



**Filière : Management des Systèmes
d'informations et de production
2^{ème} année**

Conception et réalisation d'une application de don du sang



*Don du **sang***

*Don de **vie***

Réalisé par :

- TOUIL ZOUHEIR
- KHALIL BALGA
- AKOUCH OUMAYMA
- AMINA AIT AZIZ
- KASSEL MOHAMMED ISSAM
- MAJNI NAWAL

Encadré par :

**Mme ZAYDI Hayat
Mme TIKITO Kawtar
M HOSNI Mohamed**

Remerciements

En préambule à ce projet, nous souhaitons adresser nos remerciements les plus sincères à toute l'équipe pédagogique de l'Ecole Supérieure des MINES RABAT et plus particulièrement le corps professoral du Département Informatique.

Nous profitons l'occasion pour remercier tout d'abord nos professeurs Madame ZAYDI Hayat, Madame TIKITO Kawtar et Monsieur HOSNI Mohamd qui n'ont pas cessé de nous encourager pendant toute la durée du projet, ainsi pour leur générosité en matière de formation et d'encadrement et la grande patience dont elles ont su faire preuve malgré leurs charges académiques et professionnelles.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis pour leur soutien et encouragements.

Merci à tous et à toutes.

Résumé

Nous avons procédé, dans le cadre de ce projet, à la conception et à la réalisation d'une plateforme de Don de Sang baptisée « **Gift of Blood** ». L'objectif est d'implémenter une solution générale informatisée qui non seulement aide les malades besoins de sang à avoir facilement des dons de Sang, mais aussi renforce les liens sociaux et résout partiellement le problème de manque de sang que connaît les centres de don marocaines.

Pour ce faire, nous avons commencé tout d'abord par faire une étude sommaire de marché pour identifier le besoin de l'utilisateur dans le contexte national. Cette étude a été suivie par la conception et le développement d'un site web.

La conception de ce projet a été réalisée en se s'appuyant sur le langage UML comme langage de modélisation et les méthodes agiles pour la conduite du projet. Ensuite nous avons traité le développement des applications moyennant principalement les langages web, architecturée autour du modèle MVC, d'où l'utilisation du Framework Laravel. Concernant la gestion de la base de données, nous avons opté pour le SGBD Mysql.

Abstract

As part of this project, we designed and built a Blood Donation platform called “**Gift of Blood**”. The objective is to implement a general computerized solution which not only helps patients in need of blood to have blood donations easily, but also strengthens social ties and partially solves the problem of blood shortage experienced by donation centers in Morocco.

To do this, we first started by doing a summary market study to identify the user need in the national context. This study was followed by the design and development of a website.

The design of this project was carried out using UML as a modeling language and agile methods for the conduct of the project. Then, we treated the development of the applications mainly thanks to web languages, structured around the MVC model, hence the use of the Laravel Framework. Concerning the management of the database, we opted for the Mysql DBMS.

Liste des figures

<u>Figure01</u> : Les Résultats du formulaire.....	15
<u>Figure02</u> : Diagramme de cas d'utilisation.....	21
<u>Figure03</u> : diagramme de séquence « devenir donneur »	23
<u>Figure04</u> : diagramme de séquence « déclencher cas d'urgence ».....	24
<u>Figure05</u> : Diagramme d'activité « devenir donneur ».....	25
<u>Figure06</u> : Diagramme d'activité « déclencher cas d'urgence ».....	26
<u>Figure07</u> : Diagramme de classes	27
<u>Figure08</u> : Architecture d'une application web.....	32
<u>Figure09</u> : Fichier de Routes.....	34
<u>Figure10</u> : Traitement de la requête par un contrôleur.....	35
<u>Figure11</u> : La page d'accueil.....	36
<u>Figure12</u> : Présentation des avantages de la solution.....	36
<u>Figure13</u> : Présentation de la page 'qui peut donner ?'.....	37
<u>Figure14</u> : devenir donneur.....	37
<u>Figure15</u> : La section 'login'.....	37
<u>Figure16</u> : Le Formulaire d'inscription.....	38
<u>Figure17</u> : La page de demande de sang.....	38
<u>Figure18</u> : La liste des donneurs.....	39
<u>Figure20</u> : La page 'Home'.....	39
<u>Figure20</u> : La page 'Home'2.....	40
<u>Figure21</u> : La liste des centres de don.....	40

Liste des sigles et abréviations

API: Application Programming Interface

CNTSH : Centre national de transfusion sanguine et d'hématologie

CSS: Cascading Style Sheets

HTML : HyperText Markup Language

HTTP : HyperText Transfer Protocol / protocole de transfert hypertexte

IDE : Integrated Development Environment

MVC : Model View Controller

SGBD: Système de Gestion de Base de Données

PHP : Hypertext Preprocessor

SGBD : Système de gestion de base de données

SQL : Structured Query Language

UC : Use case

UML : Unified Modeling Language

XAMPP: X (cross) Apache MariaDB Perl PHP

Tables des matières

Introduction	9
Chapitre(1): Contexte général du projet.....	10
I. Cadre général du projet :	11
I.1.Contexte et définition du problème :	11
a. Organisation de la transfusion sanguine au Maroc :	11
b. Sélection des donneurs de sang :	12
I.2.Analyse de besoin :	13
I.3.Objectif du projet :	15
I.4.Périmètre :	Erreur ! Signet non défini.
II. Conduite de projet : Méthodes agile.	15
Chapitre (2) : Conception et modélisation	16
I. Solution proposée :	17
1. Conditions d'inscription à la plateforme et de création d'un compte donneur :	17
2. Validation d'un compte donneur :	17
3. Utilisation des services (prendre des rendez-vous, proposition des check) :	18
II. Elaboration du diagramme des cas d'utilisation (UC), diagramme de séquence et diagramme d'activité :	19
1. Diagramme de cas d'utilisation :	19
a. Schéma :	19
b. Fiches descriptives et diagrammes d'activité :	21
2. Elaboration des diagrammes de séquence :	22
a. Devenir donneur :	22
b. Déclencher un cas d'urgence :	23
.....	23
3. Elaboration des diagrammes d'activité :	24
a. Devenir donneur :	24
b. Déclencher un cas d'urgence:	25
4. Elaboration de diagramme de classe :	26
Chapitre 3: Réalisation et mise en œuvre.....	27
. Présentation des outils et de l'environnement de développement :	27

1. Visual Studio Code :	27
2. PHP : Hypertext Preprocessor	28
3. Laravel :	29
4. Système de gestion de base de données :	30
5. XAMPP SERVER :	30
Conclusion	40
Références	41

Introduction

L'idée du don du sang ne date certes pas d'hier, mais le développement de ce mode **du don de sang** au Maroc est encore relativement **récent**. A l'échelle nationale, le nombre de dons collectés chaque année ne permet pas de répondre à l'augmentation de la demande, d'autant plus que la corrélation entre la consommation en produits sanguins et le nombre de dons demeure toujours déséquilibrée.

A cet égard, notre projet se propose de mettre en ligne une plate-forme web pour promouvoir ce problème de don de sang et faciliter la rencontre entre les donneurs et les demandeurs tout instaurant des mécanismes et des solutions adaptés au contexte marocain.

Le rapport s'articule autour de **trois** chapitres comme suit : un **premier** chapitre commencera par une étude de marché définissant avec précision le besoin utilisateur et l'apport de notre solution. Le **deuxième** chapitre à son tour, brosse les différentes étapes de modélisation moyennant le langage **UML** associé à une méthode conduite de **projet agile**. Quant au **troisième** chapitre, il sera consacré à la réalisation et au développement d'une application web en faisant usage aux langages de programmation web **PHP** en utilisant le framwork **Laravel** avec **bootstrap**. Et Pour la persistance de données, le choix s'est porté sur le SGBDR MySQL.

Chapitre(1):

Contexte général du projet

Dans ce chapitre, nous allons présenter le cadre général du projet et la méthodologie de conduite adoptée pour l'élaborer et la planification des tâches.

I. Cadre général du projet :

1.1.Contexte et définition du problème :

Le don de sang est un processus par lequel un donneur de sang est volontaire pour se voir prélever du sang qui sera gardé dans une banque du sang avant de transfuser une personne malade lors d'une transfusion sanguine. Que ce soit en France, en Belgique, en Tunisie, en Algérie ou au **Maroc**, le don de sang est bénévole et gratuit : autrement dit, les donneurs ne sont pas rémunérés ; contrairement à d'autres pays comme les États-Unis, l'Allemagne où le sang est considéré comme un bien marchand au sein du marché de la santé. En **Maroc**, le don du sang repose sur le monopole d'État, détenu par le Centre national de transfusion sanguine et d'hématologie (CNTSH).

La transfusion sanguine est le transfert de sang ou de constituants du sang d'un individu (donneur) à un autre (transfusé). Elle peut être vitale et les services de santé se doivent d'assurer un approvisionnement suffisant en sang sécurisé et de veiller à ce qu'il soit utilisé judicieusement.

Ce don du sang implique le respect de contraintes biologiques, immunologiques, médicales, mais aussi réglementaires et législatives encadrant cette pratique. Les principes d'identitovigilance, d'hémovigilance et de sécurité de la chaîne transfusionnelle s'appliquent notamment.

a. Organisation de la transfusion sanguine au Maroc :

Au Maroc, le don du sang est une activité médicale réglementée. Elle est organisée par le centre national de transfusion sanguine, sous tutelle du ministère de la Santé.



L'histoire de la transfusion sanguine au **Maroc** a commencé en 1943 par la création du 1er centre de transfusion sanguine (CTS) à Fès par le Médecin Commandant J. Julliard, puis à

Casablanca en 1948, et la création du centre national de transfusion sanguine (CNTS) à Rabat en 1956.

La pratique transfusionnelle est encadrée par une réglementation permettant la protection des droits et la sécurité du donneur et du receveur. La loi n° 03-94 relative au don, au prélèvement et à l'utilisation du sang humain et les décrets n° 2-94- 20 du 16 novembre 1995 et 2-96-421 du 20 novembre 1996 pris pour son application, ainsi que des arrêtés et circulaires sont les principales références réglementaires régissant la transfusion au Maroc.

Le système de TS est piloté par le centre national de transfusion sanguine et d'hématologie (CNTSH). Il comporte en outre, 16 centres de transfusion sanguine (CTS), 13 banques de sang (BS) et 30 antennes de transfusion (AT) se répartissent dans les différentes régions.

b. Sélection des donneurs de sang :

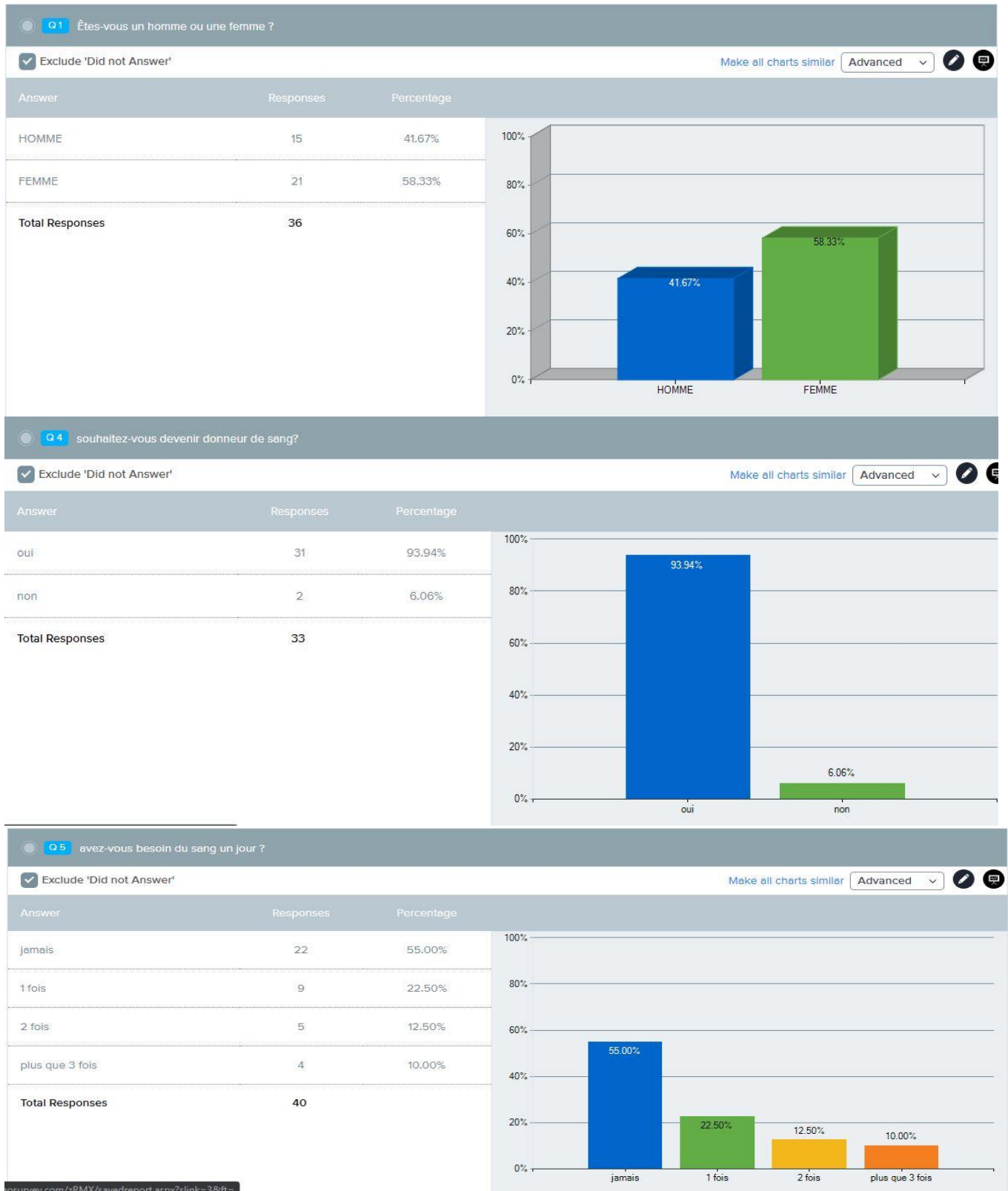
Les donneurs peuvent donner environ 500 ml de sang total jusqu'à trois fois par an ; ceci est séparé en globules rouges, plaquettes et plasma.

Le but de la sélection des donneurs est de déterminer si ceux-ci sont en bonne santé et de s'assurer que ce don de sang ne nuira pas à leur santé. En outre, la sélection a pour objet de prévenir tout risque de réactions indésirables associé à la transfusion chez le receveur notamment la transmission d'infections.

Le processus de sélection commence par le questionnaire auto-administré comportant des informations démographiques, et générales, et des renseignements sur l'état de santé, et les contacts afin de déterminer s'il réunit tous les critères associés au don de sang. La deuxième étape consiste en un entretien confidentiel avec un membre qualifié du personnel de l'établissement. Le niveau d'hémoglobine sera alors déterminé, Si tous les paramètres sont acceptables, le donneur est invité à signer un formulaire de consentement éclairé et à donner son sang.

1.2. Analyse de besoin :

Dans le cadre de notre projet nous avons mené une petite étude se basant sur un formulaire afin d'analyser les besoins, et voici les résultats de cette étude (Figure01).



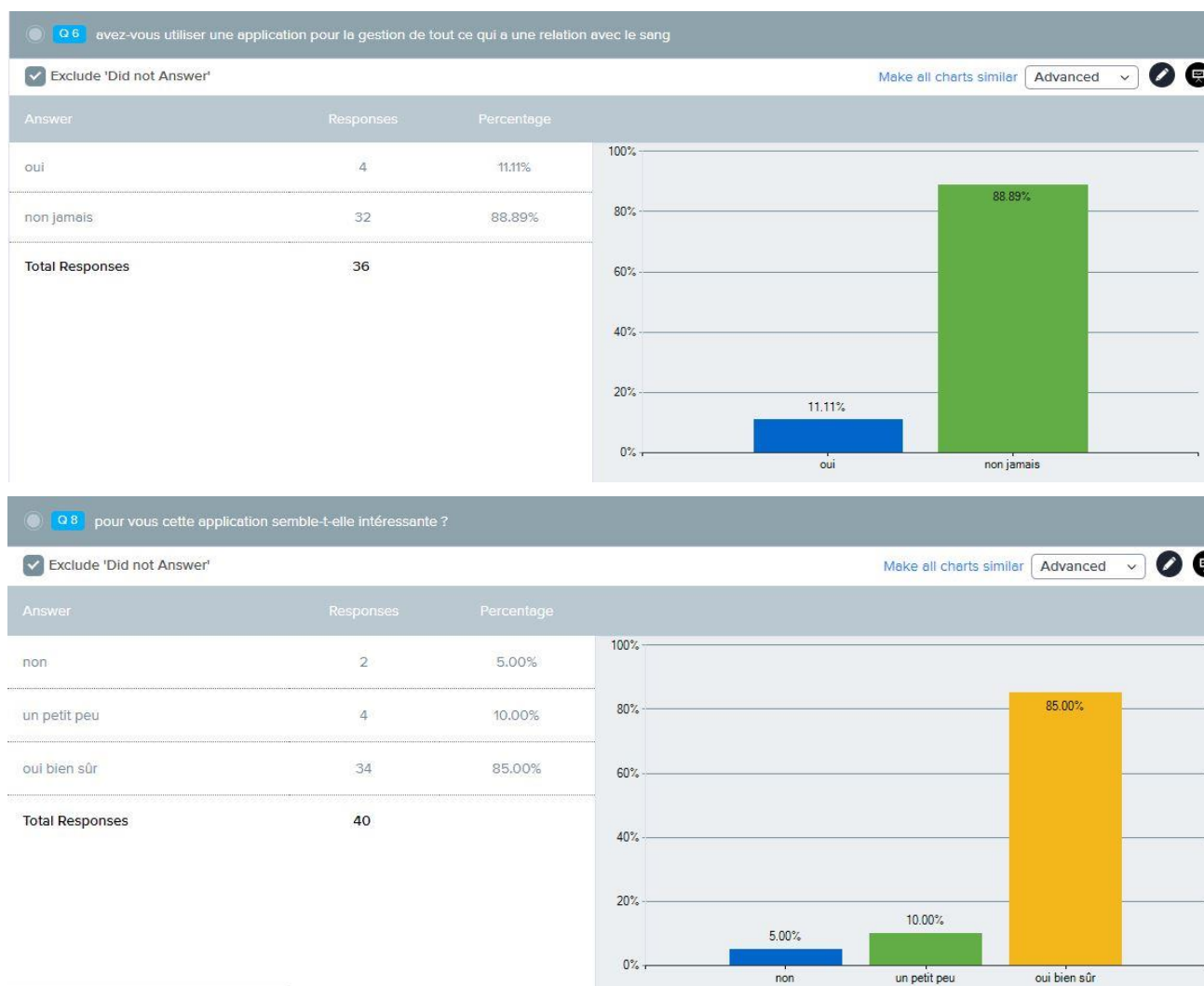


Figure01 : Les Résultats du formulaire

On remarque d'après cette étude que **93.94%** d'effective souhaitent devenir des donneurs du sang, aussi **88.89%** n'utilisent pas une application pour la gestion du don de sang. D'où l'indispensabilité de créer une application qui repend aux besoins des utilisateurs.

La dernière question est la plus importante car elle interroge l'importance de notre application et on trouve **85.00%** des gens sont intéressés par l'application.

Au cours de toute la phase de la **modélisation**, on a assuré une coopération avec un centre de don à Rabat, en effet notre groupe été guidé par les conseils de ce centre pour savoir le degré de la possibilité de réaliser telle application avec toutes les fonctionnalités proposées.

1.3.Objectif du projet :

La réalisation de notre projet a pour objectif la création et le développement d'une application adapté au contexte marocain qui permet d'assurer un contact immédiat, fiable et cohérent entre un centre du don et un donneur potentiel au Maroc, grâce à la réalisation d'une base de données pour aider à améliorer la façon de trouver un donneur d'un groupe sanguin au cas d'urgence.

II. Conduite de projet : Méthodes agile.

Avant de se lancer dans la réalisation du projet, il était primordial d'adopter une méthodologie de travail qui permettra d'analyser et de comprendre les besoins des utilisateurs, puis de définir un ensemble d'étapes concrètes et réalisables à suivre pour trouver la solution la plus fidèle à ces besoins. Le choix de l'équipe s'est porté sur l'approche agile qui se veut plus pragmatique que les méthodes traditionnelles.

La méthode Agile recommande de se fixer des objectifs à court terme. Le projet est donc divisé en plusieurs sous-projets. Une fois l'objectif atteint, on passe au suivant jusqu'à l'accomplissement de l'objectif final. Cette approche est plus flexible. Puisqu'il est impossible de tout prévoir et de tout anticiper, elle laisse la place aux imprévus et aux changements.

Avec l'approche Agile, rien n'est figé. L'équipe projet doit être capable de se remettre sans cesse en cause et de chercher continuellement à évoluer.

Chapitre (2) : Conception et modélisation

*Dans ce chapitre sera décrite la solution proposée à travers les
différents diagrammes UML.*

I. Solution proposée :

La solution proposée consiste en la mise en place d'une application mobile satisfaisant les exigences suivantes :

1. Conditions d'inscription à la plateforme et de création d'un compte donneur :

L'accès aux services avancés de notre plateforme de donneur exige des conditions d'inscriptions dont la création d'un compte, la limite d'âge et le questionnaire d'évaluations.

* Se connecter et s'inscrire comme un utilisateur normal (L'utilisateur doit s'authentifier au début pour avoir accès au contenu de l'application, pas encore ajouté dans la base des donneurs) qui peut consulter et s'informer sur l'application.

* un donneur du sang est initialement un utilisateur.

* L'utilisation de Plateforme est réservée aux personnes physiques âgées de 18 ans ou plus.

* Pour accéder à la Plateforme du donneur, l'utilisateur doit remplir une promesse de don (questionnaire d'auto-évaluation).

2. Validation d'un compte donneur :

Nous avons mis en place des fonctionnalités intégrées pour les vérifications des informations fournies par les donneurs.

*Mettre en place un système de vérification de certaines des informations que l'utilisateur fournisse lors de remplissage questionnaire.

* Mettre en place une Plateforme dédiée au centre du don sous forme d'une base de données avec la possibilité d'ajouter des nouveaux donneurs. C'est notamment le cas lorsque l'utilisateur souhaite devenir un donneur.

3. Utilisation des services (prendre des rendez-vous, proposition des check) :

Afin de rendre notre plateforme distinguée nous proposons des services supplémentaires aux donneurs tels que prendre des rendez-vous, proposition des check et un score pour le nombre des dons du sang.

* Encouragement des donneurs en laissant un avis sur leur participation dans les cas d'urgence.

*Points pour les donateurs certifiés (Votre nom et votre classement apparaissent dans la liste des donneurs actifs en fonction de votre nombre de points) pour motiver le don.

II. Elaboration du diagramme des cas d'utilisation (UC), diagramme de séquence et diagramme d'activité :

1. Diagramme de cas d'utilisation :

a. Schéma :

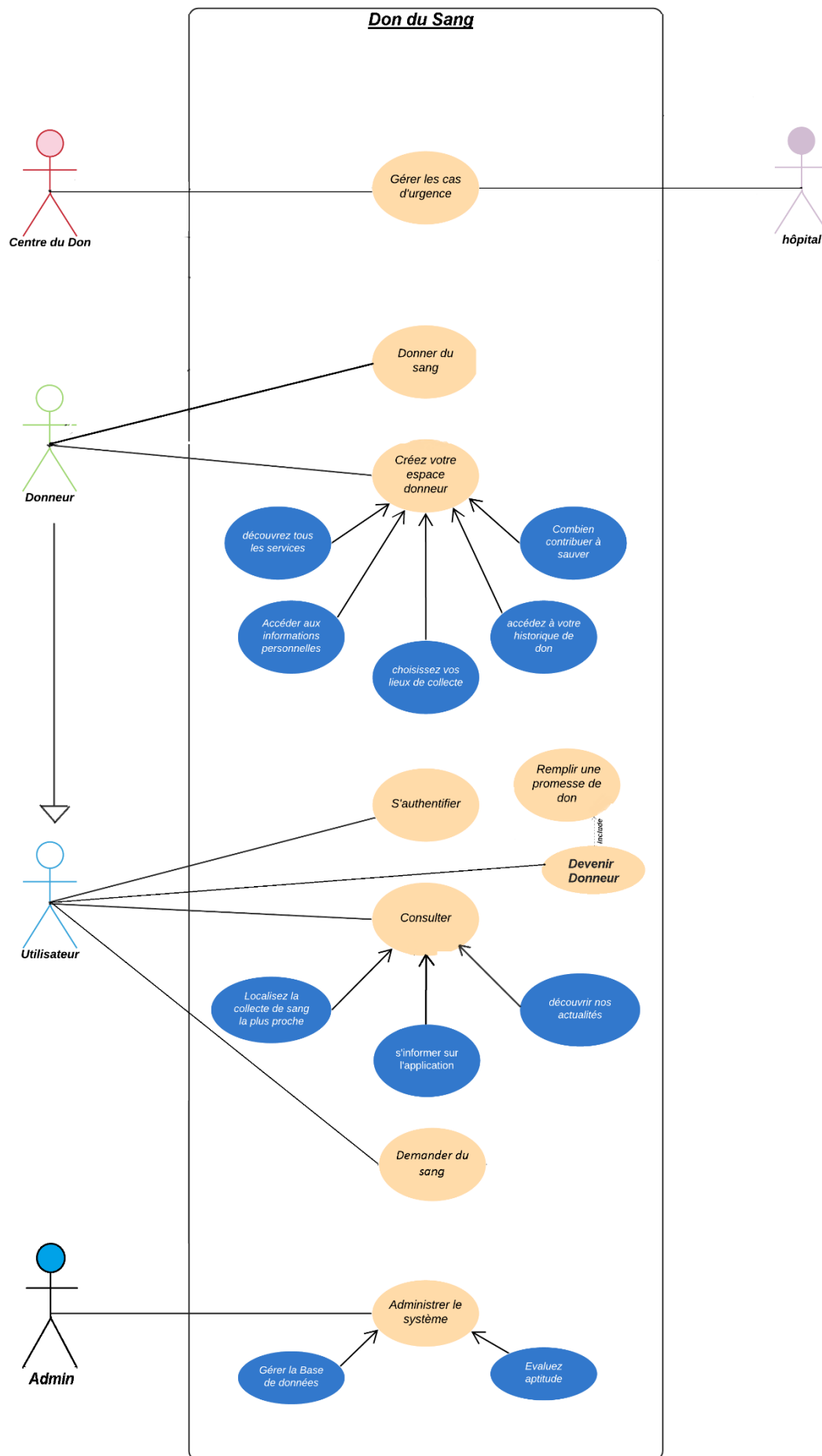


Figure02 : Diagramme de cas d'utilisation

b. Fiches descriptives et diagrammes d'activité :

Notre application contient 4 acteurs principaux plus de 2 acteurs secondaires :

- ✓ Acteurs principaux : Admin, centre du Don, Donneur, Utilisateur.
- ✓ Acteurs secondaires : Hôpital.

**Administrer* : L'admin a pour rôle d'administrer le system, c'est lui qui gère la base de données (ajouter, supprimer), notifier les utilisateurs des campagnes de don du sang et évalue les attitudes des donneurs à travers un questionnaire d'auto-évaluation pour filtrer ceux qui ne sont pas aptes pour donner du sang.

** S'authentifier* : L'utilisateur doit s'authentifier au début pour avoir accès au contenu de l'application, après il peut consulter la localisation du centre de don le plus proche, ainsi il peut s'informer sur tous ce qui a une relation avec le don et découvrir nos actualités.

**Devenir donneur* : Le donneur est un utilisateur qui veut participer dans le don du sang, donc pour que ce dernier puisse devenir un donneur, il doit remplir une promesse de don (questionnaire d'auto- évaluation).

L'utilisateur peut devenir un donneur potentiel.

**Plateforme donneur* :Au moment où l'admin accepte la demande de l'utilisateur (à travers la promesse) et l'ajoute à la base de données, où il retrouve les services proposés pour un utilisateur simple en addition des informations personnelles et un historique de sa participation et sa contribution.

** Donner du sang* :L'application va prendre en considération la date du dernier don et dans ce cas si la durée n'est pas encore complétée, le donneur va être notifier de son incapacité de donner du sang, ou bien un don dans un cas d'urgence.

** gérer les cas d'urgence* : L'application permet de gérer les cas d'urgence en ayant un contact direct entre l'hôpital où le malade est hospitalisé, et les donneurs qui souhaitent aider les gens en cas de manque du sang.

** Demander du sang (Request blood)* : L'utilisateur peut demander du sang après remplissage d'un formulaire ou le patient spécifie son type sanguin et son contact.

2. Elaboration des diagrammes de séquence :

Pour détailler plus les interactions entre les acteurs et le système, nous avons procédé à l'établissement des diagrammes de séquence pour les deux cas d'utilisation :

a. Devenir donneur :

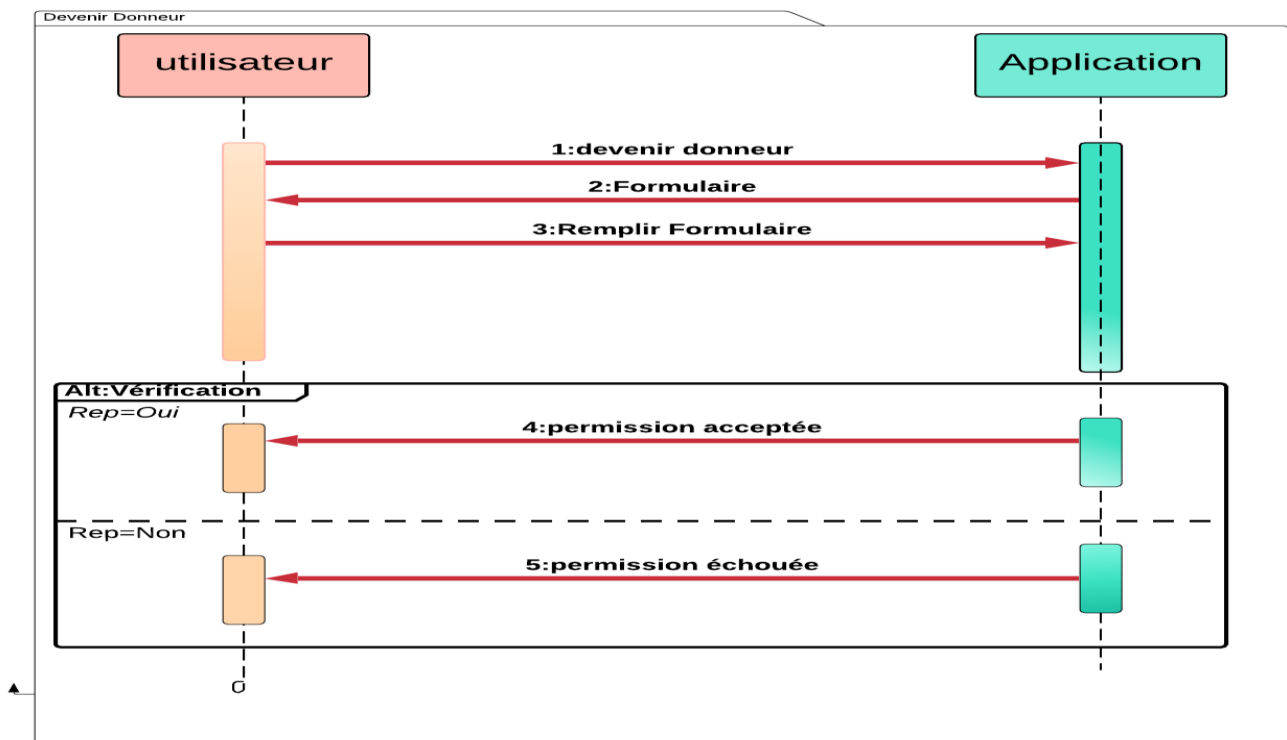


Figure03 : diagramme de séquence « devenir donneur »

Dans ce diagramme de séquence (voir Figure03), on s'intéresse au cas nommée "devenir donneur" :

Dans ce cas on a intervention de 2 acteurs :

- **L'utilisateur.**
- **L'application.**

Lors de l'authentification, si l'utilisateur choisira d'être un donneur et valide son choix. L'action d'appuyer sur « valider » est traduite par un message synchrone, c'est à dire, l'utilisateur doit attendre un laps de temps pour recevoir un message qui est sous forme d'un formulaire contient un petit questionnaire pour voir s'il est apte à donner. Les questions peuvent impliquer état actuel de santé (Êtes-vous enceinte ou avez-vous accouché depuis moins de 6 mois ?, Avez-vous été testée positif pour le VIH (sida), ou VHB (hépatite B) ou VHC (hépatite C) ou la syphilis ?, Avez-vous

déjà pris des drogues illicites par voie intraveineuse?...), après validation le système vérifie la compatibilité des informations avec les conditions exigées ce qui fait la naissance d'un alternatif traitant deux cas :

Premier cas : si le système ne valide pas les données du formulaire (à cause de l'absence d'un ou plusieurs critères principaux) ; dans ce cas il renvoie à l'utilisateur « Permission échouée ».

Deuxième cas : si l'application valide ces informations alors l'utilisateur deviendra un donneur.

b. Déclencher un cas d'urgence :

L'objectif principal de cette application est « le don du sang en situation d'urgence ». C'est dans ce cadre qu'on traite le scénario d'utilisation intitulé « déclencher un cas d'urgence ».

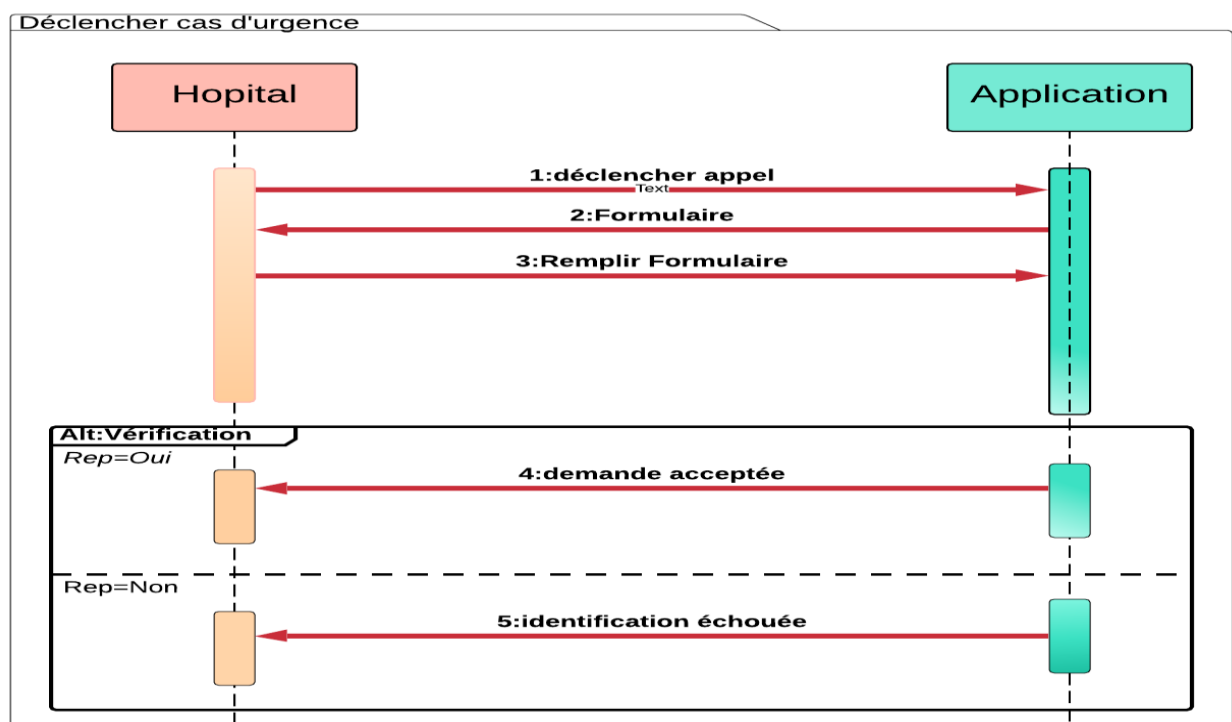


Figure04 : diagramme de séquence « déclencher cas d'urgence »

Le cas étudié dans ce diagramme de séquence (voir Figure04) nécessite l'intervention des deux acteurs :

- *Hôpital*
- *Système*

L'hôpital lance un appel de don du sang. Le système lui renvoie un formulaire qui exige une précision de son identifiant et le groupe sanguin demandé. En retour l'hôpital remplit le formulaire et attendre une réponse de la part du système. Le système vérifie cette identification. Cette phase de vérification a donné naissance à un fragment alternatif. Si l'identifiant est compatible alors le système ajoute la demande comme une actualité dans le site et renvoie les contacts des donneurs qui ont le groupe sanguin demandé à l'hôpital dans le cas de disponibilité. Sinon l'identification sera échouée.

3. Elaboration des diagrammes d'activité :

a. Devenir donneur :

Voici le diagramme d'activité qui détaille le cas en question.

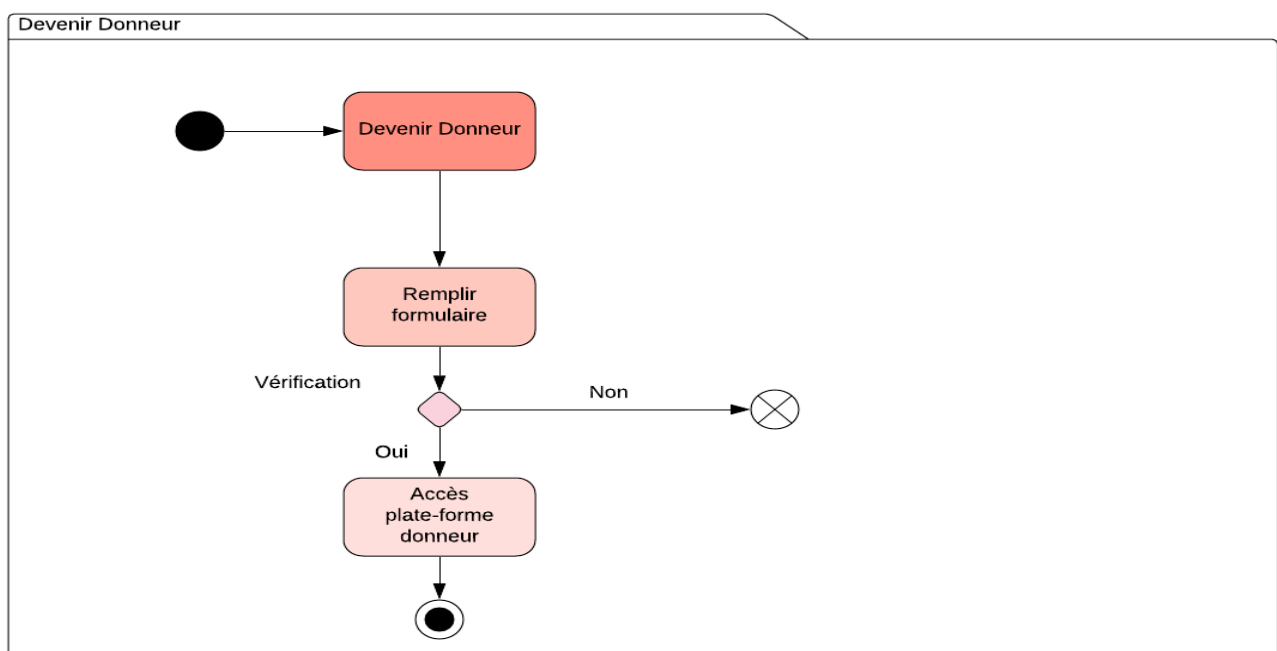


Figure05 : Diagramme d'activité « devenir donneur »

Lorsque l'utilisateur accède au site, puis choisi de devenir donneur, un formulaire à remplir s'affiche, ensuite, le système vérifie si les informations du formulaire sont en concordance avec les conditions imposées. Le système décide si l'utilisateur est capable d'être un donneur ou pas, si oui : l'accepter et lui accéder à une nouvelle plateforme, celle qui concerne un donneur, sinon fin d'action.

b. Déclencher un cas d'urgence:

Voici la formalisation graphique des actions traitées dans ce cas d'utilisation.

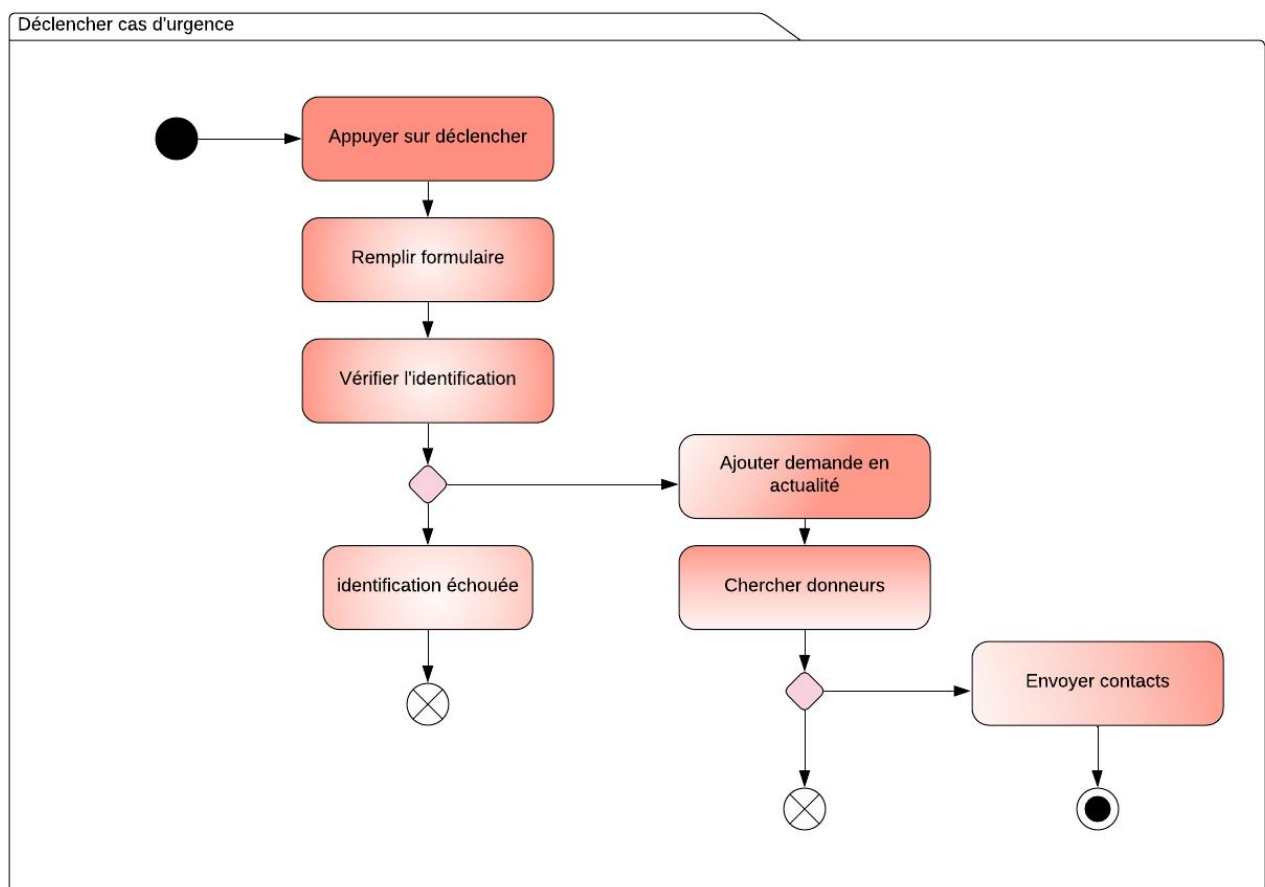


Figure06 : Diagramme d'activité « déclencher cas d'urgence »

L'hôpital déclenche une alerte de don du sang, il remplit un formulaire (identifiant, le groupe sanguin) par la suite pour informer le système de la demande exigée. Ce dernier passe à la phase de vérification qui consiste dans un premier lieu à vérifier la compatibilité de l'identifiant. S'il n'est pas compatible l'alerte sera rejetée. Sinon le système accepte la demande et l'ajoute aux actualités de

site. Ensuite, le système vérifie la base de données et renvoie à l'hôpital les contacts des donneurs qui sont capables de répondre à cet alerte et qui peuvent se déplacer vers l'hôpital concerné.

4. Elaboration de diagramme de classe :

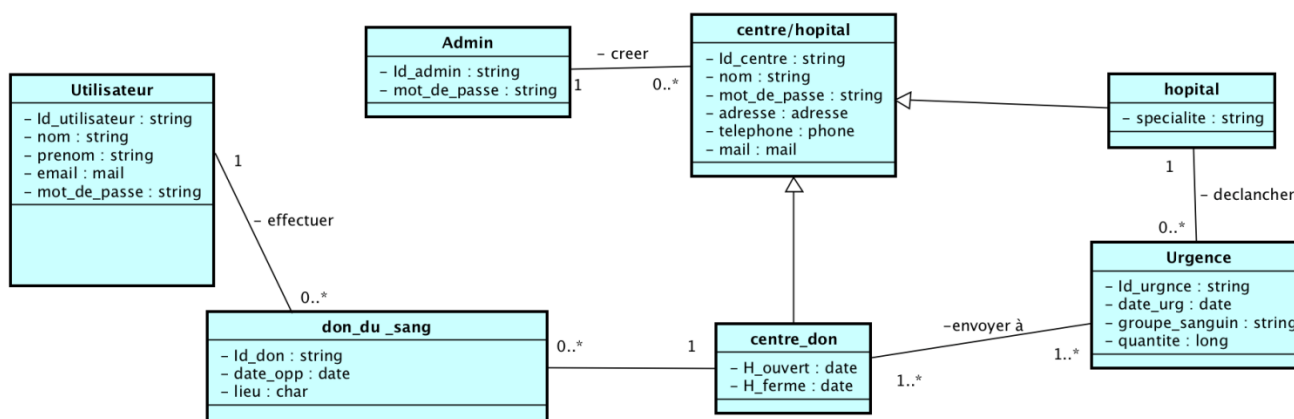


Figure07 : Diagramme de classes

Utilisateur : toute personne qui se connecte à l'application pour avoir des données générales liées au don du sang ou bien qui cherche un hôpital, un centre de don.

Promesse_don : c'est un formulaire d'auto-évaluation dédié pour les utilisateurs qui veulent devenir donneur, il permet de filtrer les utilisateurs qui ne sont pas aptes pour donner du sang.

Centre_don : c'est lui qui est en charge de la réception des demandes provenant d'un cas d'urgence, aussi il permet de localiser les différents centres de don et les contacter directement.

Admin : une classe permettant la gestion des comptes de centre de don, et des hôpitaux ayant connectés à l'application.

hôpital/centre : la classe qui permet de décrire clairement les éléments (attributs) communs entre le centre de don et l'hôpital.

Hôpital : au plus des éléments hérités, la classe hôpital a comme attribut la spécialité, cette classe permet d'identifier un hôpital qui est un acteur principal dans notre application.

Urgence : cette classe a pour rôle principal de décrire un cas d'urgence déclenché par un hôpital.

Chapitre 3:

Réalisation et mise en œuvre

Dans ce chapitre nous commencerons par la description de l'environnement de développement ainsi que les différents outils utilisés. Ensuite, nous présenterons les différents aspects de fonctionnement de notre application.

I. Présentation des outils et de l'environnement de développement :

Dans ce projet nous avons choisi comme IDE le logiciel de développement **VISUAL STUDIO CODE**, comme langage de développement du back-end le langage Orienté Objet PHP7 avec le **Framework LARAVEL**, pour le middle front le système de gestion de bases de données Mysql, et pour le front end le langage **WEB HTML5, CSS3** avec le Framework **BOOTSTRAP** qui est une collection d'outils utiles à la création du design de sites et d'applications web.

1. Visual Studio Code :

Visual Studio Code est un éditeur de code développé par Microsoft pour Windows, Linux et OS X. Il est gratuit et open source, et inclut plusieurs fonctionnalités comme la coloration syntaxique, la complétion de code intelligente, les extraits de code et la refactorisation de code. Il permet aussi à l'utilisateur de personnaliser le thème de l'éditeur, les raccourcis clavier et les préférences.

Visual Studio Code supporte divers langages de programmation tels que : le C++, le VB.NET, C#, le F#, le Python, ou encore le Ruby. Il propose également le support du XML/XSLT, du

HTML/XHTML, du JavaScript ou encore du CSS. Nous avons donc décidé que cet outil était plus que nécessaire si l'on envisage le développement de l'application en Laravel. En effet, celui-ci permet d'exploiter tout le potentiel des technologies de Google.

Pour la partie visuelle des pages, nous avons choisi d'appuyer les différents contrôles de saisie car étant du côté client, la vérification des données saisies est plus rapide que s'il s'agissait d'une vérification du côté serveur.

2. PHP : Hypertext Preprocessor

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.

PHP est un langage de script utilisé le plus souvent côté serveur : dans cette architecture, le serveur interprète le code PHP des pages web demandées et génère du code (HTML, XHTML, CSS par exemple) et des données (JPEG, GIF, PNG par exemple) pouvant être interprétés et rendus par un Navigateur web.

Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour le Web. PHP est le plus souvent couplé à un serveur Apache bien qu'il puisse être installé sur la plupart des serveurs HTTP. Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure.

C'est un langage peu typé et souple et donc facile à apprendre par un débutant mais, de ce fait, des failles de sécurité peuvent rapidement apparaître dans les applications. Pragmatique, PHP ne s'encombre pas de théorie et a tendance à choisir le chemin le plus direct.

Son utilisation commence avec le traitement des formulaires puis par l'accès aux bases de données. L'accès aux bases de données est aisé une fois l'installation des modules correspondants effectuée sur le serveur. La force la plus évidente de ce langage est qu'il a permis au fil du temps la résolution aisée de problèmes autrefois compliqués et est devenu par conséquent un composant incontournable des offres d'hébergements.

Libre, gratuit, simple d'utilisation et d'installation, ce langage nécessite comme tout langage de programmation une bonne compréhension des principales fonctions usuelles ainsi qu'une connaissance aigüe des problèmes de sécurité liés à ce langage.

3. Laravel :

Laravel, créé par Taylor Otwell, initie une nouvelle façon de concevoir un framework en utilisant ce qui existe de mieux pour chaque fonctionnalité. Par exemple toute application web a besoin d'un système qui gère les requêtes HTTP. Plutôt que de réinventer quelque chose, le concepteur de Laravel a tout simplement utilisé celui de Symfony en l'étendant pour créer un système de routage efficace. En quelque sorte Otwell a fait son marché parmi toutes les bibliothèques disponibles. C'est aussi de nombreux composants originaux et surtout une orchestration de tout ça.

Vous allez trouver dans Laravel :

- un système de routage perfectionné (RESTful et ressources),
- un système d'authentification pour les connexions,
- un système de validation,
- un système de pagination,
- un système de migration pour les bases de données,
- un système d'événements,
- un système d'autorisations,
- une gestion des sessions...

Pour conclure, on peut dire que ce Framework est un outil très performant pour la création d'applications web. C'est **Un framework** qui est fait pour gagner du temps et donne l'assurance de disposer de composants bien codés et fiables.

Laravel est un framework novateur, complet, qui utilise les possibilités les plus récentes de PHP et qui est impeccablement codé et organisé.

4. Système de gestion de base de données :

La solution développée utilise comme système de gestion de base de données (SGBD) **MYSQL** qui est un serveur de bases de données relationnelles, 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.

C'est un logiciel libre, open source, développé sous double licence selon qu'il est distribué avec un produit libre ou avec un produit propriétaire. Un logiciel qui intègre du code MySQL ou intègre MySQL lors de son installation devra donc être libre ou acquérir une licence payante. Pour notre projet on a utilisé le logiciel **HeidiSQL**.

HeidiSQL est un outil d'administration de base de données possédant un éditeur SQL et un constructeur de requête. Il a été développé et optimisé pour être utilisé avec le SGBD relationnel MySQL disponible commercialement ou gratuitement.

5. XAMPP SERVER :

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 (**X** (cross) **A**pache **M**ariaDB **P**erl **P**HP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.

Officiellement, XAMPP permet de configurer un serveur de test local avant la mise en œuvre d'un site internet, et son usage n'est pas recommandé pour un serveur dit de production.

II. Description de la solution

1. Architecture de l'application web :

Vue qu'on a utilisé le framework laravel de php notre plateforme sera soumise à l'architecture MVC de celui-ci qui se présente comme suit :

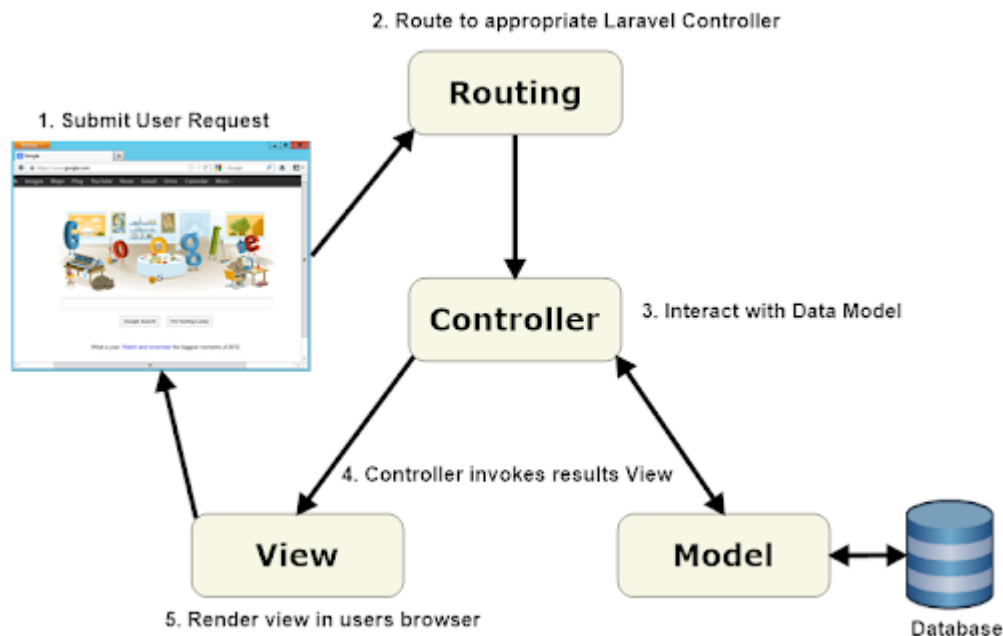


Figure08 : Architecture d'une application web

2. Une application Laravel :

a. C'est quoi l'architecture MVC ?

Laravel est l'un des frameworks web PHP les plus populaires. Il est gratuit, et son code source est ouvert et basé sur le modèle architectural MVC.

MVC est le sigle de « Model View Controller », qui représente l'architecture que les développeurs adoptent lors de la création d'applications. Avec l'architecture MVC, nous avons défini et modélisé les flux de données de notre application.

- **Le modèle** gère les comportements fondamentaux et les données de l'application.
- **La vue** fournit effectivement l'élément d'interface utilisateur de l'application.

- **Le contrôleur** reçoit les entrées de l'utilisateur et passe des appels aux objets du modèle et à la vue pour effectuer les actions appropriées.

b. Accès à la base de données

Laravel fournit une suite d'outils de pointe pour interagir avec les bases de données. Les migrations de bases de données nous a permet de concevoir et modifier facilement une base de données indépendamment de la plate-forme. Les migrations peuvent ensuite être exécutées avec n'importe quel type de base de données prise en charge par Laravel (MySQL).

c. Routage :

Laravel nous a permet d'organiser toutes les URL de votre site web via des routeurs. Si on souhaite modifier le lien d'un site web ou d'une API, nous devons le faire en un seul fichier, et notre site web fonctionnera comme prévu.

d) Le cycle de la requête :

Lorsque la requête atteint le fichier **public/index.php** l'application Laravel est créée et configurée et l'environnement est détecté. Ensuite le fichier **routes.php** est chargé. Voici l'emplacement du fichier des routes :

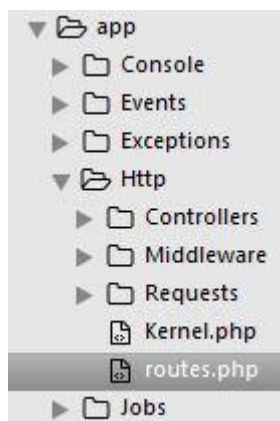


Figure09 : Fichier de Routes

C'est avec ce fichier que la requête va être analysée et dirigée.

Laravel propose plusieurs helpers qui simplifient la syntaxe. Il y a par exemple **view** pour la classe **View** comme on l'a vu dans le code ci-dessus. Laravel est équipé de nombreux helpers comme celui-ci qui facilitent et accélèrent le codage.

e) Les vues :

Dans l'application nous retournons la réponse en passant par la vue qui est un simple fichier avec du code HTML. Il faut enregistrer cette vue dans le dossier **resources/views** avec l'extension **php**.

f) Les contrôleurs :

La tâche d'un contrôleur est de réceptionner une requête (qui a déjà été triée par une route) et de définir la réponse appropriée, rien de moins et rien de plus.

Voici une illustration du processus :

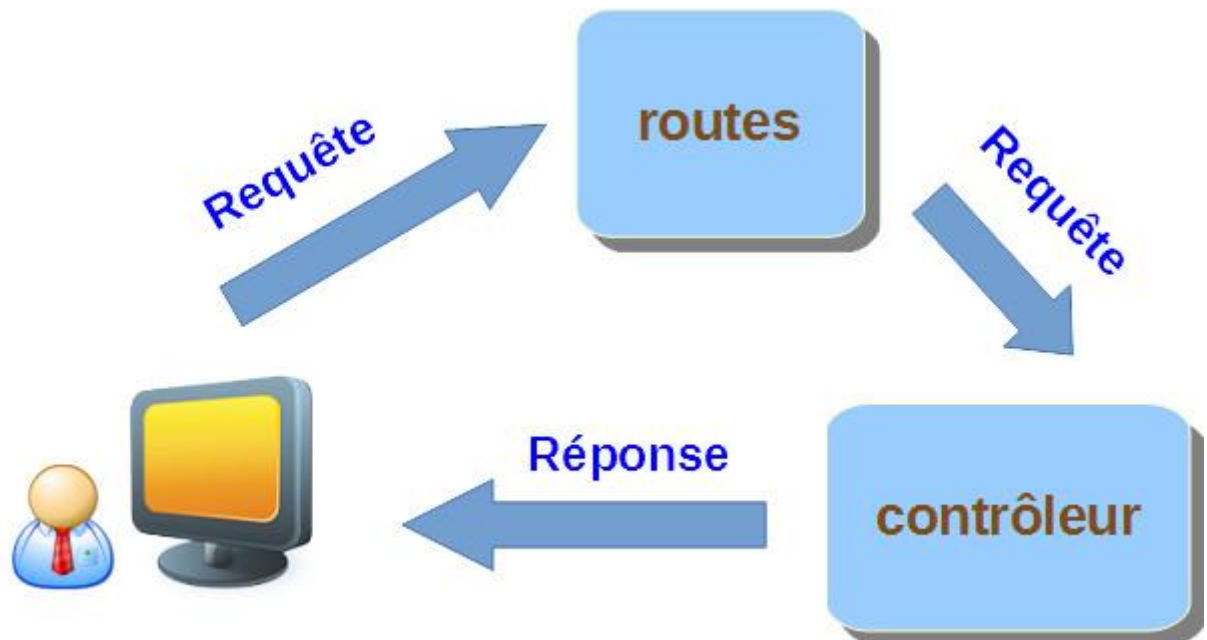


Figure10 : Traitement de la requête par un contrôleur

g) Les migrations :

Une migration permet de créer et de mettre à jour un schéma de base de données. Autrement dit, nous pouvons créer des tables, des colonnes dans ces tables, en supprimer, créer des index... Tout ce qui concerne la maintenance de vos tables peut être pris en charge par cet outil.

La configuration de la base

Nous devons dans un premier temps avoir une base de données. Laravel permet de gérer les bases de type MySQL, avec le logiciel HeidiSQL .

Il faut indiquer où se trouve votre base, son nom, le nom de l'utilisateur, le mot de passe dans le fichier de configuration **.env** .

En premier lieu, il faut installer la migration en utilisant la commande suivante :

php artisan migrate:install

La deuxième étape consiste à créer la migration pour nos tables :

php artisan make:migrations

3. La solution développée:

La page d'accueil de notre application contient les informations sur notre solution. Ainsi que des différentes possibilités de connexion, inscription, contact et accueil.

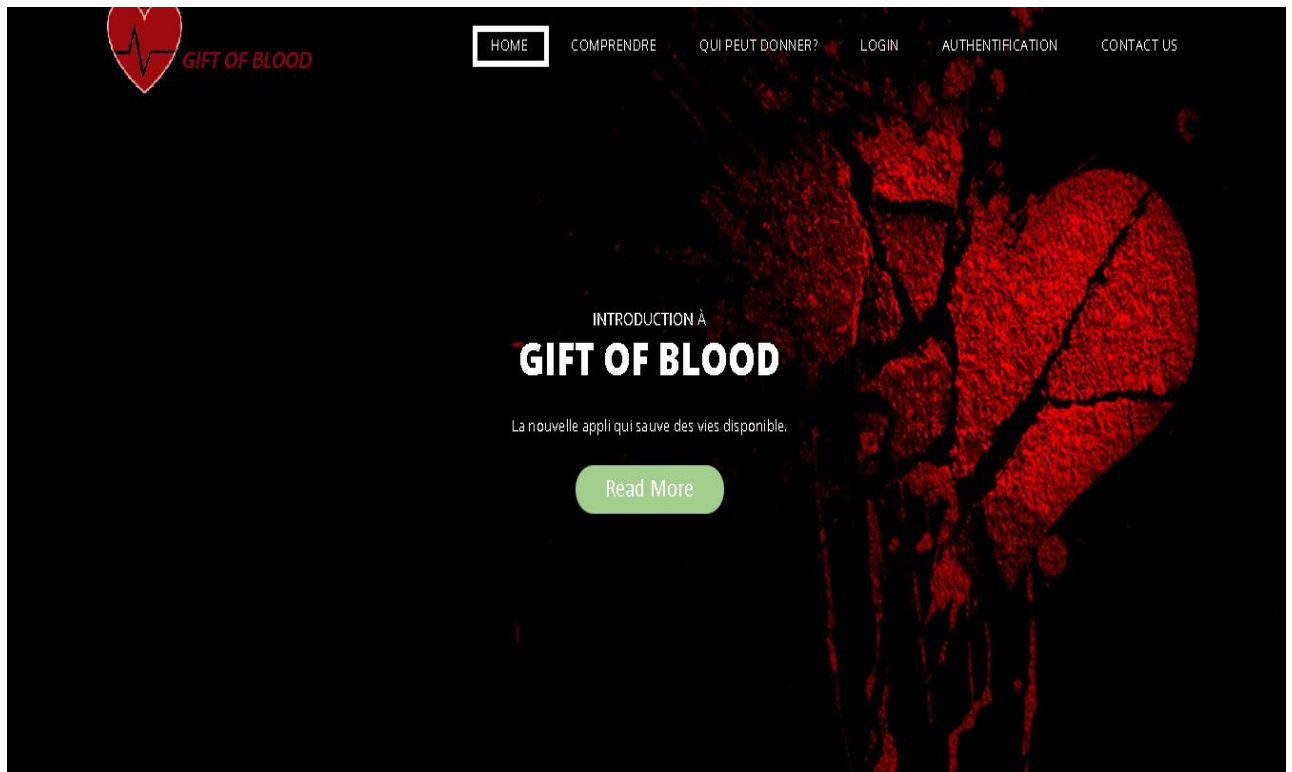


Figure11 : La page d'accueil

Cette page d'accueil contient aussi une présentation des avantages de la solution.

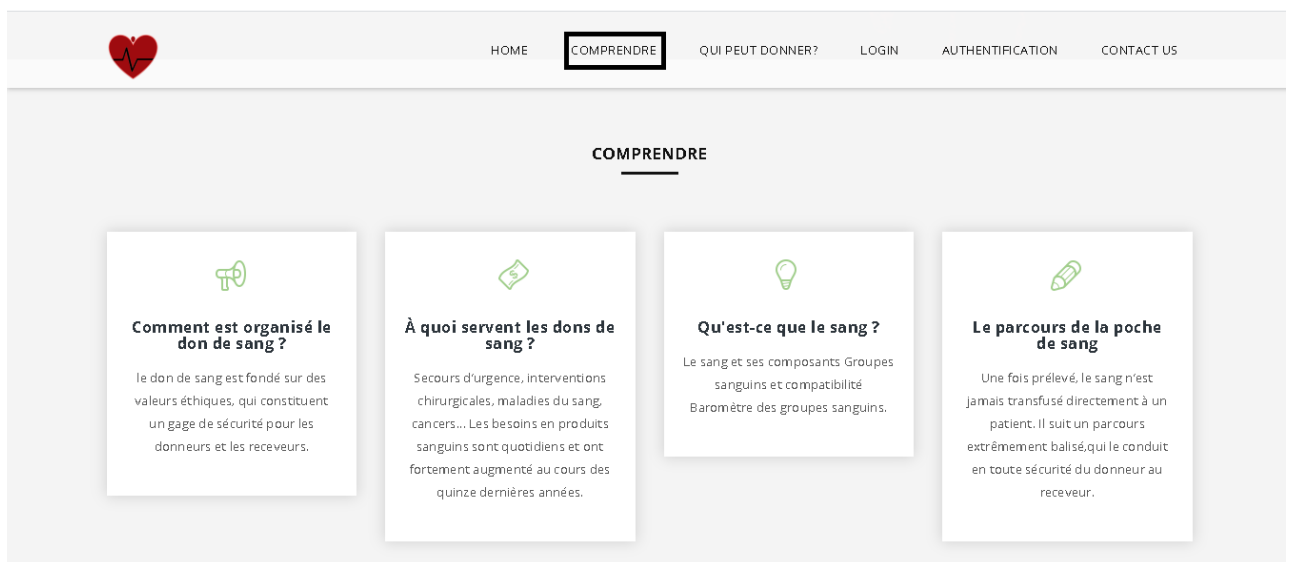


Figure12 : Présentation des avantages de la solution

On trouve des informations, Indicateurs sur la page « qui peut donner ? »

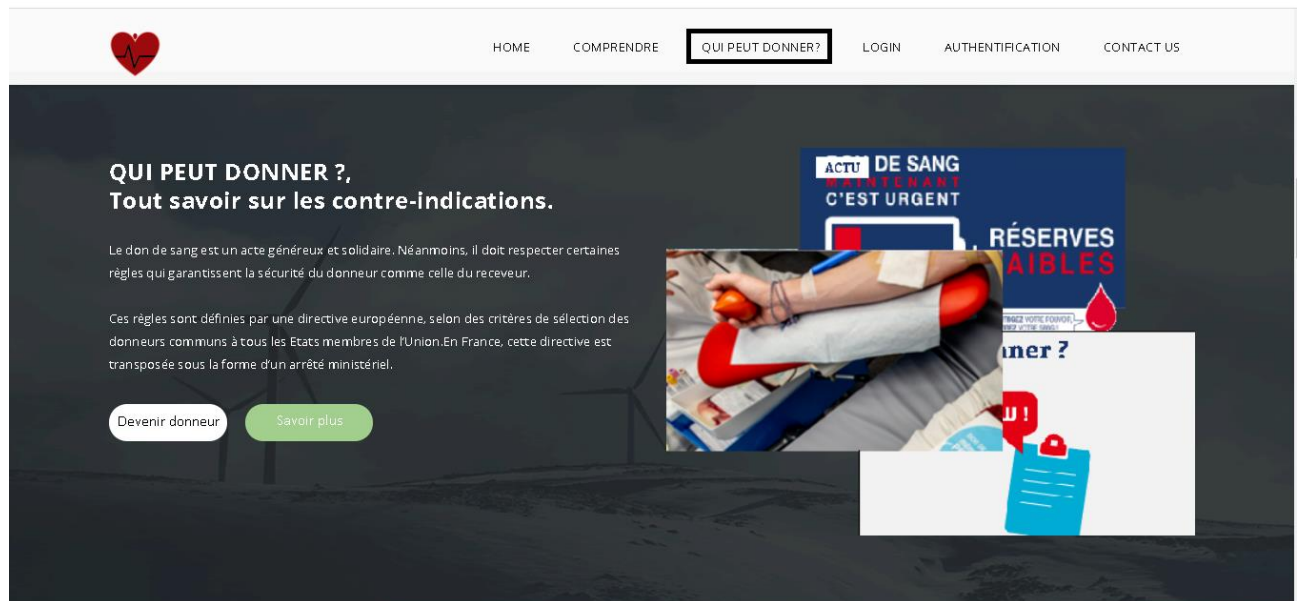


Figure13 : Présentation de la page ‘qui peut donner ?’

Sur cette page, il existe aussi un bouton pour Devenir donneur le clic sur Devenir donneur ! Dirige l'utilisateur vers la page d'inscription.

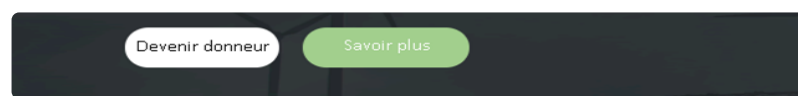


Figure14 : devenir donneur

Avant toute opération de demande de don, l'utilisateur est amené à se connecter à la plateforme en saisissant son email et son mot de passe.

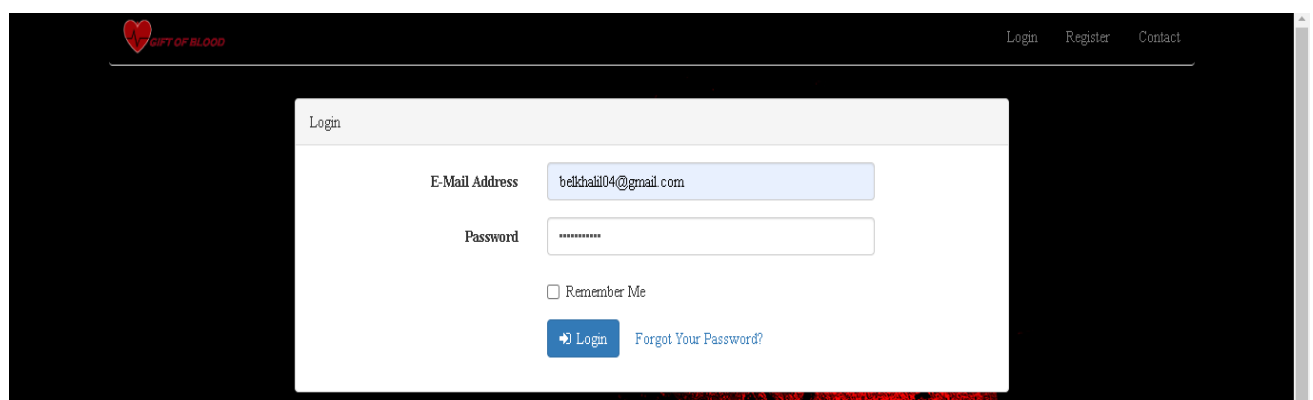


Figure15 : La section ‘login’

Si l'utilisateur n'est pas inscrit dans la plateforme, il doit s'inscrire en saisissant les informations en dessous. Également, Si l'utilisateur veut être un donneur, il doit cocher la case yes pour la question 'devenir donneur ?'.

The screenshot shows the 'Authentification' registration form on a website with a dark blue header. The header includes a red heart logo with 'GIFT OF BLOOD' text, and links for 'Login', 'Register' (highlighted with a red box), and 'Contact'. The form itself is white and titled 'Authentification'. It contains the following fields: 'Nom' (text input), 'Age' (text input), 'Sexe' (dropdown menu with 'Homme' selected), 'Poids' (text input), 'Groupe sanguine' (dropdown menu with 'AB +' selected), 'Téléphone' (text input), 'Adresse' (text input with a small icon at the bottom right), 'Ville' (text input), and 'Région' (text input).

Figure16 : Le Formulaire d'inscription

Après qu'un utilisateur soit connecté à notre plateforme, l'utilisateur a la possibilité de déposer une demande de don de sang.

The screenshot shows the 'Demander du sang' page on the same website. The dark blue header now includes a navigation menu with links: 'Home', 'Donneurs', 'Hopitaux', 'Centres du don', 'Demander du sang' (highlighted with a red box), 'Cas urgent', and 'KHALIL BALGA' with a dropdown arrow. The main content area is white and titled 'Demander du sang'. It contains the following fields: 'Nom de Patient' (text input), 'AB +' (dropdown menu), 'Pays' (text input), 'Région' (text input), 'Ville' (text input), 'Nom d'Hôpital & Adresse' (text input with a small icon at the bottom right), 'Contacte de Personne' (text input), 'Contacte Email' (text input), 'Contact téléphone' (text input), and 'jj/mm/aaaa' (text input with a calendar icon). A 'Demander' button is located at the bottom left of the form.

Figure17 : La page de demande de sang

Une fois la demande est faite, une autre page affiche la liste des donneurs disponibles correspondants à la même adresse demandée.

The screenshot shows the 'Donneurs' page with a navigation bar and search filters. Below the filters, three donor profiles are displayed:

- KHALIL BALGA**: Groupe sanguine : AB +ve, Ville : Nador, Région : Oriental, Pays : Maroc, Poids : 170 kgs, Sexe : M, Age : 22, Dernier don le : December 03, 2017 09:12 AM, Media Sociale : —
- OUMAYMA AK**: Groupe sanguine : O +ve, Ville : berkane, Région : Oriental, Pays : Maroc, Poids : 165 kgs, Sexe : F, Age : 21, Dernier don le : June 10, 2020 01:06 AM, Media Sociale : —
- Issam**: Groupe sanguine : O -ve, Ville : fes, Région : fes, Pays : Maroc, Poids : 170 kgs, Sexe : M, Age : 22, Dernier don le : January 01, 1970 12:01 AM, Media Sociale : f

Figure18: La liste des donneurs

Pour faire une recherche rapide ou bien avoir des informations sur le don de sang : la liste de compatibilité entre les groupes sanguines... l'utilisateur doit accéder à la page 'home'.

The screenshot shows the 'Home' page with a search bar and a table of red blood cell compatibility. The table is titled 'Tableau de compatibilité des globules rouges' and has columns for Recipient and Donor.

Recipient	Donor							
	O-	O+	A-	A+	B-	B+	AB-	AB+
O-	✓							
O+	✓	✓						
A-	✓		✓					
A+	✓	✓	✓	✓				
B-	✓				✓			
B+	✓	✓			✓	✓		
AB-	✓		✓		✓		✓	

Figure19 : La page 'Homme'

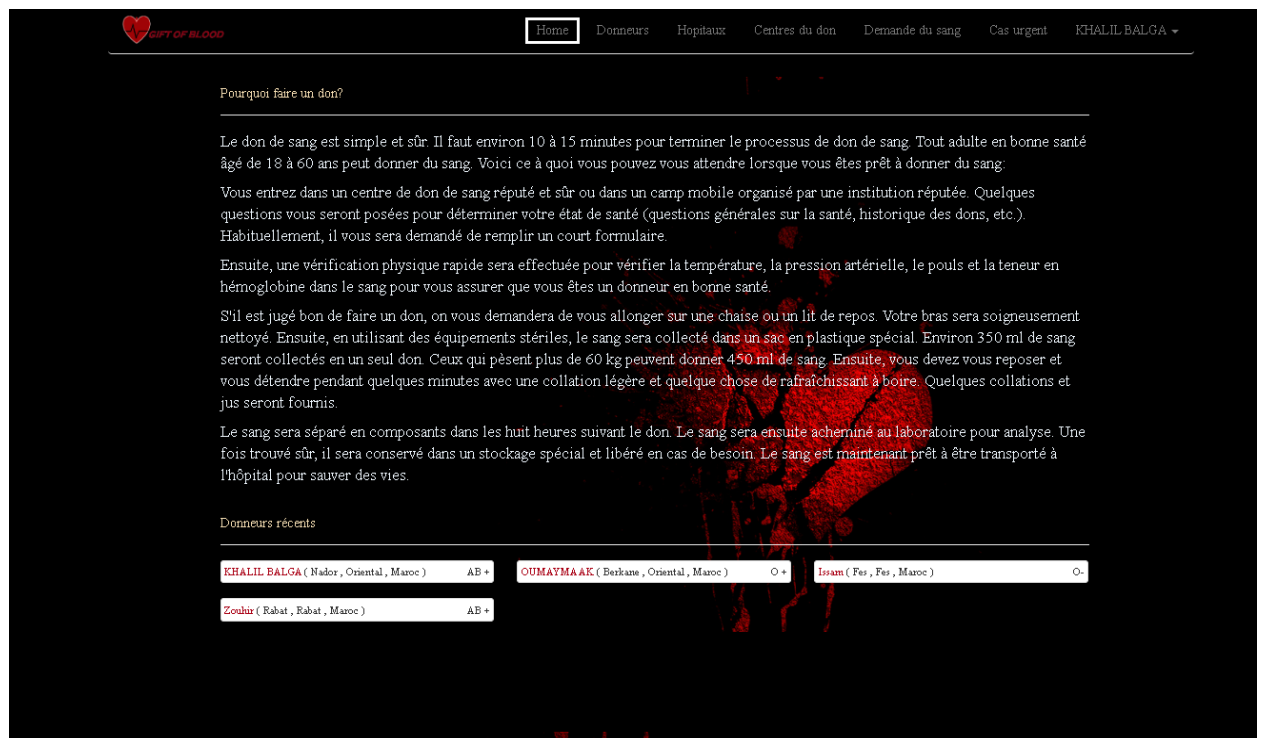


Figure20 : La page 'Home'2

Dans la page 'centres de don' on trouve la liste des centres de don connectés à notre application.

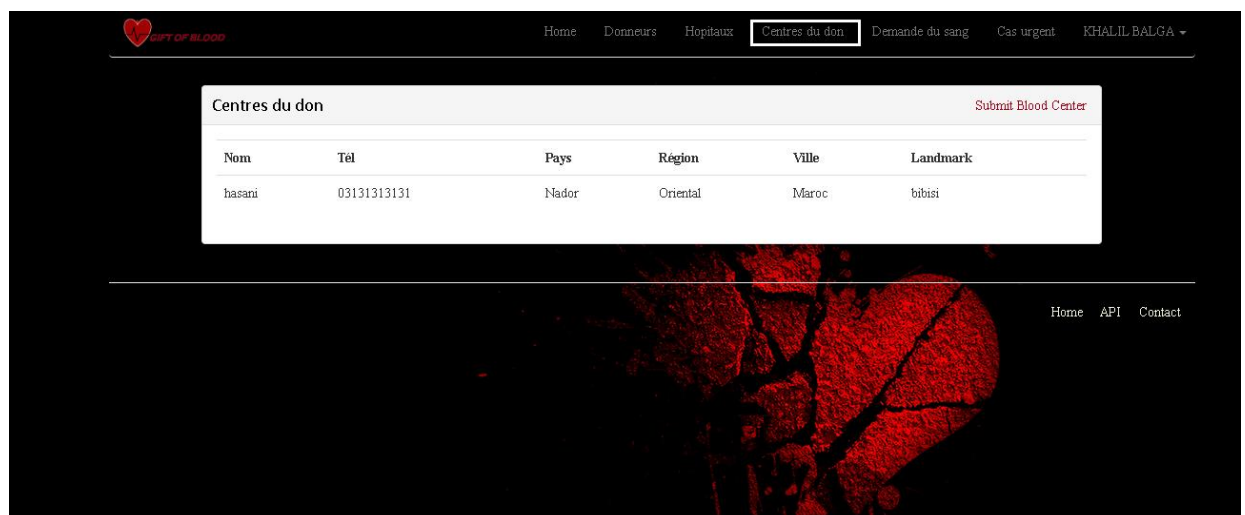


Figure21 : La liste des centres de don

Conclusion

En arrivant à la conclusion, nous avons abordé tous les chapitres prévus pour notre projet en effet nous avons défini le contexte général du projet, nous avons défini une conception complète à notre jugement du problème. Une conception qui comporte et décrit tous les besoins de l'application afin d'arriver à la partie réalisation et mise en œuvre qui fut notre objectif dès le début.

Au terme de notre projet de don de sang, nous avons pu suivre le processus nécessaire à la conception d'une application Web et les différentes étapes qui constituent les piliers de base de cette réalisation tout en facilitant les fonctionnalités de recherche et de navigation au sein d'une application maîtrisée au lieu d'aller chercher sur des pages Facebook ou autre.

Nous avons pu également développer nos capacités et nos compétences de conduite d'un projet informatique et les mettre en œuvre dans un environnement compréhensible par les utilisateurs, cela nous a permis également de se familiariser avec la digitalisation des projets afin de les exploiter dans nos futures réalisations.

Références

Site WEB :

<https://www.google.com/search?q=php+7&oq=php+7&aqs=chrome..69i57j69i60.1670j0j1&sourceid=chrome&ie=UTF-8>

<https://fr.wikipedia.org/wiki/PHP>

<https://fr.wikipedia.org/wiki/MySQL>

<https://fr.wikipedia.org/wiki/HeidiSQL>

<https://fr.wikipedia.org/wiki/XAMPP>

<https://getbootstrap.com/>

<https://www.sante-sur-le-net.com/donneurs-don-sang/>

https://www.sciencesetavenir.fr/fondamental/biologie-cellulaire/la-difference-entre-un-bon-et-un-mauvais-donneur-de-sang-ce-sont-ses-globules-rouges_141168

[https://sehati.gov.ma/article/don de sang](https://sehati.gov.ma/article/don_de_sang)

<http://sante.lefigaro.fr/sante/traitement/don-sang/sang-quest-ce-que-cest>

<http://albayane.press.ma/don-sang-affaire-de.html>

<https://www.who.int/features/qa/61/fr/>

<https://dondesang.efs.sante.fr/comprendre-quoi-servent-les-dons-de-sang/les-besoins-au-quotidien>

<https://www.jeancoutu.com/sante/conseils-sante/don-de-sang/>