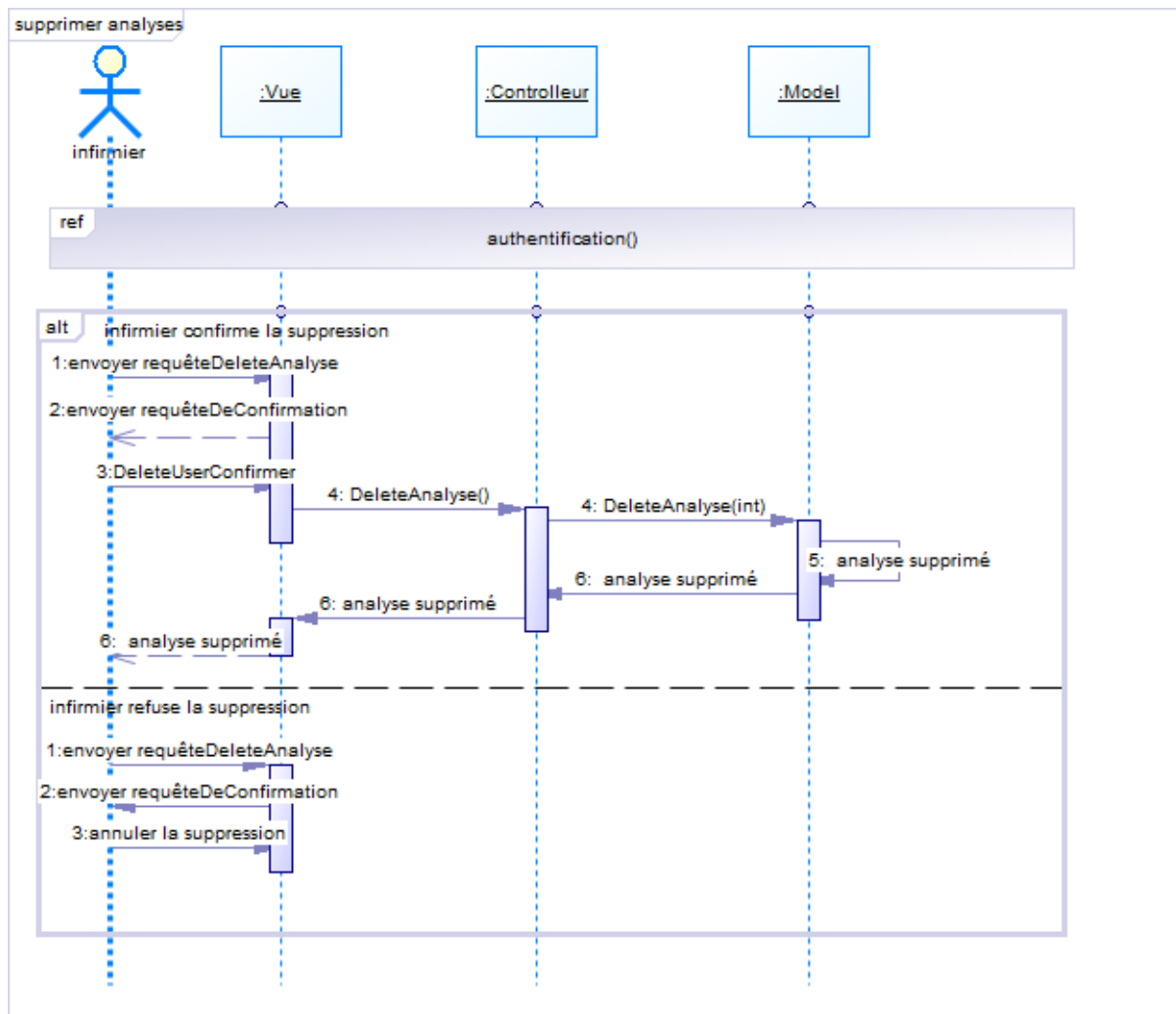


Figure II.7 : diagramme de séquence « supprimer des analyses»

## Conclusion

Dans ce chapitre on a exposé une analyse et une conception de notre application. Cette étude nous mène au chapitre réalisation.

# Chapitre 4 : réalisation

## Introduction

Après avoir achevé la conception du système, nous exposons dans ce chapitre les détails liés au développement de l'application. Nous commençons par décrire les choix techniques et les choix de développement.

### 1. Choix techniques

#### 1.1. Environnement logiciel

##### *PowerDesigner®*



Figure IV.1 : PowerDesigner

Pour la modélisation de notre application, nous avons choisi d'utiliser le logiciel PowerDesigner qui représente une solution graphique afin de faire la modélisation d'entreprise pour assumer les méthodologies et les notations standard et donner une ingénierie inverse et une génération de code automatisées via des modèles personnalisables.

##### *Eclipse IDE*



Figure IV.2 : Eclipse IDE

Concernant le développement de notre application, nous allons choisir l'environnement Eclipse IDE qui représente un environnement de développement libre. Ce dernier permet de créer des projets de développement dans de nombreux langages de programmation. Parmi lesquels on cite le langage java.

## ***Apache Tomcat***



Figure IV.3 : Apache Tomcat

Apache Tomcat est développé par Apache Software Foundation (ASF). Ce dernier est un serveur Web pour JSP. Il permet l'implémentation de Java Servlets et JavaServer Pages (JSP) pour lancer un environnement de serveur Java efficace.

## ***Visual Studio Code***



Figure IV.4 : Apache Tomcat

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS2.

Visual Studio Code est un éditeur de code source qui peut être utilisé avec une variété de langages de programmation, notamment java, JavaScript, Go, node.js et c++.

## ***android studio***



Figure IV.5 : Apache Tomcat

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux.

## ***Mysql***



Figure IV.6 : WampServer

MySQL est un système de gestion de bases de données relationnelles SQL open source développé et supporté par Oracle.

MySQL est développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multi-thread et multi-utilisateur.

## ***Adobe Photoshop***



Figure IV.7 : Adobe Photoshop

Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur, lancé en 1990 sur MacOS puis en 1992 sur Windows.

Photoshop possède son propre format de fichier (extension psd). Celui-ci permet de conserver distincts les différents calques formant l'image afin de les manipuler séparément. Le programme accepte également d'importer et d'exporter des fichiers d'image dans les formats les plus courants (extensions : gif, jpg, tif, png, etc.).

## **1.2. Choix des frameworks**

Un Framework est une implémentation réutilisable des designs patterns. L'objectif d'un Framework est généralement de simplifier le travail des développeurs en leur offrant une architecture "prête à l'emploi" et qui leur permette de ne pas repartir de zéro à chaque nouveau projet.

### **JEE**

JEE est une norme proposée par la société Sun, qui définit un standard de développement d'applications d'entreprises multi-niveaux, basées sur des composants. Cette norme s'appuie entièrement sur le Java et comme conséquence il bénéficie des avantages et inconvénients de ce langage, en particulier une bonne portabilité et une maintenabilité du code.



Figure IV.8 : logo JavaJEE

## Spring

C'est un framework libre pour construire et définir l'infrastructure d'une application java, dont il facilite le développement et les tests. Spring est considéré comme un conteneur dit « léger ». Il s'appuie principalement sur l'intégration de trois concepts clés :

- L'inversion de contrôle qui permet de réduire la dépendance d'une classe à un algorithme particulier ou une configuration particulière quand les méthodes de cette classe ou d'une autre sont utilisées pour effectuer une fonction complexe.
- La programmation orientée aspect qui propose de séparer le code technique du code métier d'une application pour des préoccupations transversales qu'elles soient techniques ou architecturales.
- Une couche d'abstraction qui permet d'intégrer d'autres frameworks et bibliothèques avec une plus grande facilité.



Figure IV.9: logo JavaJEE

## Hibernate

Hibernate query Language (HQL) est développé par la société jboss dans le domaine de l'architecture java jee. Ce dernier, est une solution pour résoudre le problème de mapping des objets relationnel (ORM). Hibernate permet donc de représenter une base de données en objets Java et vice versa.



Figure IV.10: Hibernate

## Spring Boot

C'est un micro framework qui a notamment pour but de faciliter la configuration d'un projet Spring et de réduire le temps alloué au démarrage d'un projet.



Figure IV.11 – Logo Spring Boot

## Angular

Angular est une plateforme de développement qui permet de créer des applications web dynamiques et immersives.

Il est livré avec de nombreux blocs de construction prêts à l'emploi qui couvre la plupart des scénarios courants dans le développement d'une application web. Il y a aussi une séparation claire des rôles des différents éléments :

- Services : Éléments injectables utilisés pour consommer des données à partir d'une API ou pour partager un état entre plusieurs composants.
- Composants : blocs de construction de l'interface utilisateur qui consomment des services. Ils peuvent être imbriqués les uns dans les autres grâce à des sélecteurs de directives structurelles.
- Directives : Divisées en directives structurelles et attributs. Les directives structurelles (ex. ngFor) Manipulent le DOM (Document Object Model) alors que les directives d'attribut font partie d'éléments et contrôlent leur style et leur état.
- Pipes - Utilisés pour formater la façon dont les données sont affichées dans la couche de vue
- Modules - Blocs exportables de l'application qui isolent les composants, les directives, les services et les routes.



Figure IV.12: logo angular



## Flutter

Flutter est un framework développé par Google, le plus récent de tous. De ce fait, les ingénieurs ont pu observer les points forts et les faiblesses de chaque outil existant pour n'en extraire que la quintessence.



Figure IV13. : Logo flutter

## 2. Présentation des interfaces de l'application

### 2.1. Application Web

Nous allons présenter dans cette partie certaines interfaces développées pour notre application afin de faciliter les procédures de don.

#### 2.1.1. Partie Administrateur

Dans cette partie l'administrateur peut réaliser la gestion des utilisateurs et la gestion des infirmiers.

##### 2.1.1.1. Authentification

Dans cette vue l'administrateur saisie son identifiant et son mot de passe.

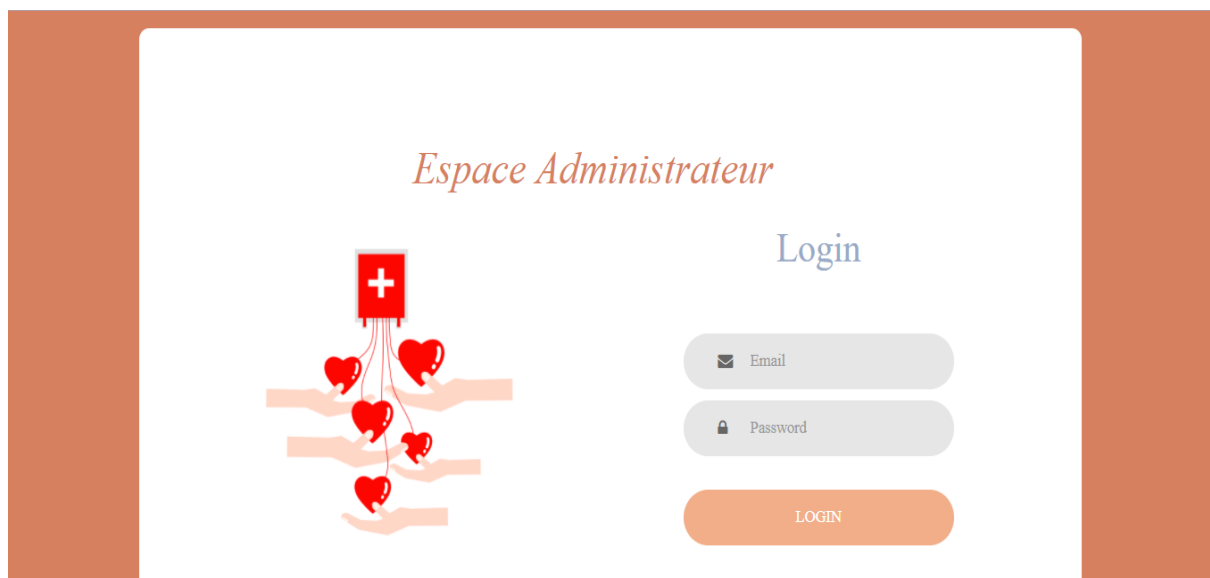


Figure IV14 : Interface d'authentification

### 2.1.1.2. Gestion des utilisateurs

Cette interface comporte tous les actions de CRUD tel est le cas supprimer, modifier et rechercher pour bien conduire le processus de gestion pour les utilisateurs.

**ESPACE ADMINISTRATEUR**

**liste des utilisateurs:**

nom

Email	Telephone	Sexe	Supprimer	Modifier
mili@gmail.com	53697887	femme	<input type="button" value="SUPPRIMER"/>	<input type="button" value="MODIFIER"/>
hendadouha@gmail.com	5282582	femme	<input type="button" value="SUPPRIMER"/>	<input type="button" value="MODIFIER"/>
hendadouha16@gmail.com	9923232	femme	<input type="button" value="SUPPRIMER"/>	<input type="button" value="MODIFIER"/>
benhenda@gmail	95941170	femme	<input type="button" value="SUPPRIMER"/>	<input type="button" value="MODIFIER"/>
adouha16@episousse.com.tn	968585	femme	<input type="button" value="SUPPRIMER"/>	<input type="button" value="MODIFIER"/>

**gerer les utilisateurs**  
    > Liste des utilisateurs  
**gerer les infirmiers**  
**consulter les demandes**  
**consulter les donneurs**

Figure IV. 15 : Interface de gestion de la liste des utilisateurs

### Editer un utilisateur existant

L'administrateur va cliquer sur le bouton editer qui se trouve devant l'utilisateur concerné.Par la suite une formulaire de modification est affiché .donc l'administrateur va modifier n'importe quel information se trouve dans le formulaire sauf que le code d'un utilisateur ou son id.

**ESPACE ADMINISTRATEUR**

**modifier un utilisateur:**

Cin

nom

prenom

date-naissance

mot de passe

groupe don

**gerer les utilisateurs**  
    Liste des utilisateurs  
**gerer les infirmiers**  
**consulter les demandes**  
**consulter les donneurs**

Figure IV16 : Interface de modifier un employé existant

### 2.1.2. Partie Infirmier

Dans cette partie les infirmiers peuvent effectuer la gestion des analyses. Ainsi, ils peuvent consulter la liste des donateurs et la liste des demandes.

#### 2.1.2.1. Consultation de la liste des donateurs

Cette vue représente la liste des donateurs qui peuvent arriver au centre pour le don.



The screenshot shows a web application interface for nurses. On the left is a dark brown sidebar with the title 'ESPACE INFIRMIER' and a circular logo containing medical icons. Below the logo are three menu items: 'consulter les demandes', 'consulter les donneurs', and 'gerer les analyses'. The main content area has a light gray background and is titled 'liste des donateurs:' in bold orange text. Below the title is a table with five columns: 'Prénom', 'Nom', 'Email', 'Telephone', and 'Sexe'. The table contains two rows of data.

Prénom	Nom	Email	Telephone	Sexe
malek	bouaziz	95941170	femme	malek@gmail
dhouha	ben henda	5282582	femme	benhendadouha@gmail.cc

Figure IV17 : Interface de consultation des donateurs

## 2.2. Application mobile :

Nous allons présenter dans cette partie certaines interfaces développées pour notre application afin de faciliter la communication indirecte entre le donateur et le demandeur du sang.

### 2.2.1. Espace d'inscription

La Figure IV18 montre le scénario d'inscription .quel que soit demandeur ou utilisateur, il peut faire l'inscription.

prenom

Password

date de naissance

Email

Numero de telephone

cin

sexe

type de sang

**SIGN IN**

Figure IV18 : Interface d'inscription

### 2.2.2. Authentification

Dans cette vue quel que soit demandeur ou utilisateur, il peut faire l'authentification.

*don du sang*

donner votre email

Donner votre mot de passe

**valider**

[créer un nouveau compte](#)

Figure IV19 : Interface d'authentification

### 2.2.3. Liste des demandes

Dans cette vue le donateur peut consulter la liste des demande de don du sang.

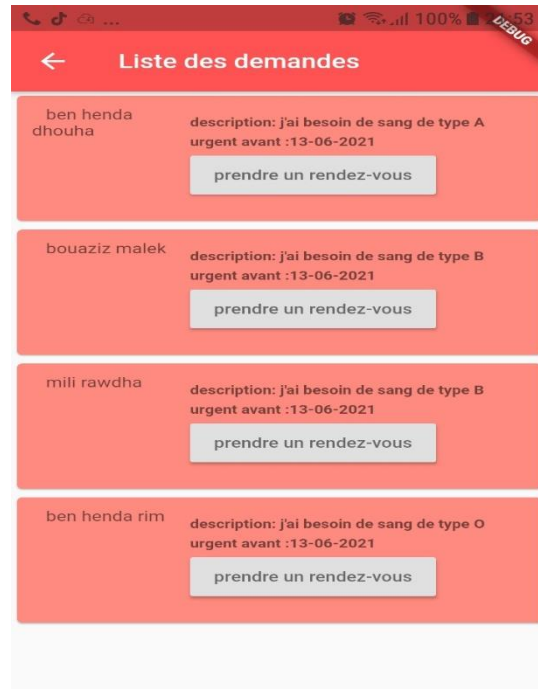


Figure IV19 : Interface de liste des demandes

### 2.2.4. Créer un rendez-vous

A partir de cette vue, le donateur peut créer un rendez-vous après la consultation de la liste.

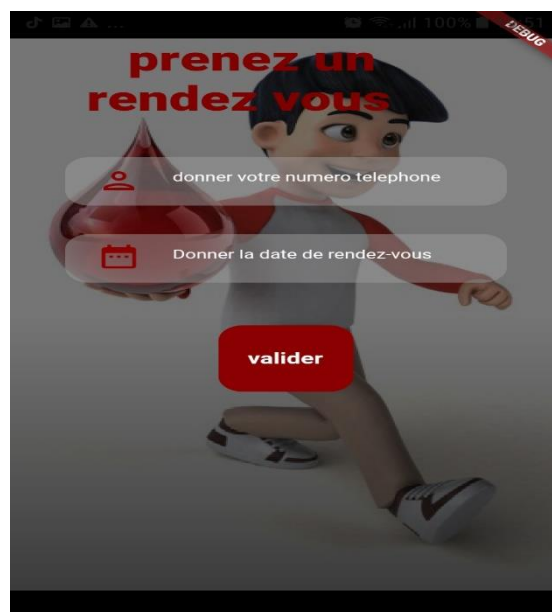


Figure IV.20 : Interface Pour créer un rendez-vous

## **Conclusion**

Dans ce chapitre, on vient d'exposer, en premier lieu, les choix techniques ainsi que les choix de Framework. Puis nous avons présenté la partie front de notre application à travers quelques interfaces web et mobile.

## Conclusion générale

Ce projet de fin d'année m'a donné l'occasion de développer une application web et mobile pour gérer le processus de don du sang.

Dans ce rapport on a présenté le chapitre d'état de lieu qui consiste à présenter l'école pluridisciplinaire internationale. Par la suite on a abordé le problème de don du sang en Tunisie. Ensuite on a donné des solutions pour résoudre ce problème, ce qui nous a amené à donner les besoins fonctionnels et non fonctionnels et à représenter le diagramme de cas d'utilisation général pour mieux comprendre les besoins de l'application. Après, l'état de lieu on a exposé dans le chapitre suivant l'analyse et la conception afin de mieux comprendre l'application du côté base de données et du côté de la réalisation. Ensuite, on a exposé le chapitre de réalisation.

Dans ce chapitre on a présenté les environnements techniques, l'environnement logiciel utilisé et les imprimés d'écrans d'interfaces de l'application.

Malgré toutes les difficultés rencontrées au niveau du processus de l'application et les contraintes de temps, ce travail était très intéressant puisqu'il m'a permis d'appliquer mes connaissances théoriques dans un contexte professionnel, de bénéficier de nouvelles connaissances tel que le back end, le front end et le développement mobile qui viennent compléter celles acquises tout au long de notre formation.

Cette application, peut être améliorée. En effet, plusieurs fonctionnalités peuvent être ajoutées à notre application et notamment, l'envoi d'une demande de don pour le plus proche donateur. On pourra également envoyer des rappels pour le don ou aussi créer une carte de donateur à chacun.

