



Faculty of Mathematical, Physical and
Natural Sciences of Tunis

Faculté des Sciences Mathématiques,
Physiques et Naturelles de Tunis

كلية العلوم للرياضيات
والفيزياء والطبيعية بتونس

RAPPORT DE PROJET DE FIN D'ANNÉE

Application de don du sang (Donate Blood)

Réalisé par : **Motez Lassoued**
Ahmed Hammami

Encadré par : **Heithem Abbes**

Année Universitaire 2021 - 2022

Remerciements

Il était agréable de nous acquitter d'une dette de reconnaissance envers tous ceux, dont la contribution au cours de ce projet, a favorisé son aboutissement.

Ainsi, nous tenons vivement à remercier notre encadrant Mr Heithem Abbes qui nous a aidé et nous a orienté le long de notre projet par son encadrement précieux.

Que le corps professoral et administratif de l'FST trouve ici nos vifs remerciements. Nous remercions toute personne qui a contribué de près ou de loin à l'élaboration de ce rapport

Table de matière

Introduction Générale	5
Chapitre I : Contexte Général	6
I.1 Introduction.....	7
I.2 Présentation du projet	7
Chapitre II : Spécification des besoins.....	7
II.1 Introduction	7
II.2 Identification des acteurs.....	7
II.3 Identification des besoins fonctionnels	7
II.3.1 Besoins fonctionnels.....	7
II.3.1 Besoins non-fonctionnels	8
II.4 Spécification de fonctionnalités de l'application	9
II.5 Identification des scenarios	10
II.5.1 Le cas d'utilisation <i>s'authentifier</i>	10
II.5.2 Le cas d'utilisation <i>Consulter la liste de donateurs</i>	10
II.5.3 Le cas d'utilisation <i>Consulter la liste des demandes de sang</i>	11
II.5.4 Le cas d'utilisation <i>Gérer le compte</i>	11
II.5.5 Le cas d'utilisation <i>Créer une demande de sang</i>	12
II.5.6 Le cas d'utilisation <i>Gérer les requêtes</i>	12
II.5.7 Le cas d'utilisation <i>Gérer les blogs</i>	13
Chapitre III : Architecture et Conception.....	14
III.1 Introduction	14
III.2 Conception générale.....	14
III.2.1 Architecture	14
III.3 Conception détaillée.....	15
III.3.1 Diagramme de classes	15
III.3.1 Diagrammes de Séquences.....	16
Chapitre IV : Réalisation	18

IV.1 Introduction	18
IV.2 Environnement matériel	18
IV.3 Environnement logiciel	19
IV.3.1 Architecture logiciel	19
IV.3.2 Choix technologiques	19
IV.4 Présentation de l'application	19
Conclusion générale	27
.....	27
Références	28

Introduction Générale

Le processus de don de sang dans notre pays est mal organisé de la part de l'Etat puisque nous voyons toujours des personnes qui cherchent quelqu'un, capable d'être un donneur et sauver leurs vies ou la vie d'un proche, en publiant leur demande sur les groupes du Facebook ou en achetant du sang qui est un acte inhumain.

Notre projet de fin d'année consiste à la conception et le développement d'une application mobile qui a pour but de faciliter le processus de don de sang et aider les personnes qui nécessitent un don de trouver un donneur.

Il est ainsi essentiel de donner un peu de soi en prenant un peu de son temps pour sauver une vie d'où nous avons facilité cette tâche pour le donneur qui va se connecter avec la personne en besoin de sang et faire son don en s'assurant qu'il a contribué à sauver une personne.

L'application va être minimaliste et optimisée avec une esthétique de l'application est assez sympathique. La personne qui nécessite un don de sang fait une demande (request) qui contient le type sanguin, le nom de l'hôpital et écrit une courte description de la situation, d'où une correspondance qui va se faire entre le donneur et le receveur soit par un appel du donneur ou par localisation de ses coordonnées.

Chapitre I : Contexte Général

I.1 Introduction

Dans ce chapitre composé de trois parties, nous exposons en premier lieu la présentation du projet. La deuxième partie est consacrée à l'étude de l'existant et sa critique. Et finalement nous présenterons les solutions proposées.

I.2 Présentation du projet

Notre projet de fin d'année consiste à la conception et le développement d'une application mobile de don de sang.

Notre application crée pour chaque acteur (Bénéficiaire, donneur et administrateur) un espace sur lequel bénéficier des fonctionnalités qui lui ai accordé selon son rôle.

Ces fonctionnalités sont principalement les suivants :

- S'inscrire au système pour créer une requête de sang (Bénéficiaire)
- S'inscrire au système pour consulter les requêtes de sang et les coordonnées de bénéficiaire (Donneur)
- Consulter les nouveautés de recherches de domaine (administrateur + utilisateurs)
- Gérer les utilisateurs inscrits (administrateur)

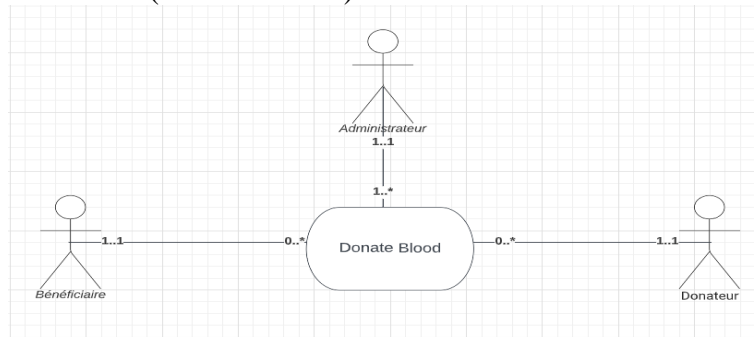


Figure I.3: diagramme de contexte [2]de notre application

Conclusion

Ce premier chapitre a été consacré à la présentation et la mise du projet dans son cadre général, en introduisant les problèmes à résoudre et les objectifs du projet.

Chapitre II : Spécification des besoins

II.1 Introduction

Dans ce chapitre, nous allons spécifier les besoins de l'application et les acteurs sur le système. Ensuite, on va présenter les diagrammes de cas d'utilisation associés et leur descripteurs textuels.

II.2 Identification des acteurs

Un acteur représente une abstraction d'un rôle joué par des entités externes qui interagissent directement avec le système.

Dans cette application, on distingue trois types d'acteurs :

- Le donneur : Un acteur qui représente toute personne qui peuvent donner de sang.
- Le Bénéficiaire : Un acteur qui représente toute personne qui créent une demande de sang.
- L'administrateur : Un acteur qui représente les personnels qui sont tachés de la gestion des demandes de sang et le bon fonctionnement de l'application.

II.3 Identification des besoins fonctionnels

II.3.1 Besoins fonctionnels

- Un donneur ou bénéficiaire peut créer, modifier, ou supprimer un compte.
- Un bénéficiaire peut créer, modifier, ou annuler une demande de sang.
- Un donneur peut savoir la liste des acteurs qui demande de sang et qui possèdent un type de sang compatible avec son type.
- Un donneur peut contacter un bénéficiaire.
- Un donneur peut connaître la localisation d'un bénéficiaire.
- Un administrateur créer, modifier, ou supprimer un blog.
- Un administrateur peut tirer les demandes selon l'ordre de priorité.

II.3.2 Besoins non-fonctionnels

Pour améliorer la qualité de ses services, le système doit répondre aux besoins suivants :

Performance : A travers ses fonctionnalités, l'application doit répondre à toutes les exigences des utilisateurs d'une manière optimale.

Ergonomie : Toute interface existante dans l'application doit être simple et pratique afin que l'utilisateur puisse l'exploiter sans se référer à des connaissances particulières. En d'autres termes, les informations doivent être lisibles et faciles à accéder par n'importe quel utilisateur.

Rapidité de traitement : La durée d'exécution des traitements doit s'approcher au maximum du temps réel.

Maintenabilité : Les différents modules développés de l'application doivent être faciles à maintenir. Nous devons respecter les standards de codage concernant par exemple les noms des attributs et des variables, les noms des méthodes ainsi que la disposition des commentaires.

II.4 Spécification de fonctionnalités de l'application

La figure II.1 présente le diagramme de cas d'utilisation [2] de notre application, nous distinguons les

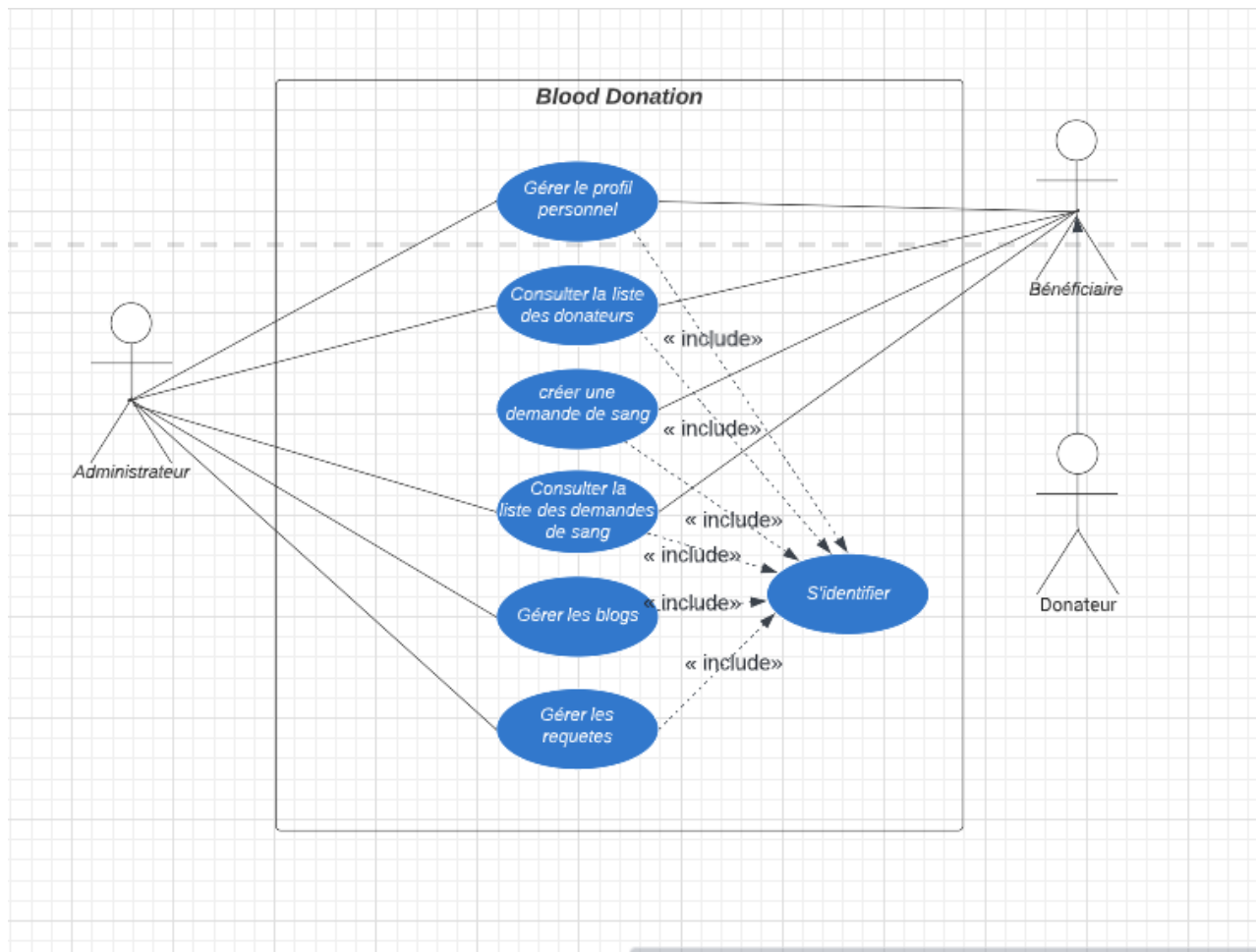


Figure II.1: Diagramme de cas d'utilisation [2] de Donate Blood

II.5 Identification des scenarios

Dans cette partie, on va clarifier chaque cas d'utilisation en forme textuelle.

II.5.1 Le cas d'utilisation *s'authentifier*

Cas d'utilisation	S'authentifier.
Acteur	Bénéficiaire ou Donneur ou administrateur.
Pré conditions	Avoir déjà un compte créé et connaître les identifiants.
Post conditions	Accès au compte.
Description	Ce cas d'utilisation permet à un acteur d'accéder aux fonctionnalités de l'application par saisie de son adresse mail et son mot de passe.
Scénario nominal	Le système vérifie la validité des données entrées dans la page d'inscription et renvoi la page d'accueil.
Scénario alternatif	Si les informations ne sont pas valides, on affiche une erreur correspondante. Si le compte n'existe pas, l'utilisateur doit créer un nouveau compte pour accéder à l'application.

II.5.2 Le cas d'utilisation *Consulter la liste de donateurs.*

Cas d'utilisation	Consulter la liste de donateurs.
Acteur	Bénéficiaire ou donneur ou Administrateur.
Pré conditions	Acteur authentifié.
Post conditions	Session active.
Description	Ce cas d'utilisation permet à un acteur d'accéder à la liste des donateurs.
Scénario nominal	Le système affiche la liste des donateurs inscrit dans l'application que l'acteur peut choisir pour les contacter.
Scénario alternatif	Si aucun donateur n'est inscrit, il s'affiche une liste vide.

II.5.3 Le cas d'utilisation Consulter la liste des demandes de sang

Cas d'utilisation	Consulter la liste des demandes de sang
Acteur	Bénéficiaire ou donneur ou Administrateur.
Pré conditions	Acteur authentifié.
Post conditions	Session active.
Description	Ce cas d'utilisation permet à un acteur d'accéder à la liste des requêtes de sang.
Scénario nominal	Le système affiche la liste des requêtes de sang dans l'application que l'acteur peut consulter les détails de requête (le bénéficiaire, la localisation, numéro téléphone..).
Scénario alternatif	Si la session est expirée on renvoi l'utilisateur à la page login.

II.5.4 Le cas d'utilisation Gérer le compte

Cas d'utilisation	Gérer le compte.
Acteur	Bénéficiaire ou donneur.
Pré conditions	Acteur authentifié.
Post conditions	Session active.
Description	Ce cas d'utilisation permet à un acteur de modifier les données du compte (email, nom, prénom, mot de passe) ou de le supprimer.
Scénario nominal	L'acteur doit entrer les nouvelles données aussi que l'ancienne mot de passe. Les modifications seront enregistrées dans la base de données.
Scénario alternatif	Si l'ancien mot de passe n'est pas correct, les modifications ne seront pas enregistrées.

II.5.5 Le cas d'utilisation Créer une demande de sang.

Cas d'utilisation	Créer une demande de sang
Acteur	Bénéficiaire.
Pré conditions	Acteur authentifié.
Post conditions	Session active.
Description	Ce cas d'utilisation permet à un acteur de créer une demande de sang contenant les données nécessaires et de pouvoir la modifier en cas d'erreur.
Scénario nominal	Le système vérifie la validité des données entrées et enregistre la demande dans la base de données.
Scénario alternatif	Si les informations ne sont pas valides, on affiche une erreur correspondante.

II.5.6 Le cas d'utilisation Gérer les requêtes.

Cas d'utilisation	Gérer les requêtes.
Acteur	Administrateur.
Pré conditions	Acteur authentifié ayant le privilège d'administrateur.
Post conditions	Session active.
Description	Ce cas d'utilisation permet à un acteur d'avoir accès à la liste des demandes de sang et les trier par un ordre de priorité (selon la priorité du cas).
Scénario nominal	Le système vérifie l'acteur est un administrateur. L'option de gestion des demandes sera accessible. L'administrateur pourra accéder la liste des requêtes non achevées et les trier par un ordre de priorité.
Scénario alternatif	Si l'acteur n'a pas le privilège d'administrateur, ces fonctionnalités ne seront pas disponibles.

II.5.7 Le cas d'utilisation *Gérer les blogs*.

Cas d'utilisation	Gérer les blogs.
Acteur	Administrateur.
Pré conditions	Acteur authentifié ayant le privilège d'administrateur.
Post conditions	Session active.
Description	Ce cas d'utilisation permet à un acteur d'avoir accès à la liste des blogs contenant les actualités sur le don de sang dans le monde.
Scénario nominal	Le système vérifie l'acteur est un administrateur. L'option de gestion des blogs sera accessible. L'administrateur pourra accéder la liste des blogs, ajouter, modifier et supprimer les blogs.
Scénario alternatif	Si l'acteur n'a pas le privilège d'administrateur, ces fonctionnalités ne seront pas disponibles.

Conclusion :

Dans ce chapitre nous avons identifié les différents besoins fonctionnelles et non fonctionnelles du système. Ensuite nous avons présenté le diagramme de cas d'utilisation global tout en identifiant chaque scénario. Le chapitre suivant sera consacré à détailler la conception de la présente application.

Chapitre III : Architecture et Conception

III.1 Introduction

Dans ce chapitre nous allons présenter les différents types de conception du projet :

- Conception de l'architecture
- Conception détaillée

En conséquence, on va présenter les architectures utilisées dans Backend et le Frontend, le diagramme de classes et les diagrammes de séquences qui représentent notre application.

III.2 Conception générale

III.2.1 Architecture

L'architecture technique est une vue tournée vers les différents éléments matériels et l'infrastructure dans laquelle le système informatique s'inscrit, les liaisons physiques et logiques entre ces éléments et les informations qui y circulent. La figure III.2 présente un schéma de l'architecture technique de notre application.

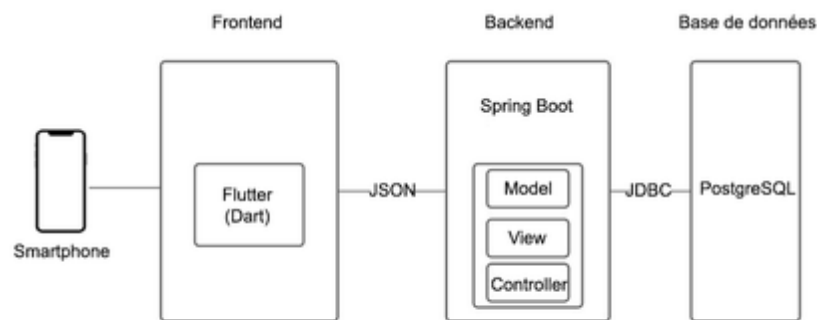


Figure III.2: Architecture technique de Donate Blood

III.2.2.1 Architecture du Backend

Le développement du backend a été assuré par le Framework Spring Boot. Ce Framework utilise l'architecture Model-View-Controller MVC.

- Un modèle (Model) contient les données à afficher.
- Une vue (View) contient la présentation de l'interface graphique.
- Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.

III.2.2.2 Architecture du Frontend

Le développement du Frontend a été assuré par le Flutter. Il s'agit d'un Framework de la langage Dart.

III.3 Conception détaillée

Dans cette partie on présente la conception technique de la vue dynamique de notre application ainsi que sa vue statique. La vue dynamique est représentée par des diagrammes UML de séquence.

III.3.1 Diagramme de classes

Le diagramme de classes [2] modélise les entités de notre système présenté par la figure III.3.

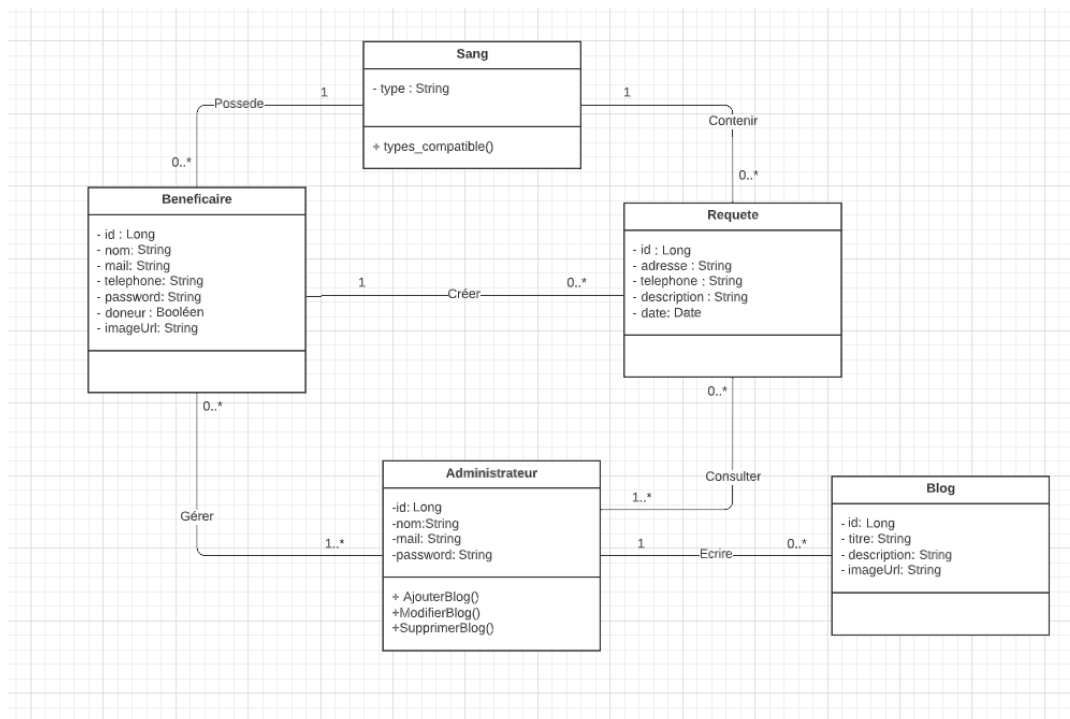


Figure III.3: Diagramme de classes [2] de Donate Blood

III.3.1 Diagrammes de Séquences

Dans cette section nous allons présenter le vu dynamique des cas d'utilisations s'authentifier et créer une demande de sang à travers les diagrammes de séquences des figure III.4 et III.5.

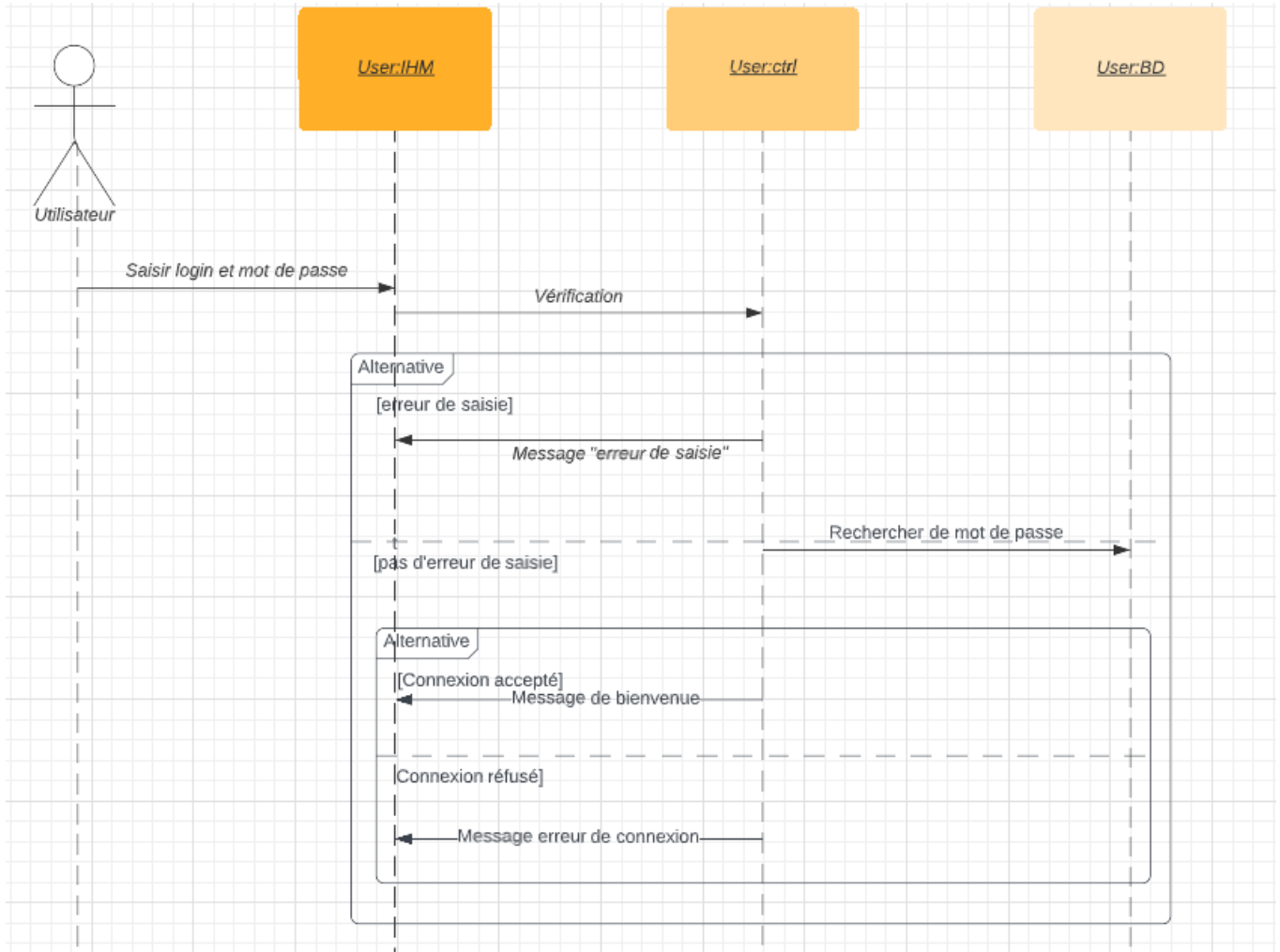


Figure III.4: Diagramme de séquence pour « s'authentifier »

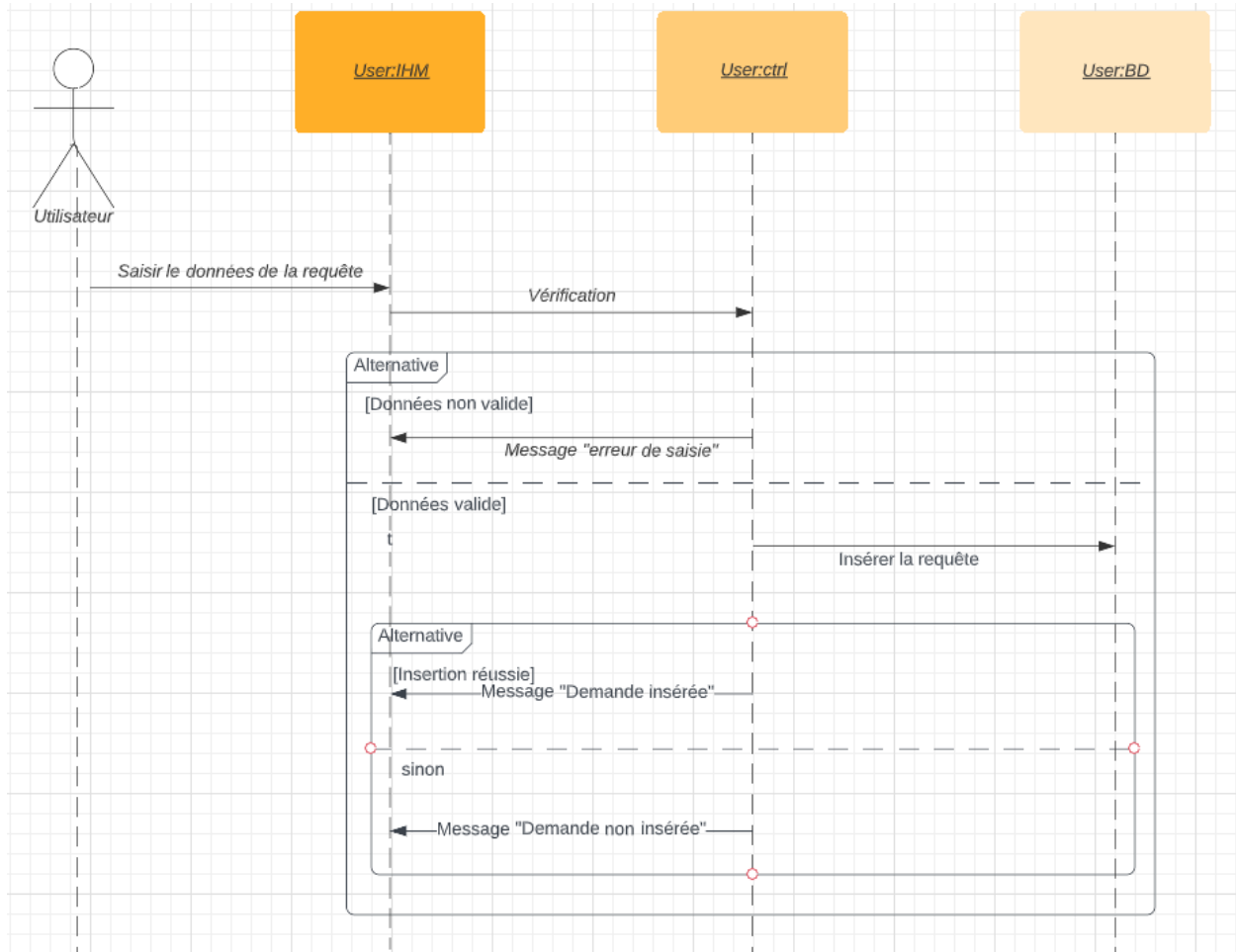


Figure III.5: Diagramme de séquence pour « Créer une demande de sang »

Conclusion :

Dans ce chapitre nous avons présenté l'architecture opérationnelle et l'architecture technique de notre application. Aussi nous avons développé le diagramme de classes, ainsi que le diagramme de Séquences

Chapitre IV : Réalisation

IV1 Introduction

Ce chapitre sera consacré à la description des technologies utilisées. On présentera l'environnement matériel et logiciel de notre application, aussi que les différents modules à travers l'interfaces graphiques correspondantes.

IV2 Environnement matériel

Notre environnement matériel est constitué par un ordinateur portable dont les caractéristiques sont présentées ci-dessous :

Informations système

Date/heure du jour : lundi 30 mai 2022, 16:35:58
Nom de l'ordinateur : LASSOUED
Système d'exploitation : Windows 10 Professionnel 64 bits (10.0, build 19042)
Langue : français (Paramètres régionaux : français)
Fabricant du système : LENOVO
Modèle du système : 81Y4
BIOS : EGCN36WW
Processeur : Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz (8 CPUs), ~2.5GHz
Mémoire : 24576MB RAM
Fichier de pagination : 17975 Mo utilisé(s), 10052 Mo disponible(s)
Version DirectX : DirectX 12

IV3 Environnement logiciel

IV31 Architecture logiciel

Dans cette section, on va présenter les différents logiciels utilisés dans chaque partie de l'application tout en exposant les différentes couches de la partie Backend et la partie Frontend.

IV32 Choix technologiques

Nous présentons dans cette partie les choix concernant l'environnement logiciel que nous avons utilisé.

- **Spring** : se définit comme étant un Framework libre. Il permet de construire l'infrastructure d'une application Java et d'en faciliter le développement. Le Spring offre plusieurs fonctionnalités, à savoir la gestion transactionnelle, celle des exceptions Java DataBase Connectivity et un Framework MVC.
- **Flutter** : le Framework de Google permettant de concevoir des applications multiplateformes pour Android et iOS.
- **Postman** : sert à exécuter des appels HTTP directement depuis une interface graphique. Vous pourrez simplement choisir l'URL, la méthode HTTP (le plus souvent GET, POST, PUT, PATCH et DELETE), les headers, les query params et dans certains cas le body de la requête.
- **PostgreSQL** : est un système de gestion de base de données relationnelle orienté objet puissant et open source qui est capable de prendre en charge en toute sécurité les charges de travail de données les plus complexes.
- **Android Studio** : est un environnement de développement pour développer des applications mobiles Android.
- **IntelliJ IDEA** : est un environnement de développement intégré destiné au développement de logiciels informatiques reposant sur la technologie Java

IV4 Présentation de l'application

On présentera notre application à travers une série de captures d'écran, chacune représentant une fonctionnalité du système.

IV41 Page d'accueil



Figure 1: La page d'accueil de Donate Blood

IV42 Authentification

L'utilisateur doit remplir le formulaire avec des données valide pour avoir accès à l'application.

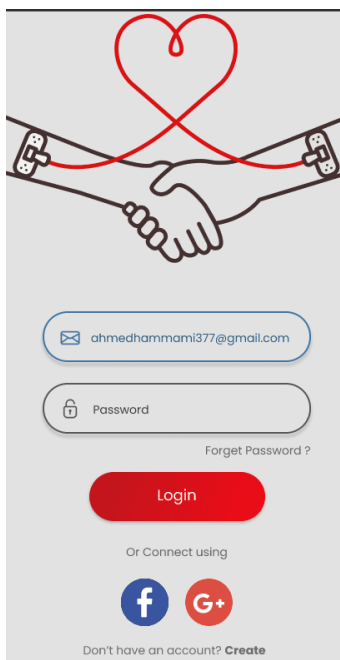
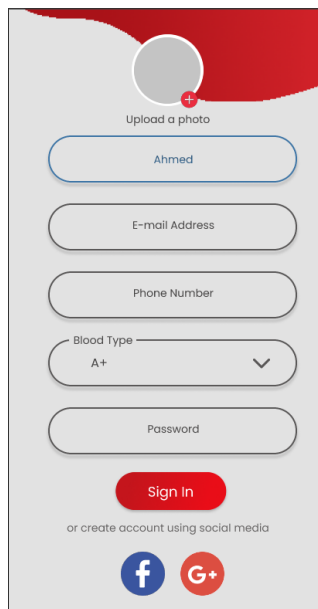


Figure 2: La page de login de Donate Blood

IV43 Création du compte

Si l'utilisateur n'a pas un compte, il doit le créer d'abord.

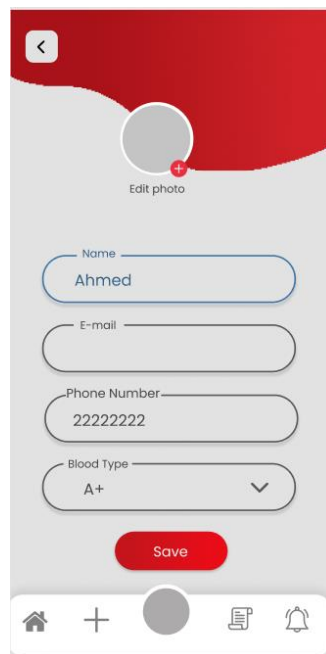


The registration form is displayed on a mobile screen with a red header. It features a circular profile picture placeholder with a red '+' icon and the text 'Upload a photo'. Below this are input fields for 'Ahmed', 'E-mail Address', 'Phone Number', and 'Blood Type' (set to 'A+' with a dropdown arrow). A 'Password' field is also present. A red 'Sign In' button is located below the password field, followed by the text 'or create account using social media'. At the bottom are icons for Facebook and Google+.

Figure IV.3: Page d'inscription de Donate Blood

IV44 Modifier Profil

Cette page permet à l'utilisateur de modifier son profil.



The profile modification form is shown on a mobile screen with a red header. It includes a back arrow in the top left corner. The profile picture placeholder has a red '+' icon and the text 'Edit photo'. The form contains input fields for 'Name' (Ahmed), 'E-mail', 'Phone Number' (22222222), and 'Blood Type' (A+ with a dropdown arrow). A red 'Save' button is positioned below the blood type field. At the bottom is a navigation bar with icons for home, add, profile, list, and notifications.

Figure IV.4: La page de modification de profil de Donate Blood.

IV45 Consulter le profil

Le profil contient toutes les données de l'utilisateur de l'application

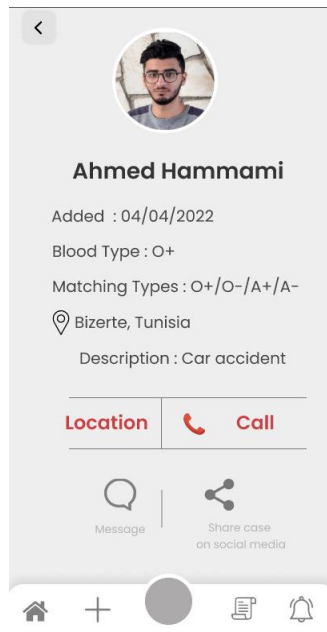


Figure IV.5: La page de profil d'un utilisateur de Donate Blood

IV46 Ajouter une requête.

Ce formulaire est doit être rempli avec des données valides pour qu'il sera enregistré. Ces données seront ensuite modifiables à partir du même formulaire.

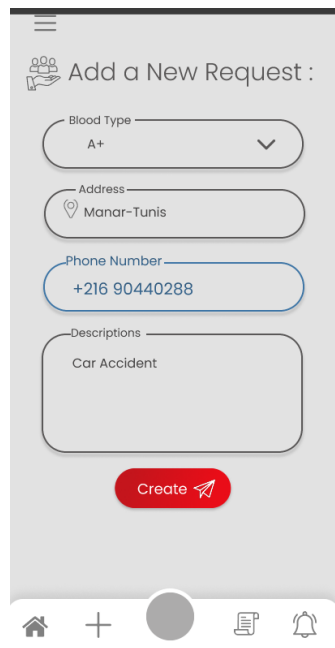


Figure IV.6: Page d'ajout d'une requête.

IV47 Consulter les demandes de sang.

Cette page contient des données relatives à la liste des demandes de sang.

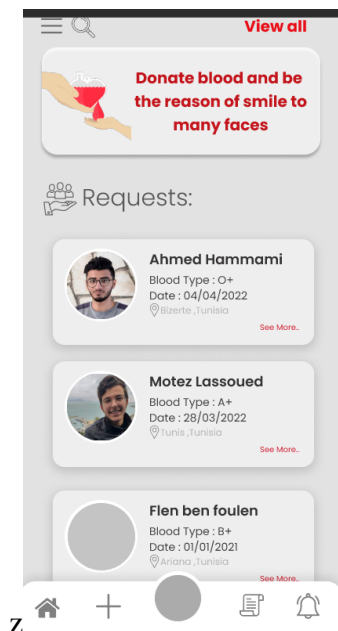


Figure IV.7: Page demandes de sang de Donate Blood

IV48 Devenir un donneur.

Cette page permet à l'utilisateur de devenir un donneur.

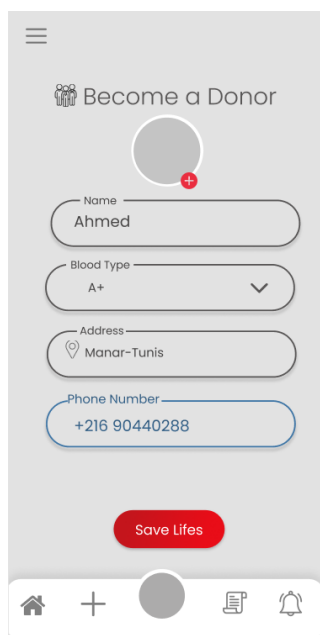


Figure IV.8: Page devenir donneur de Donate Blood

IV49 Consulter la liste des demandes de vaccination

Cette permet de consulter la liste des donneurs dans l'application.

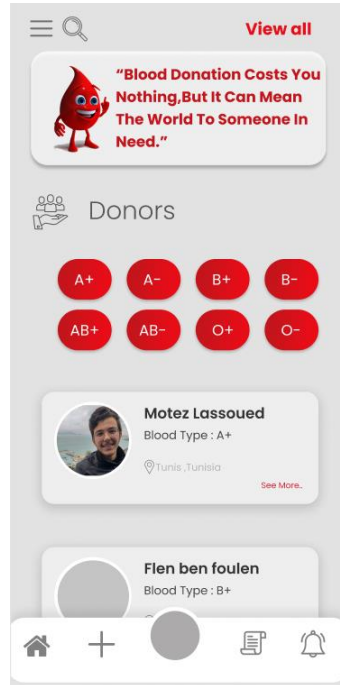


Figure IV.9: Page de donneurs de Donate Blood

IV410 Consulter liste de blogs

Cette page permet de consulter la liste des blogs écrite par l'administrateur.

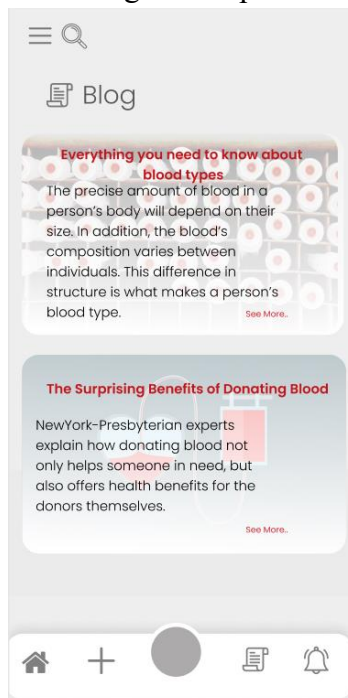


Figure IV.10: Page de liste blog de Donate Blood.

IV411 Blog

Cette page permet à un utilisateur lire le contenu du blog.

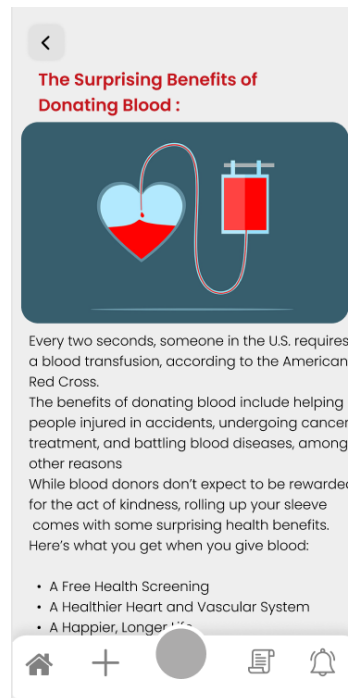


Figure IV.11: Page de blog de Donate Blood.

IV412 Localisation

Cette page permet d'afficher la localisation de bénéficiaire pour faciliter la procédure de don.

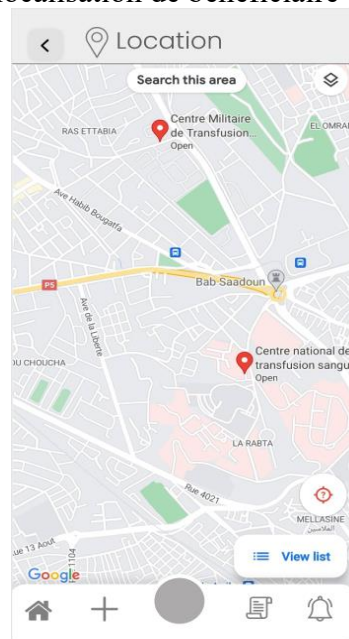


Figure IV.12: La page de localisation de Donate Blood.

IV.13 Communication téléphonique

Quand le donneur veut contacter le bénéficiaire en appuyant sur ‘Call’ cette page apparaisse et permet la communication avec lui.

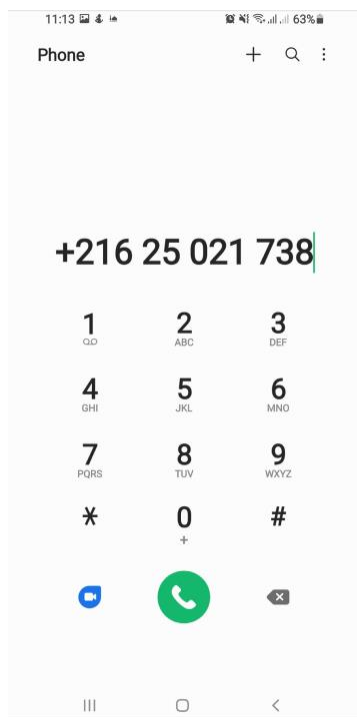


Figure IV.13: La page d'appel téléphonique de Donate Blood.

Conclusion

Dans ce dernier chapitre, nous avons présenté l'environnement matériel et logiciel de notre application, ainsi que les technologies utilisées en Backend et en Frontend.

Ensuite, à travers des captures d'écran, nous avons présenté les différentes pages et interfaces de notre application Donate Blood.

Conclusion générale

Le présent rapport est l'effort d'un travail successif réalisé dans le cadre de notre projet de fin d'année en IF4.

Le travail a pour but de faciliter et accélérer la procédure de don de sang entre le donneur de le bénéficiaire sans intervention d'un intermédiaire. Peut-être notre application va sauver des vies.

Nous avons entamé notre projet par une présentation du cadre général. Ensuite nous nous sommes penchés sur l'analyse minutieuse des besoins fonctionnels et non fonctionnels de notre système. Après, nous avons réalisé une conception détaillée de l'application en utilisant différents diagrammes du langage UML.

Nous avons enchaîné avec la partie la plus importante de notre projet, il s'agit de la phase de réalisation qu'on a présenté en insérant plusieurs captures d'écran de notre plateforme.

Finalement, nous espérons avoir été à la hauteur des attentes du jury et de notre encadrant Monsieur Heithem Abbes.

Références

- [1] Flutter: <https://docs.flutter.dev/>
- [2] Dart: <https://dart.dev/>
- [3] Spring : <https://spring.io/>
- [4] Spring Boot: <https://spring.io/projects/spring-boot>
- [5] Spring Security : <https://spring.io/projects/spring-security>
- [6] PostgreSQL : <https://www.postgresql.org/>
- [7] Hibernate : <https://hibernate.org/>
- [8] Postman : <https://www.postman.com/>
- [9] Maven: <https://maven.apache.org/>
- [10] Figma: <https://www.figma.com/>

Un grand remerciement à AmigosCode et Maximilian Schwarzmüller pour les tutoriaux de Spring Boot et Flutter.

* <https://amigoscode.com/>

* <https://academind.com/>