

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'enseignement Supérieur et de la Recherche Scientifique

Université de Mouloud Mammeri de Tizi-Ouzou

Faculté DE Génie Électrique et d'Informatique

Département d'Informatique.



Mémoire de fin d'étude

En vue de l'obtention du Diplôme de Master 2 en Informatique

Option : Conduite de projets Informatiques.

Thème :

Développement d'une application logicielle multiplateforme pour l'assistance médicale des étudiants.

Présenter par

- ❖ Boubchir Yasmina.
- ❖ Toumi Abdelkrim.

Devant le jury composé de :

* Monsieur Sadi.S

* Madame Bellatef.

Sous la direction du professeur

Bouarab.F

Année Universitaire :2019/2020.

Dédicaces

Je dédie ce travail

À mes très chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, et leur soutien tout au long de mes études,

À mes chères sœurs et chers frères pour leurs encouragements permanents, et leur soutien moral,

À la mémoire de ma grande-mère Saadia qui m'aimait beaucoup ,qui aurait aimé voir ce jour.

À mes chères amies Nadine.H ,Karima.S et chers amis Abdelhak.Y et Anis.K pour leur soutien et encouragement durant ce parcours.

À mon binôme Abdelkrim qui m'a toujours compris, soutenu, et cru en moi et sans qui ce travail n'aurait jamais été fait.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible. Merci d'être toujours là pour moi.

Yasmina.B

Dédicaces

Je dédie cet ouvrage

*A mes parents qui m'ont soutenu et encouragé durant ces années d'étude,
qu'ils trouvent ici le témoignage de ma profonde reconnaissance.*

*A mes frère et sœurs, mes grands-parents et ceux qui ont partagé avec moi tous les moments
d'émotion lors de la réalisation de ce travail. Ils m'ont chaleureusement supporté et encouragé
tout au long de mon parcours.*

*A tous mes amis ; Camprz ; Zery, Katia, Daoud, Massyl qui m'ont toujours encouragé, et à
Qui je souhaite plus de succès.*

A ma binôme Yasmina qui, sans elle ce travail ne serait jamais fait.

A tout ceux que j'aime.

Merci !

Abdelkrim. T

Remerciements

A l'issue de ce travail ,on remercie ,en premier lieu le bon Dieu de nous avoir donné la force et le courage de le mener à terme.

Á nos très chers parents, notre fierté et bien sur la source de notre réussite pour leurs sacrifices et encouragement. Que dieux nous les garde et les protège.

Á notre Encadreur le professeur BOUARAB.F pour son professionnalisme, ses remarques pertinentes, son soutiens et sa motivation, nous la remercions infiniment de nous avoir fait confiance et d'avoir autant cru en nous et pour chaque petit détail qu'il nous a appris tout au long de notre cursus.

Aux membres du jury qui ont accepté d'évaluer notre travail.

Á nos très chers frères, soeurs, ami (es) et à toutes les personnes qui ont contribué de loin ou de près à la réussite de ce travail.

Yasmina.B Abdelkrim.T

Table des matières:

Liste des Figures	8
Liste des Tableaux	9
Introduction générale	10
Chapitre I :Les systèmes d'aide à la décision (DSS)	12
Introduction	12
1. Définition d'un système d'aide à la décision	12
2. Historique	13
3. Objectifs du DSS.....	14
4. Utilité d'un DSS.....	14
5. L'assistance d'aide à la décision en santé.....	15
5.1 Demande d'assistance	15
5.2 L'assistance médicale	15
5.3 Services d'assistance	16
5.3.1 Métiers des services d'assistance	16
6. L'assistance médicale dans les universités	17
6.1 Un service de santé universitaire	17
6.2 Agent d'Hygiène Sécurité Environnement (HSE)	17
6.3 Exemples d'assistance médicale	17
7. Utilisation de Chat en santé	18
7.1 Usages de chat	18
7.1.1 Usages personnels	18
7.1.2 Usages professionnels	18
7.2 Comment utiliser le chat	19
Conclusion	20
Chapitre II :Le multiplateforme.....	21
Introduction	21
1. Définition du logiciel multiplateforme.....	21
2. L'intérêt du logiciel multiplateforme.....	22
3. Le développement natif et multiplateforme.....	23
4. Inconvénients du développement multiplateforme	23

5. Les différentes solutions multiplateformes	24
5.1 Full web	24
5.2 Applications Hybrides	25
5.3 Xamarin	27
6. Présentation de quelques frameworks dédiés au développement multi plateformes	27
6.1 Unity	27
6.2 React-Native.....	28
6.3 Flutter.....	28
Conclusion	29
Chapitre III : Analyse et conception	30
Introduction	30
1.Processus de développement de notre solution.....	30
2. Spécification et analyse des besoins	30
2.1 Etablissement d'un cahier des charges	31
2.2 Etude préliminaire	31
2.2.1 Besoins Fonctionnels.....	31
2.2.1.1 Identification des acteurs	31
2.2.1.2 Elaboration du diagramme de contexte.....	32
2.2.1.3 Elaboration de diagramme de cas d'utilisation	33
2.2.2 Besoins non Fonctionnels	38
2.3 La phase de conception.....	38
2.3.1 Conception préliminaire	38
2.3.1.1 Architecture logicielle du système	38
2.3.2 La vue logique de notre cas d'étude	39
2.3.2.1 Diagramme de classe pour notre cas d'étude	40
2.3.2.2 Diagramme de séquence de notre cas d'étude	42
Conclusion	48
Chapitre IV:Implémentation et réalisation	49
Introduction	49
1. Les langages de programmation et outils utilisés	49
1.1 Flutter	49

1.2	Dart	50
1.3	Firebase	51
1.4	CloudFirestore	51
1.4	Base de données à temps réel (Firebase Real Time Database)	52
1.4	Git	52
1.4	GitKraken	53
1.5	Visual Studio Code.....	54
2.	La description de notre application :	55
2.1	Interface Authentification et demande d'inscription	55
2.2	Interface Agent HSE	56
2.2.1	Interface d'Accueil.....	56
2.2.2	Interface Agent HSE (Liste Demandes)	56
2.2.3	Interface Détails de la demande	57
2.2.4	Interface Liste D'étudiants.....	57
2.2.5	Interface Détails Etudiant	58
2.2.6	Interface Liste des Alertes Validées	58
2.2.7	Interface Détails de l'Alerte Validée	59
2.2.8	Interface Ajout Etudiant	59
2.2.9	Interface Liste De Médecins	60
2.2.10	Interface Détails de Médecin	60
2.2.11	Interface Ajout d'un Médecin	61
2.3	Interface Etudiant Inscris.....	62
2.3.1	Interface Accueil Etudiant	62
2.3.2	Interface Profil Etudiant	62
2.3.3	Interface Calendrier Etudiant.....	63
2.3.4	Interface Validation d'Alerte	63
2.3.5	Interface de Liste des maladies	64
2.3.6	Interface d'information sur la maladie.....	64
2.4	Interface Messagerie	65
2.4.1	Interface liste des contacts	65
2.4.2	Interface recherche de Contact.....	65

2.4.3	Interface Discussion	66
2.4.4	Interface message d'alerte textuelle	66
2.5	Interface des Partenaires d'alerte	67
2.5.1	Interface du Samu.....	67
2.5.2	Interface de la Protection Civile.....	67
2.5.3	Interface Details D'une Alerte	68
2.5.4	Interface d'affectation d'un Ambulancier	68
2.5.5	Exemple d'un SMS envoyé à un Ambulancier	69
2.5.6	Interface Ajout D'ambulancier	69
2.5.7	Interface de la carte GoogleMaps	70
2.5.8	Interface « fiche message reçu »	70
2.5.9	Interface liste des alertes	71
2.5.10	Interface de la carte GoogleMaps	71
	Conclusion	72
	Conclusion générale	73
	webographie	75
	Annexe	81

Liste des figures :

Figure 1: Architecture full web.....	25
Figure 2 : application hybride.....	26
Figure 3 : Diagramme de contexte.....	33
Figure 4 : Diagramme de cas d'utilisation.	37
Figure 5 : Architecture globale de système.	39
Figure 6 : Diagramme de classe.....	41
Figure 7 : Diagramme de séquence : demande d'inscription de l'étudiant.....	42
Figure 8 :Diagramme de séquence :connexion de l'utilisateur a la plateforme.	43
Figure 9 : Diagramme de séquence : alerter le partenaire d'alerte... ..	43
Figure 10: Diagramme de séquence : demande assistance médecin.....	44
Figure 11: Diagramme de séquence : manipulation sur la liste d'étudiants.....	45
Figure 12: Diagramme de séquence :contacter un ambulancier	46
Figure 13: Diagramme de séquence :ajouter un médecin	46
Figure 14: Diagramme de séquence :envoi de message d'alerte par le médecin	47
Figure 15: Interface gitKraken de notre projet	53
Figure 16: Interface Authentification.....	55
Figure 17:Interface Demande d'inscription.....	55
Figure 18: Interface d'Accueil	56
Figure 19: Interface Agent HSE (Liste Demandes)	56
Figure 20: Interface Details de la demande	57
Figure 21: Interface Liste D'étudiants.	57
Figure 22: Interface Details Etudiant	58
Figure 23: Interface Liste des Alertes Validées.....	58
Figure 24: Interface Details de l'Alerte Validée.....	59
Figure 25: Interface Ajout Etudiant	59
Figure 26: Interface Liste De Médecins.	60
Figure 27: Interface Détails de Médecin.....	60
Figure 28: Interface Ajout d'un Médecin.....	61
Figure 29: Interface Accueil Etudiant.....	62

Figure 30: Interface Profil Etudiant .	62
Figure 31: Interface Calendrier Etudiant .	63
Figure 32: Interface Validation d'Alerte.	63
Figure 33: Interface de Liste des maladies .	64
Figure 34: Interface d'information sur la maladie.....	64
Figure 35: Interface liste des Contacts.	65
Figure 36: Interface recherche de Contact.....	65
Figure 37: Interface Discussion .	66
Figure 38: Interface message d'alerte textuelle.	66
Figure 39: Interface du Samu	67
Figure 40: Interface de la Protection Civile	67
Figure 41: Interface Details D'une Alerte.	68
Figure 42: Interface d'affectation d'un Ambulancier	68
Figure 43: Exemple d'envoi d'SMS.	69
Figure 44: Interface Ajout D'ambulancier.	69
Figure 45: Interface de la géolocalisation de l'alerte	70
Figure 46: Interface « fiche message reçu ».	70
Figure 47: Interface liste des alertes.....	71
Figure 48: Interface liste des messages reçus.	71

Listes des tableaux :

Figure 1:Tableau des taches et scénarios.	34
--	----

Introduction générale :

Depuis l'arrivée des systèmes d'information au sein des entreprises, puis les systèmes d'aide à la décision dans les différents domaines précisément au sein de la santé ; de nombreux logiciels comme les assistances médicales sont venus se concurrencer en proposant toujours de nouvelles fonctionnalités et spécialisations.

Les applications de santé contribuent à améliorer la qualité de vie des patients souffrant d'une maladie ou en cas de besoin d'aide, grâce aux possibilités de surveillance, système d'alerte et au suivi qu'elles offrent. L'objectif de notre projet présenté dans ce mémoire ; est de développer une application d'assistance médicale dédiée aux étudiants à maladies chroniques ; et qui peut toucher éventuellement les étudiants à besoin spécifiques . Cette application offre un ensemble de fonctionnalités permettant une intervention rapide et efficace en cas d'urgence d'un partenaire d'alerte (SAMU et la Protection Civile par exemple dans notre cas) tel que ce dernier va alerter des ambulanciers de son service afin d'évacuer l'étudiant malade, aussi, faciliter la communication de l'étudiant avec son médecin qui va lui offrir l'assistance dont cet étudiant aura besoin.

Pour mener à terme notre travail ; Le manuscrit est composé de quatre chapitres :

❖ Chapitre 1 : les systèmes d'aide à la décision (DSS) en santé :

Ce chapitre présente le système d'aide à décision, son historique, ses objectifs, ainsi que l'assistance d'aide à la décision en santé où on définit qu'est ce qu'est une l'assistance médicale et présenter les services cette dernière et leurs métiers. Parler sur l'assistance médicale dans les universités en définissant le service de santé universitaire, l'agent HSE (Hygiène Sécurité Environnement) ; quelques exemples d'assistance médicale ainsi que l'utilisation de CHAT (Messagerie Instantanée) en santé.

Chapitre 2 : Le multiplateforme

Ce chapitre définit le multiplateforme, les avantages et inconvénients de développement multiplateforme ; ainsi nous avons présenté de différentes solutions multiplateformes et quelques frameworks dédiés au développement multiplateforme.

❖ **Chapitre 3 : Analyse et conception**

Ce chapitre illustre la phase de conception de notre projet en utilisant plusieurs diagrammes UML.

❖ **Chapitre 4 : Implémentation et réalisation**

Dans ce chapitre nous présentons les outils et langages utilisés pour la réalisation de notre application et nous montrons leurs interfaces.

Chapitre I : Les systèmes d'aide à la décision (DSS)

Introduction

La contribution des systèmes d'aide à la décision (DSS : Decision Support Systems) dans les soins de santé a été de grande envergure et continue d'évoluer ainsi que l'augmentation de la demande d'assistance ; Dans notre vie quotidienne, plusieurs personnes sont atteintes de maladies chroniques et trouvent beaucoup de difficultés à prendre des décisions en cas de risque sur leurs vies.

Dans notre cas d'étude nous nous concentrons sur les étudiants à maladies chroniques, ainsi qu'aux étudiants à besoins spécifiques (étudiant a besoin spécifique est un étudiant en situation de handicap ou souffrant d'une maladie invalidante, d'une déficience avérée) en besoin d'aide à tout moment. Dans certains cas, c'est des décisions pas familières qui nécessite quelqu'un qui est plus expérimenté pour demander de l'aide ; tel qu'un système d'assistance cherche à guider l'utilisateur vers des décisions qu'il aura à prendre dans une situation particulière.

Dans ce chapitre on va définir c'est quoi un système d'aide à la décision (DSS), ces objectifs et utilité ainsi l'assistance d'aide à la décision en santé ou nous avons défini qu'est ce qu'est une demande d'assistance, assistance médicale et présenter les services d'assistance et leurs métiers ainsi que l'utilisation de chat dans le domaine de santé.

1. Définition d'un système d'aide à la décision

Un système d'aide à la décision (**DSS: Decision Support System**) est un système d'assistance informatisé qui recueille et analyse les données, les synthétisant pour produire des rapports d'information complets et aident les décideurs à utiliser les technologies de communication, les données, les documents, les connaissances et / ou les modèles pour identifier et résoudre les problèmes, effectuer les tâches du processus de décision et prendre des décisions qui peuvent changer rapidement et ne pas être facilement précisées à l'avance .Cela constitue une aide qui permettra au décideur d'élaborer une représentation pertinente de la situation.[1]

Selon Roy et Bouyssou (1993), l'aide à la décision est définie comme étant l'activité de celui (homme d'étude) qui, prenant appui sur des modèles clairement explicités et plus ou moins complètement formalisés, cherche à obtenir des éléments de réponses aux questions que se pose

un intervenant (décideur) dans un processus de décision, éléments concourant à éclairer la décision et normalement à prescrire un comportement de nature à accroître la cohérence entre l'évolution du processus d'une part, les objectifs et le système de valeurs au service duquel cet intervenant se trouve placé d'autre part.[2]

2. Historique

Le concept d'aide à la décision a principalement évolué à partir des études théoriques sur la prise de décision organisationnelle effectuées au Carnegie Institute of Technology à la fin des années 1950 et au début des années 1960, et du travail de mise en œuvre effectué dans les années 1960 par les chercheurs qui ont commencé à étudier systématiquement l'utilisation de modèles quantitatifs informatisés pour aider à la prise de décisions et à la planification.[3]

Vers 1970, des revues commerciales ont commencé à publier des articles sur les systèmes de Décision de gestion, les systèmes de planification stratégique et les systèmes d'aide à la décision (cf. Sprague et Watson 1979). Par exemple, Scott Morton et ses collègues McCosh et Stephens ont publié des articles sur l'aide à la décision dans 1968. La première utilisation du terme système d'aide à la décision a été dans l'article de Sloan Management Review de Gorry et Scott-Morton (1971). Ils ont fait valoir que les systèmes d'information de gestion étaient principalement axés sur les décisions structurées et ont suggéré que les systèmes d'information à l'appui des décisions semi-structurées et non structurées soient appelés « systèmes d'aide à la décision ». Ce qui a rendu le DSS un domaine de recherche à part entière au milieu des années 1970, avant de gagner en intensité au cours des années 1980.[3]

Les systèmes d'aide à la décision ont une longue histoire dans le milieu de la santé ,en 1966 l'une des premières applications de la théorie de la décision se trouve dans les méthodes d'évaluation de la qualité des soins médicaux décrites par Donabedian et l'Agence pour la recherche et la qualité des soins de santé (AHRQ) a financé les premières recherches qui ont mené à l'utilisation des technologies de l'information pour surveiller et évaluer la prise de décision clinique et les DSS ont contribué à la qualité, à l'efficacité, à l'atténuation des risques et afin de prendre de meilleures décisions, les établissements de santé ont activement intégré les capacités DSS dans leurs systèmes existants.[4]

Les systèmes d'aide à la décision fonctionnent à de nombreux niveaux, et il existe de nombreux exemples d'utilisation courante au quotidien. Par exemple, la planification d'itinéraire GPS détermine l'itinéraire le plus rapide et le meilleur entre deux points en analysant et en comparant

plusieurs options possibles. De nombreux systèmes GPS inclut également des capacités d'évitement de la circulation qui surveillent les conditions de circulation en temps réel, permettant aux automobilistes d'éviter les encombrements. Les agriculteurs utilisent des outils de planification des cultures pour déterminer le meilleur moment pour planter, fertiliser et récolter. Un logiciel de diagnostic médical qui permet au personnel médical de diagnostiquer des maladies en est un autre exemple.[5]

3. Objectifs du DSS

- Pour gagner du temps et des efforts dans le processus de prise de décision.
- Pour aider au traitement des données collectées et à la production d'une solution suggérée à un problème.
- Pour fournir une analyse sophistiquée et rapide d'une grande quantité de données et d'informations à tous les niveaux de gestion, principalement dans des situations semi-structurées ou non structurées en réunissant le jugement humain et l'information informatisée.
- Promouvoir l'apprentissage, ce qui conduit à de nouvelles exigences et au raffinement de l'application.
- Améliorer l'efficacité des administrateurs dans la prise de décision.
- Fournir une solution efficace et efficiente de chaque problème complexe, vérifier l'impact des changements sur la solution proposée à l'aide d'une analyse par simulation.
- Utiliser l'analyse de recherche d'objectifs pour trouver la valeur des entrées nécessaires pour atteindre le niveau de sortie souhaité.

4. Utilité d'un DSS

Afin de mieux expliquer l'utilité et les avantages que pourrait avoir un DSS ; nous devrions tous d'abord définir ceux qui peuvent profiter des solutions qu'un DSS pourrait offrir.

Les systèmes d'aide à la décision, dans notre cas, aiderait les étudiants et les établissements de santé à définir l'état de cet étudiant concerné et aider ces établissements à mieux prendre la décision optimale.

Les avantages :

- Fourniture automatique de conseils d'experts, d'expertises et de recommandations pertinents et personnalisés provenant de connaissances à jour et des meilleures pratiques.
- Peut aider à surmonter les problèmes de codage inefficace des données.
- Peut être rentable après les coûts d'investissement initiaux et les coûts de mise à jour et de maintenance.
- Peut fournir une rétroaction immédiate aux patients.
- S'il est intégré à un EMR (Electronic Medical Record), peut aider à rationaliser le flux de travail (prise de l'historique, diagnostic, traitement) et encourager une collecte de données plus efficace.
- Peut maintenir et améliorer la cohérence des soins.

5. L'assistance d'aide à la décision en santé

Dans le domaine de la santé tout est critique et nécessite une grande réflexion pour savoir trouver de l'aide nécessaire à un cas précis pour cela l'assistance est toujours requis.

5.1 Demande d'assistance

En Informatique, une demande d'assistance est une demande faite par l'utilisateur d'une application informatique, concernant un incident rencontré, une évolution souhaitée, ou une mauvaise compréhension des règles de gestion. Cette demande peut parvenir à partir de plusieurs canaux Téléphone ;Courriel ou Logiciel de gestion des services d'assistance . [6]

5.2 L'assistance médicale

Une assistance désigne l'action de porter aide ou secours a quelqu'un en difficulté. C'est l'aide, l'appui ou le secours organisés en toutes matières, et peut désigner ou qualifier une institution ou un organisme.[7]

Une assistance médicale est un système utilisé dans un contexte médical désigné pour apporter de l'aide aux personnes souffrantes d'un problème de santé compromettant à court terme leur état de santé. Elle permet de donner aux patients de meilleures relations avec les médecins et les services de santé.[7]

5.3 Services d'assistance

Ils consistent à garantir que les utilisateurs d'un système puissent continuer à profiter de la disponibilité de l'ensemble de ses composants pour l'accomplissement de leurs tâches. L'assistance peut se porter sur des applications, ou sur des composants techniques (plateformes informatiques, réseaux). Les services d'assistance sont parfois réalisés par des équipes rassemblées dans un centre d'assistance, ou réparties sur plusieurs sites dans des cellules d'assistance spécialisées, de taille plus réduite.[8]

5.3.1 Métiers des services d'assistance :

➤ **Téléassistant (*technicien hotline*)**

Celui-ci emploie le plus souvent une base de connaissance préétablie permettant de répondre aux questions des utilisateurs.[9]

➤ **Technicien helpdesk**

Il possède en plus de la base de connaissances d'autres outils pour assister l'utilisateur. Citons, parmi ceux-ci, la télémaintenance et l'assistance sur site comme dans notre cas concernant les étudiants à besoin spécifiques ou y'aura soit affectation d'outils en cas de besoin (par exemple un objet connecté tel qu'un fauteuil roulant) ou l'envoi de personnes expérimentés pour ce genre de situations. Le centre d'assistance peut apporter une réponse, donner un conseil, transmettre le problème à un spécialiste suivant le degré d'urgence et le niveau de difficulté.[9]

➤ **Consultant ou ingénieur en support technique ou fonctionnel**

Il intervient sur la gestion d'incidents complexes nécessitant une connaissance approfondie de la technologie utilisée et de ses diverses composantes techniques et fonctionnelles. Il participe à la résolution des problèmes, et propose une maintenance corrective et évolutive et participe à la gestion des configurations et des activités connexes. Ces plateformes d'assistances ne sont cependant ouvertes que pendant les horaires de bureau, mais permettent d'être traitée rapidement. 10]

6. L'assistance médicale dans les universités

Plusieurs étudiants sont atteints de maladies chroniques et d'autres à besoins spécifiques nécessitant une assistance médicale en urgence, pour cela on trouve les services de santé.

6.1 Un service de santé universitaire

Chaque université dispose d'un service de santé qui représente un lieu d'accueil, d'écoute et d'information santé. Il est composé d'une équipe pluridisciplinaire : médecin, gynécologue, psychiatre, psychologue, infirmière/infirmier, assistante sociale ; un agent d'Hygiène Sécurité Environnement (HSE). L'équipe pluridisciplinaire se tient à votre disposition pour tout conseil ou problème de santé et pour vous aider à mener à bien vos études dans les meilleures conditions physiques, psychiques et sociales. [11]

6.2 Agent d'Hygiène Sécurité Environnement (HSE)

Le chargé d'hygiène sécurité environnement (HSE) est un professionnel de la sécurité et de la prévention des risques et des maladies dans une entreprise. Il fait respecter les conditions de travail des salariés, s'assure de leur sûreté, du bon état des matériaux et du respect de l'environnement. La principale mission d'un chargé HSE est de veiller à la sécurité afin d'éviter les risques professionnels au sein de la société où il est employé. Il conçoit et présente des plans de prévention pour former le personnel aux normes de sécurité et de respect de l'environnement. Il veille au respect des consignes de sécurité et des conditions de travail.[12]

6.3 Exemples d'assistance médicale

- **Assistance médicale Service de diagnostic de santé** ; Cette assistance donne accès à différents types d'information médicale, que ce soit sur les médicaments et leur dosage, les maladies et leurs symptômes, les protocoles, etc. C'est une application généraliste qui couvre les besoins de l'étude clinique.
- **MedPics**, vous permettra de montrer des radiographies, IRM ou encore photos d'une lésion cutanée à d'autres médecins dans l'anonymat total des patients. L'objectif étant de partager la découverte de cas médicaux rares et faciliter les diagnostics ; destinée aux professionnels de santé.
- **Mon Coach douleur** ; elle permet le suivi au jour le jour par le patient de ses douleurs liées au cancer. Il peut ainsi décrire sa douleur (localisation, intensité, date, durée, type de douleur), les répercussions de celle-ci sur les activités quotidiennes

et l'humeur, noter la prise des traitements prescrits, générer une synthèse de l'ensemble des douleurs survenues sur une période donnée, ou encore gérer ses rendez-vous médicaux.

7. Utilisation de Chat en santé

Le chat est un système de messagerie instantanée qui se prononce “**Tchat**” en français permet d'échanger des messages textuels ou des fichiers (images, vidéo, son...) en temps réel entre plusieurs utilisateurs connectés à un même réseau.[13]

7.1 Usages de chat

Le chat est destiné à être utilisé soit pour :

7.1.1 Usages personnels

L'un des usages les plus répandus du tchat est la conversation entre amis. La majorité des réseaux sociaux proposent ainsi cette fonctionnalité. Sur Facebook par exemple, dans la version desktop, la barre de droite indique les “amis” actuellement connectés et avec qu'il est possible d'engager une discussion instantanée. Les sites de rencontre proposent également des fonctions de discussion en temps réel afin de favoriser l'échange entre leurs membres.[14]

7.1.2 Usages professionnels

De nos jours, le **chat** est également un outil au service des entreprises. Il est notamment utilisé pour le support et le conseil client avant et après-vente. Comme dans notre cas le chat est entre l'étudiant et le médecin qui strictement professionnel. [14]

7.2 Comment utiliser le chat

- ❖ L'utilisation de la messagerie instantané (chat) dans le domaine de santé renforce les relations entre les services de santé et leurs patients ; et la création d'une connexion en ligne entre ces deux parties est devenu indispensable. Le chat joue un grand rôle dans la maintenance de cette relation comme dans notre cas entre l'étudiant et le médecin qui va servir de gain de confort et de temps pour l'étudiant malade et d'un accompagnement humain pour l'étudiant où il aura une meilleure compréhension de ses problèmes de santé ainsi elle permet au médecin le suivi de l'état de son patient (étudiant) et établir des pré-diagnostic en fonction des informations transmises par ce dernier.

Conclusion

Le DSS est l'un des systèmes conçus pour aider les utilisateurs à prendre de meilleures décisions et fournir une assistance efficace et rapide surtout dans notre cas d'assistance médicale pour les étudiants à maladies chroniques mais dans le cas des étudiants à besoins spécifiques nous aurons besoin d'approfondir notre étude pour le développement de ce type d'assistance qui assez complexe.

Dans ce projet nous allons créer une application logicielle pour l'assistance médicale qui aiderait, comme cité, un étudiant avec une maladie chronique à communiquer avec un médecin, et aussi aider les établissements de santé à mieux le prendre en charge en cas de détresse. Nous allons créer ce système en multiplateforme, ce qui veut dire qu'il sera accessible de toute sorte de périphérique tant qu'il y'a connexion internet. Nous allons mieux vous parler sur ce terme « le multiplateforme » dans le prochain chapitre.

Chapitre II : Le multiplateforme

Introduction

Plusieurs opérateurs de technologies ont lancé sur le marché une multitude de systèmes d'exploitation pour faire fonctionner nos téléphones mobiles de nouvelle génération.

La diversité des plateformes présente sur le marché des Smartphones, rend le développement des applications mobiles assez difficile et très coûteux. Le développement multiplateforme pourrait être une bonne solution pour cette problématique. Nous allons commencer par définir le logiciel multiplateforme, puis présenter l'intérêt de développement multiplateforme et ses inconvénients ainsi que les différentes solutions multiplateformes.

1. Définition du logiciel multiplateforme

Le logiciel multiplateforme (également un logiciel multiplateforme ou un logiciel indépendant de la plate-forme) En anglais on parle souvent de « cross-platform » est un type d'application logicielle mis en œuvre sur plusieurs systèmes d'exploitation ou appareils, souvent appelés plates-formes. Une plate-forme désigne un système d'exploitation tel que Windows, Mac OS, Android ou iOS.[15]

Le logiciel multiplateforme peut être divisé en deux types ; l'une nécessite une construction ou une compilation individuelle pour chaque plate-forme qu'elle prend en charge, et l'autre peut être exécutée directement sur n'importe quelle plate-forme sans préparation spéciale, par exemple, un logiciel écrit dans un langage interprété ou un bytecode portable précompilé pour lequel les interprètes ou l'exécution des packages sont des composants communs ou standard de toutes les plates-formes.[16]

Le développement multiplateforme consiste simplement à développer une application qui peut fonctionner sur plusieurs plates-formes. Par exemple, une application multiplateforme peut fonctionner sur Iphone, samsung, HTC, ipad, BlackBerry etc.

Selon (CHEHADE, 2011), Un logiciel multiplateforme est un logiciel conçu pour fonctionner sur plusieurs plates-formes, c'est-à-dire le couple liant ordinateur et système d'exploitation. En anglais on parle souvent de « cross-Platform software » ou « Platform Independent software » ou encore de « multi-Platform software ». Donc, nous pouvons dire que le développement multiplateforme

peut être défini comme le procédé par lequel un développeur conçoit un logiciel pouvant être déployé sous différentes plateformes.[17]

(VANDERDONCKT, 2012), est parvenue à mettre en place un logiciel qui fonctionne sous trois plateformes (tablette et PC) utilisant le système Ios, son logiciel était chargé de faire de réservation de clients pour la visite d'un salon d'exposition.[17]

2. L'intérêt du logiciel multiplateforme

L'avantage d'une application logicielle ou d'un programme multiplateforme est que vous pouvez utiliser le même programme que vous soyez sur un PC Windows ou que vous vous connectiez depuis votre ordinateur portable ou votre smartphone. La suite d'applications Microsoft Office, qui comprend Word, Excel et PowerPoint, est disponible sur Windows, Mac OS, iOS (iPhone / iPad) et Android. Bien qu'il existe des différences basées sur le fonctionnement des plates-formes, vous aurez une expérience similaire au sein de l'application entre tous vos appareils.

Avoir une expérience similaire sur n'importe quelle plate-forme signifie qu'il y a une courbe d'apprentissage beaucoup plus petite, si elle existe même, donc vous serez plus productif et pourrez utiliser un produit logiciel que vous connaissez quel que soit le système d'exploitation ou l'appareil que vous utilisez choisir. De plus, vos fichiers peuvent être déplacés beaucoup plus facilement entre vos appareils afin que vous puissiez utiliser le logiciel avec n'importe quel appareil que vous avez avec vous à l'époque. Et il existe un moyen de synchroniser l'ensemble de votre travail sur tous vos appareils, en utilisant le cloud.[18]

En effet, le fait de pouvoir déployer un logiciel sur plusieurs environnements permet non seulement d'économiser du temps (temps passé à reproduire le code pour chaque plateforme cible) mais aussi d'économiser des ressources (coût de production). De plus, un gain non négligeable émane de cette notion de développement multiplateforme : la maintenance. Dans le domaine informatique, la maintenance constitue en moyenne près de 70% des coûts, avoir ainsi un seul logiciel à maintenir reviendrait considérablement moins cher que si on en avait plusieurs.[19]

3. Le développement natif et multiplateforme

En développement mobile on qualifie de natif, une application qui a été développée pour un système spécifique dans un langage unique, par système spécifique on peut citer Android (l'OS mobile propre à Google) couramment développé en Java, iOS (l'OS mobile pour Apple) développé soit en Objective-C ou plus récemment en Swift, langage exclusivement créé et maintenu par Apple pour ses applications mobiles, ou encore Windows Phone (l'OS mobile pour Microsoft).

Contrairement à cela, un développement multiplateforme va, comme son nom l'indique, essayer de s'orienter vers un support sur plusieurs plateformes parmi celles citées plus haut par exemple. A savoir que le développement multiplateforme connaît une croissance depuis ces dernières années, avec de nombreux outils et Framework qui ont été développés en ce sens. En effet, de nombreuses entreprises, sociétés, associations, services ... ou même des jeux voudront toucher un public de plus en plus mobile, et de plus en plus diversifié concernant les systèmes d'exploitation du smartphone, même si Android et iOS restent majoritaires parmi ces systèmes.[20]

4. Inconvénients du développement multiplateforme

➤ **Petit problème de performance :**

La performance est l'une des contraintes majeures d'une application mobile. Cela dépend de nombreux facteurs, mais en général, si vous comparez deux applications ayant les mêmes fonctionnalités, l'une native et l'autre multi-plateforme, vous remarquerez que la version native est légèrement plus rapide. Cependant, cette différence de performance est généralement faible et souvent pas très visible pour l'utilisateur non averti.

➤ **Caractéristiques spécifiques à la plate-forme :**

Les plates-formes IOS et Android ne communiquent pas de la même manière avec les appareils mobiles, ce qui oblige les développeurs à réécrire parfois deux scripts pour les mêmes actions. Certaines fonctionnalités sont présentes sur iOS et non sur Android et inversement.

➤ **Longue attente pour les nouvelles fonctionnalités :**

Chaque fois qu'une nouvelle fonctionnalité est introduite par Google ou Apple, il faut un certain temps pour mettre à jour les applications prenant en charge cette nouvelle fonctionnalité.

5. Les différentes solutions multiplateformes

Pour développer une application "multi-plateforme", de nouveaux frameworks et outils basés sur un seul langage ont commencé à émerger. Nous allons étudier les 3 possibilités majeures, à savoir le full web, une application accessible depuis votre navigateur sur mobile, le développement hybride et enfin le développement cross-platform entièrement réalisé à partir d'un langage unique.

5.1 Full web

Une application web mobile s'exécute dans le navigateur web du mobile. Cette solution implique d'utiliser des technologies Web standards comme HTML, CSS, JavaScript pour créer une application ainsi que lui donner un comportement et une apparence "natif". Cette solution facilitant le développement présente le désavantage de ne pas pouvoir tirer entièrement partie du support mobile et de certains de ses capteurs ; de plus en passant par un navigateur web le développement hérite de tous les problèmes du développement web : compatibilité sur tous les navigateurs, performances limitées.[20]

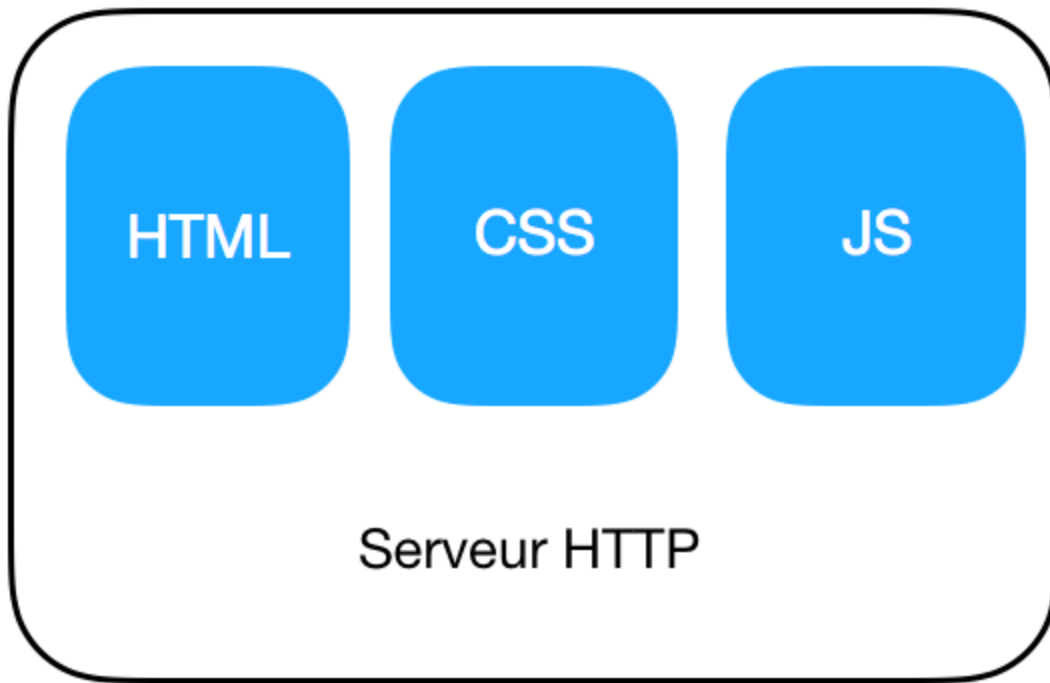


Figure 01 : architecture full web.

5.2 Applications Hybrides

Le principe est de fournir un moyen de délivrer des applications qui tournent localement avec des technologies web restituées via le moteur de rendu web, et tirant profit des capacités matérielles locales, comme le GPS, la caméra, etc. Cette solution est similaire à la création d'un site web, les problèmes restent les mêmes vu que ces frameworks utilisent les Webview du système sur lequel ils sont déployés. Néanmoins via ces technologies il est possible de développer des modules spécifiques pour proposer des fonctionnalités propres à chaque système.[21]

Parmi les frameworks existants on peut citer : Apache Cordova, phoneGap et Ionic (qui se base sur Apache Cordova)

- Cordova : est un framework développé par Apache qui va apporter une facilité de développement pour une application basique dont on voudra rapidement faire l'export sur les différents supports mobiles, c'est la solution si vous souhaitez

développer rapidement et à faible coût une application multiplateforme ne nécessitant pas une utilisation complexe mais pour laquelle vous aurez tout de même besoin d'accéder à certains éléments natifs du téléphone (GPS, caméra, ...), et PhoneGap pour la partie visuelle de votre application. [22]

- Ionic quant à lui est aussi un framework, basé sur Cordova concernant la partie UI de votre application, associé à la technologie AngularJS. C'est ce qu'on pourrait définir de surcouche à Cordova, il va vous permettre de développer plus facilement la partie interface utilisateur de votre application hybride tout en profitant des spécificités que confèrent Cordova.[23]

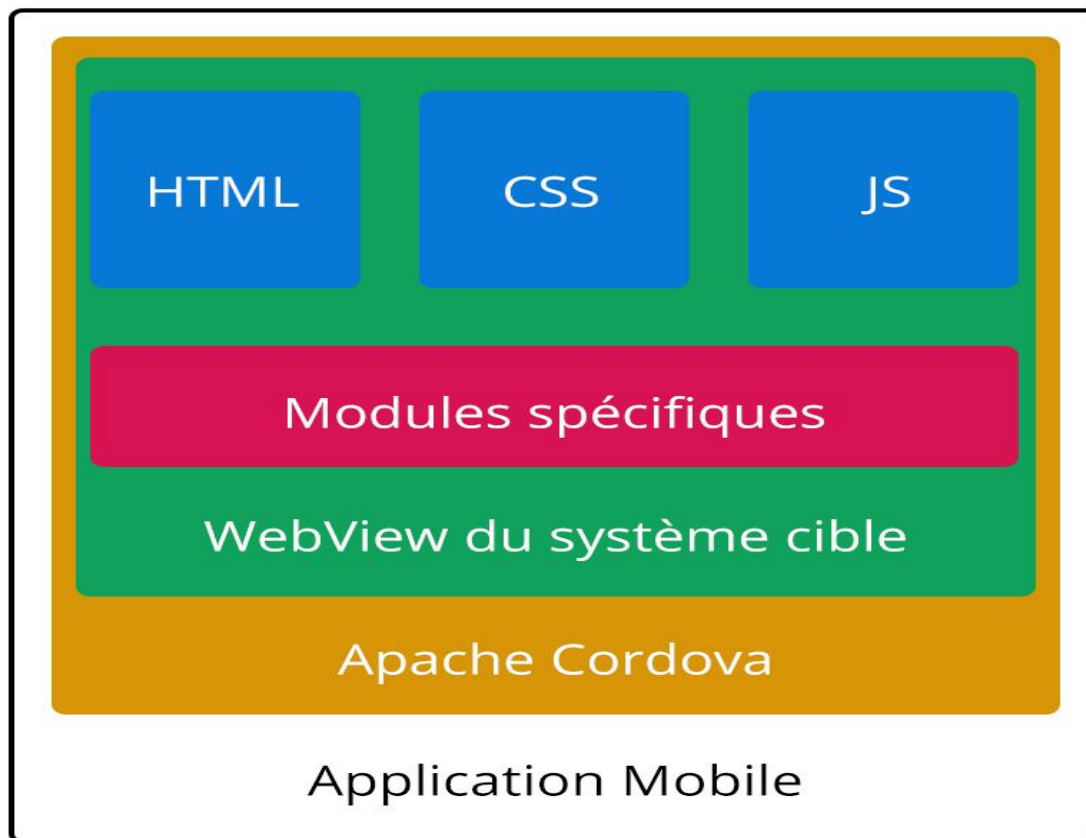


Figure 2 : application hybride.

5.3 Xamarin

Xamarin est une plate-forme open source racheté par Microsoft en 2016 pour développer des applications multiplateformes sans utiliser de webApps. Avec Xamarin votre application est entièrement développée dans le même langage, du C#, et sera disponible sur Android, iOS et Windows Phone, les éléments natifs sont entièrement gérés grâce au code en C# ; aussi il s'exécute dans un environnement géré qui offre des fonctionnalités telles que l'allocation de mémoire et la récupération de place.[24]

Ce modèle permet aux développeurs d'écrire toute leur logique métier dans un seul langage (ou de réutiliser le code d'application existant) mais d'obtenir des performances, un aspect et une convivialité natifs sur chaque plate-forme.[24]

6. Présentation de quelques frameworks dédiés au développement multi plateformes

L'application multiplateforme peut s'exécuter sur autant de plates-formes existantes que sur deux plates-formes. Des frameworks multiplateformes (tels que Flutter, NativeScript, Xamarin, Phonegap, Ionic et React Native) existent pour faciliter le développement multiplateforme.

Parmi ces frameworks nous allons nous concentrer sur les trois frameworks qui suivent : Unity, React-Native et flutter. Nous avons ciblé ces trois framework pour leurs communautés existantes ou grandissantes, leurs supports, leurs utilisations et leurs complexités d'utilisations.

6.1 Unity

Ce framework est principalement dédié à la conception de jeu vidéo, néanmoins il permet de créer des applications multi-plateformes aussi bien mobiles que bureautiques. L'utilisation d'Unity devient vite limitée lorsqu'il est question de créer une application qui n'est pas un jeu, l'IDE étant centré sur des concepts de « scène » et « assets » propres au Jeu vidéo.[25]

6.2 React-Native

Ce framework, récemment propulsé par Facebook, propose d'utiliser la simplicité d'utilisation de la technologie React qui permet de construire et accélérer le développement des applications natives sous **iOS et Android**, en utilisant les composants natifs de la plateforme. Il permet aux développeurs de créer des interfaces utilisateur web en JavaScript à partir de composants visuels réutilisables.[26]

6.3Flutter

C'est le framework de Google permettant de concevoir des applications multiplateformes pour Android et iOS. Ce framework est utilisé pour tout ce qui est interface utilisateur. Mais aujourd'hui Flutter se fait surtout connaître pour sa capacité à concevoir des applications natives multiplateforme pour Android et iOS (Windows/Mac/Linux sont également supportés).[27]

Conclusion

Le développement multiplateforme vous fait gagner du temps et de l'argent. Cela vous permet d'être très rapidement sur le marché tout en atteignant un grand nombre d'utilisateurs.

Cependant, vous devez choisir le cadre multiplateforme à adopter en fonction de l'objectif et de la complexité de votre application ; dans certaines situations extrêmes, vous pouvez envisager de faire référence à un développement d'applications natives.

Dans le prochain chapitre, nous aborderons les méthodes et langages utilisées pour l'analyse et conception de notre application.

Chapitre III : Analyse et Conception

Introduction

Le futur d'un logiciel dépend beaucoup de cette phase, elle nous permet d'éviter le développement d'une application non satisfaisante. Pour cela le client et le développeur doivent être en étroites relations, voire avoir un intermédiaire entre eux s'il le faut.

En effet nous présenterons l'approche que nous avons suivie pour concevoir notre application et ses différentes étapes en les illustrant avec quelques diagrammes conformes au langage UML.

1. Processus de développement de notre solution

Le processus de développement d'un logiciel décrit un ensemble d'activités ordonnées qui permet la spécification, la conception et la réalisation d'un logiciel et proposer une méthodologie pour gérer les changements tant dans le processus.[28]

Le développement de notre application respecte les étapes suivantes :

- ✓ Définition des besoins et des exigences.
- ✓ Analyse des besoins.
- ✓ Conception.
- ✓ Programmation.
- ✓ Tests sur le logiciel.

2. Spécification et analyse des besoins

La définition des besoins et des exigences correspond à l'étape dans laquelle nous discutons avec le client et les futurs utilisateurs afin de comprendre de quoi ils ont besoin : QUI doit pouvoir faire QUOI ?

Lors de cette étape, nous définissons également les demandes précises, telles que le respect de certaines normes graphiques, les temps de réponse, le matériel sur lesquels le logiciel devrait fonctionner, etc. [29]

2.1 Etablissement d'un cahier des charges

C'est un document qui décrit les fonctions que devra remplir la solution à l'issue du projet rédigé par le client, ainsi que ses spécifications techniques et les besoins auxquels elle devra répondre.[29]

2.2 Etude préliminaire

L'étude préliminaire est la toute première étape de notre processus de développement. Effectuer un premier extrait des besoins fonctionnels et opérationnels, en utilisant principalement le texte, ou des diagrammes très simples. Dans cette étape on doit définir les entités externes et leurs interactions avec le système.[30]

2.2.1 Besoins fonctionnels

L'application à réaliser doit offrir un ensemble de fonctionnalités afin de satisfaire les exigences de la totalité des utilisateurs.

2.2.1.1 Identification des acteurs

Un acteur est une entité externe (utilisateur humain, opérateur, dispositif matériel ou autre système) qui interagit avec le système étudié ; Il peut consulter ou modifier directement l'état du système, en émettant ou en recevant des messages susceptibles d'être porteurs de données. [31]

Les acteurs en interaction avec notre système sont :

- ✓ Partenaires d'alerte (Protection civile & Samu)
- ✓ Etudiant
- ✓ Agent-HSE-Université.
- ✓ Médecin.
- **Etudiant** : son rôle est :
 - Inscription à l'application
 - Authentification.
 - Demande d'assistance.

- Alerter le partenaire d'alerte.
- **Partenaires d'alerte** : représente l'un des services d'évacuation comme dans notre cas le SAMU (service d'aide médicale urgente) et la Protection Civile.
Son rôle est :
 - Authentification.
 - Traitement d'alertes.
 - Envoie des secours (contacter les ambulanciers).
- **Agent -HSE-Université** :
Son rôle est :
 - Authentification.
 - Gestion de liste étudiants.
 - Ajouter un médecin assistant.
 - Garder L'historique des Alertes.
- **Médecin** :
 - Offrir l'assistance à l'étudiant (proposer des solutions par rapport à son état de santé et répondre aux questions de l'étudiant).

2.2.1.2 Elaboration du diagramme de contexte

Ce diagramme exprime l'environnement du système dans une situation donnée, il permet donc de définir les frontières et les acteurs de système en cours d'étude ; Il répond à la question :

« Quels sont les acteurs et éléments environnants au système ? » [32]

Les acteurs sont : Partenaire d'alerte, Agent-HSE-Université, Etudiant, Médecin.

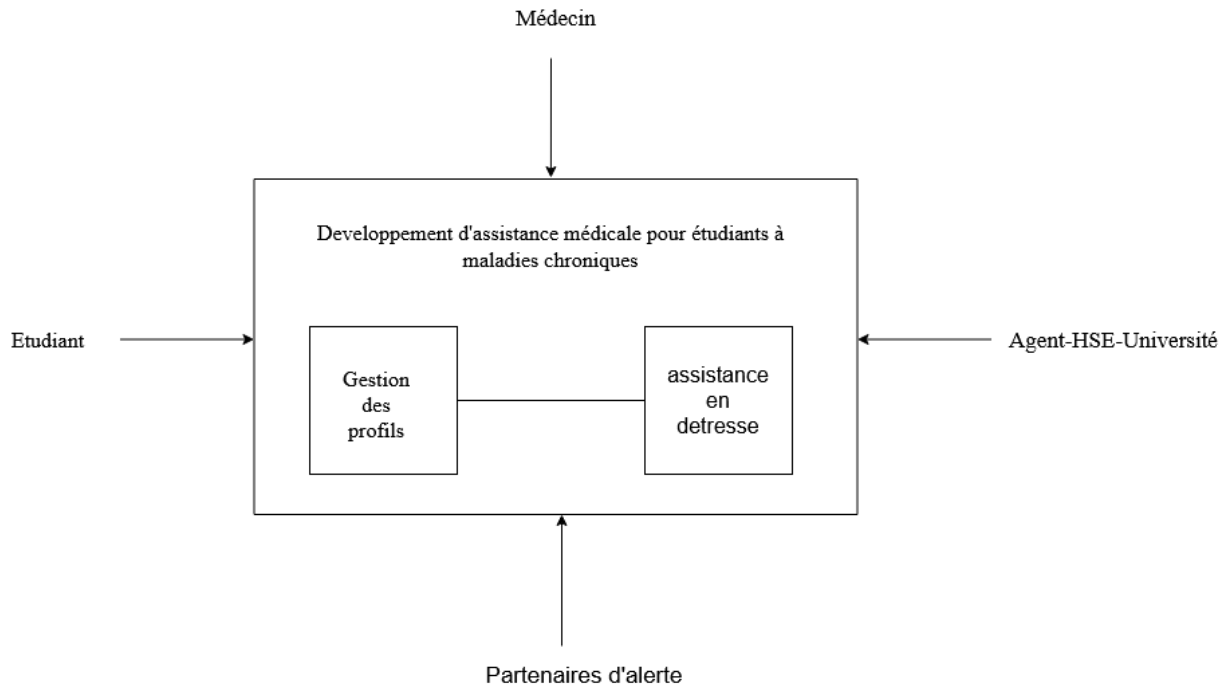


Figure 03 : diagramme de contexte.

2.2.1.3 Elaboration de diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre.[33]

Dans un diagramme de cas d'utilisation, il existe des acteurs qui interagissent avec des cas d'utilisation. Les cas d'utilisation permettent de structurer les besoins des utilisateurs et les objectifs du système. Une fois identifiés et structurés, ces besoins permettent d'identifier les fonctionnalités principales ou critiques du système.[33]

❖ Description des cas d'utilisation :

Nous présentons dans le tableau ci-dessous les différentes tâches qu'effectuent les acteurs de notre système ainsi leurs scénarios.

Tableau1 : tableau des tâches et scénarios.

Acteur	Tache	Scénario
Etudiant	T1 : accéder à l'application	-télécharger et installer l'application.
	T2 : demande d'inscription	-cliquer sur le bouton s'inscrire et remplir le formulaire.
	T3 : s'authentifier.	- saisir le nom d'utilisateur et mot de passe.
	T4 : consulter listes de maladies.	- cliquer sur le bouton assistance.
	T5 : alerter le service hospitalier.	- cliquer sur le bouton alerter.
	T6 : discuter avec un médecin.	-cliquez sur le bouton contacter un médecin et ouvrir la discussion.
	T7 : consulter le profil.	-Cliquer sur le bouton profil.
	T8 : se déconnecter.	- cliquer sur le bouton déconnexion.

Agent-HSE-Université	<p>T9 : s'authentifier.</p> <p>T10 : valider l'inscription de l'étudiant.</p> <p>T11 : vérifier la liste des étudiants.</p> <p>T12 : ajouter un étudiant à la liste.</p> <p>T13 : supprimer un étudiant dans la liste des étudiants.</p> <p>T14 : Vérifier la liste des Alertes Validées.</p> <p>T15 : Afficher les détails d'une alerte validée.</p> <p>T16 : Afficher la liste des Médecins.</p> <p>T17 : Afficher les informations d'un médecin</p> <p>T18 : Ajouter un médecin assistant.</p>	<p>-Saisir le nom d'utilisateur et mot de passe.</p> <p>- consulter les demandes d'inscription.</p> <p>- Valider les demandes d'inscription.</p> <p>- Cliquer sur le bouton liste étudiants pour accéder à la Liste.</p> <p>- Cliquer sur un étudiant pour modifier son profil ou le supprimer.</p> <p>- consulter les alertes validées.</p> <p>- Cliquer sur le bouton Liste médecins pour consulter la liste.</p> <p>- Cliquer sur le bouton ajouter médecin et remplir le formulaire affiché ensuite ; le valider.</p>
Partenaires d'alerte	<p>T19 : Consulter la liste des Alertes.</p> <p>T19 : Traitement d'une l'alerte.</p> <p>T20 : Envoyer les secours</p>	<p>-Réception d'une liste des alertes effectuées par les étudiants ou les médecins</p> <p>-envoyer un Sms d'urgence aux ambulanciers</p>

	T21 : Ajouter un Ambulancier.	
Médecin	<p>T22 : Se connecter.</p> <p>T23 : Offrir l'assistance à l'étudiant a besoin spécifique.</p> <p>T24 : envoie d'un message textuel d'alerte aux partenaires d'alertes.</p>	<p>-Saisir le nom d'utilisateur et mot de passe.</p> <p>-Discuter avec l'étudiant une fois dans l'interface discussion.</p> <p>-Remplir le formulaire d'alerte.</p>

❖ Diagramme de cas d'utilisation :

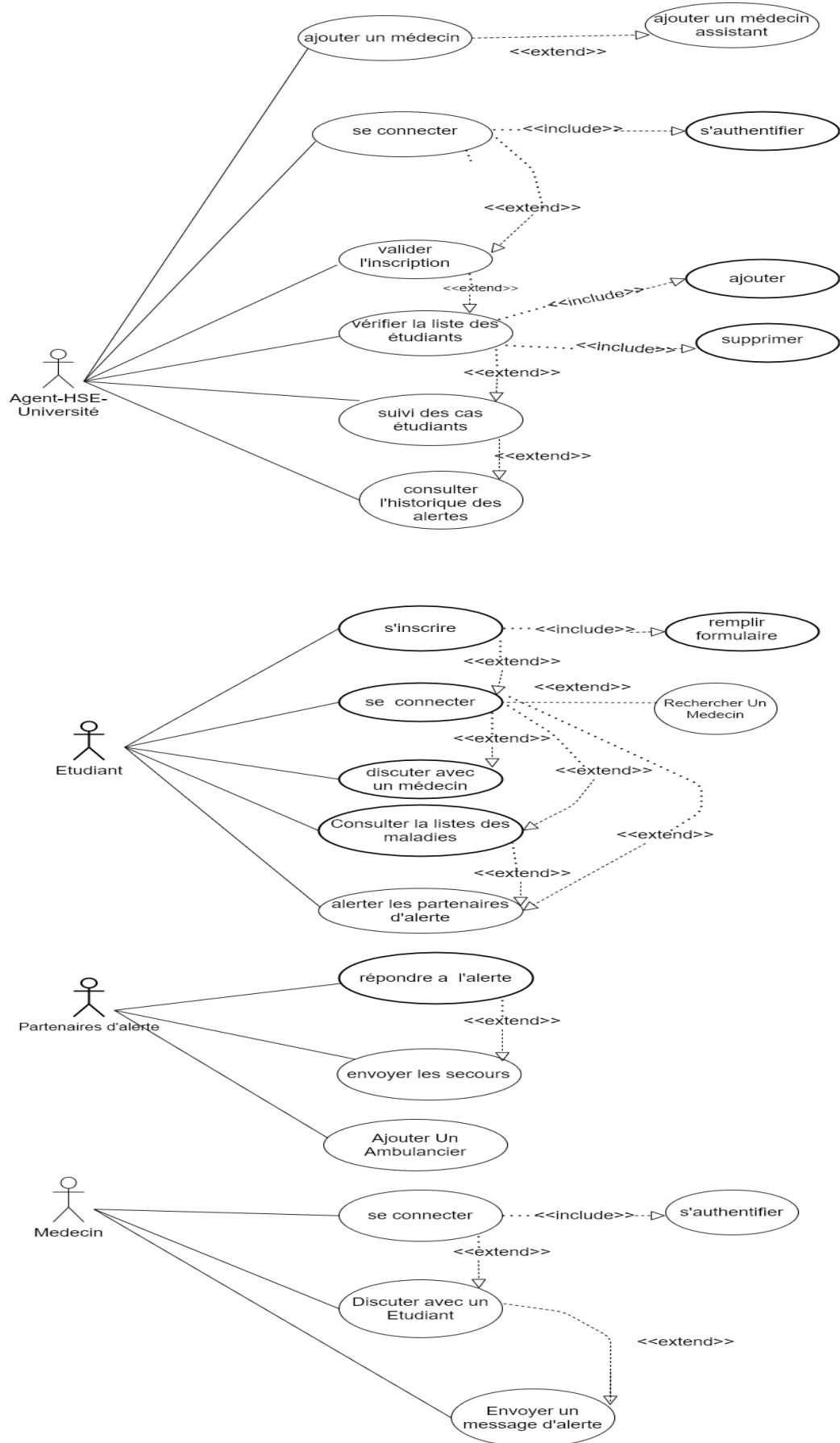


Figure 04 : diagramme de cas d'utilisation

2.2.2 Besoins non fonctionnels

A part les besoins fondamentaux, notre système doit répondre à plusieurs critères par exemple l'application devra être extensible afin de permettre d'ajouter de nouvelles fonctionnalités ou modifiés d'autres avec un coût et un délai raisonnable ; ainsi un niveau de sécurité fiable qui permet l'accès des utilisateurs à la plateforme et la protection de leur données partagées ; l'application doit être rapide en terme de réponse et facile à utiliser pour les utilisateurs (les interfaces sont simples et adaptés au utilisateur).

2.3 La phase de conception

La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation. Il en découle la description des bases de données éventuelles à créer et les programmes à écrire et la manière dont tout cela va être intégré.[34]

2.3.1 Conception préliminaire

Notre solution est une application concernant les étudiants à maladies chroniques dans un cas d'urgence ou de détresse nécessitant une intervention de SAMU (Service d'Aide Médicale Urgente) ou de la Protection Civile qui sont considérés comme étant des partenaires d'alerte afin d'évacuer l'étudiant qui a déclenché une alerte qui est un service offert par l'application téléchargée. Cette dernière récupère la géolocalisation de l'étudiant et la transmet vers la base de données qui à son tour la transmet vers les partenaires d'alertes. Par la suite ces derniers vont s'occuper de contacter des ambulanciers en envoyant un SMS d'urgence avec le nom et le cas critique de l'étudiant. Une fois cette alerte est effectuée l'application va exécuter automatiquement un envoi d'SMS d'urgence aux parents de l'étudiant évacué.

L'étudiant aussi peut discuter avec un médecin afin de partager les informations sur son état actuel et donner l'aide qu'il faut grâce au système de chat.

2.3.1.1 Architecture logicielle du système

L'architecture d'un système logiciel fonctionne comme un plan pour le système et le projet de développement, établissant les tâches nécessaires à exécuter par les équipes de

conception. Décrit comment un système informatique doit être conçu de manière répondre aux spécifications et réduit la complexité du système abordé en le structurant en composants logiciels.[35]

❖ **L'architecture globale de notre solution :**

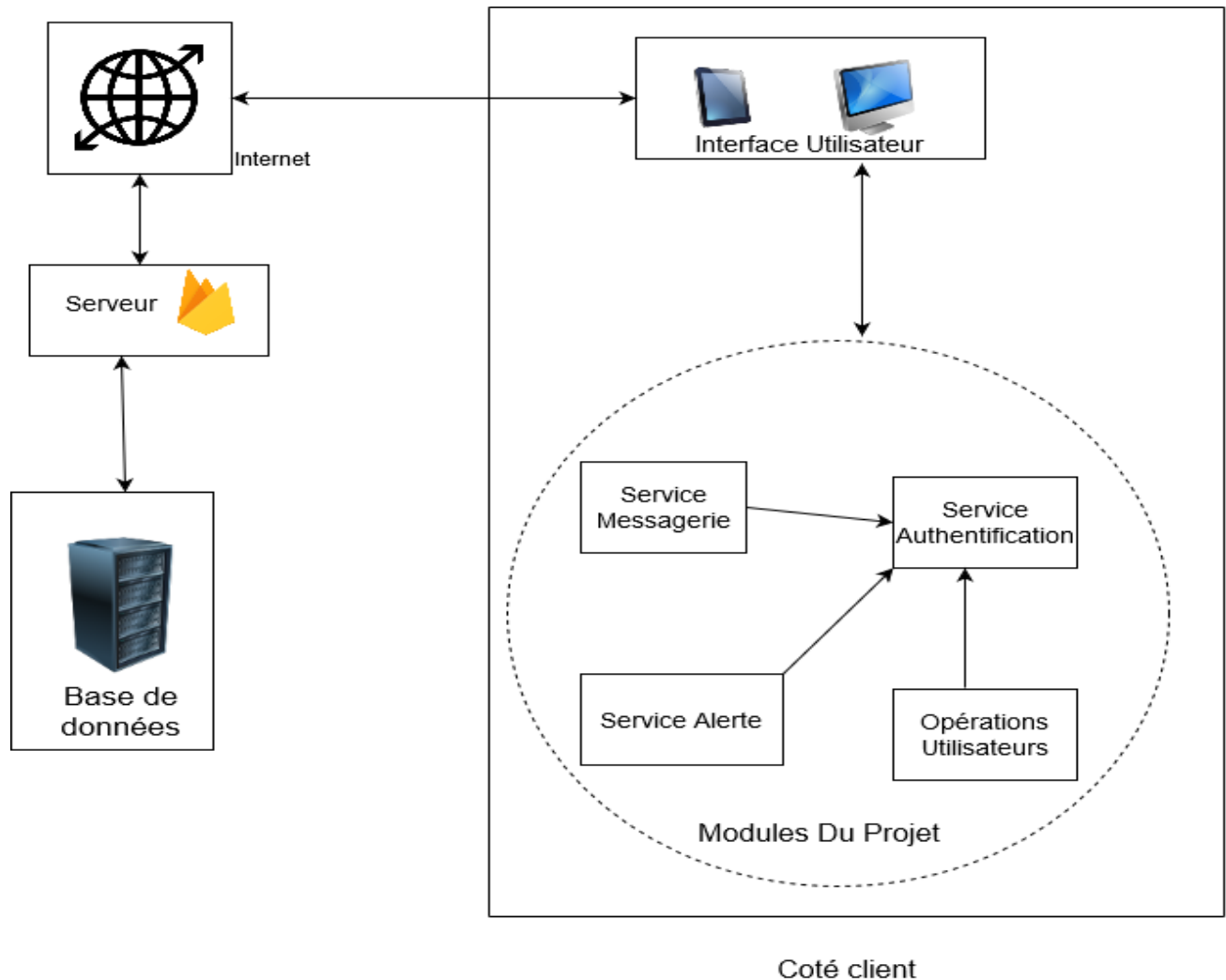


Figure 05 : architecture globale de système.

2.3.2 La vue logique de notre cas d'étude

Constitue la principale description architecturale d'un système informatique et beaucoup de petits projets se contentent de cette seule vue. Cette vue décrit, de façon statique et

dynamique, le système en termes d'objets et de classes et permet d'identifier les différents éléments et mécanismes du système à réaliser.

La vue logique est représentée, principalement, par des diagrammes statiques de classes et d'objets enrichis de descriptions dynamiques : diagrammes d'activités, de séquence, diagrammes de communication ou d'états-transitions. [36]

2.3.2.1 Diagramme de classe pour notre cas d'étude

Les diagrammes de classes représentent les structures statiques d'un système, y compris ses classes, attributs, opérations et objets. Un diagramme de classes fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques. Il est nécessaire lors de la modélisation objet d'un système.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.[37]

Nous présentons ci-dessous le diagramme de classe de notre cas d'étude :

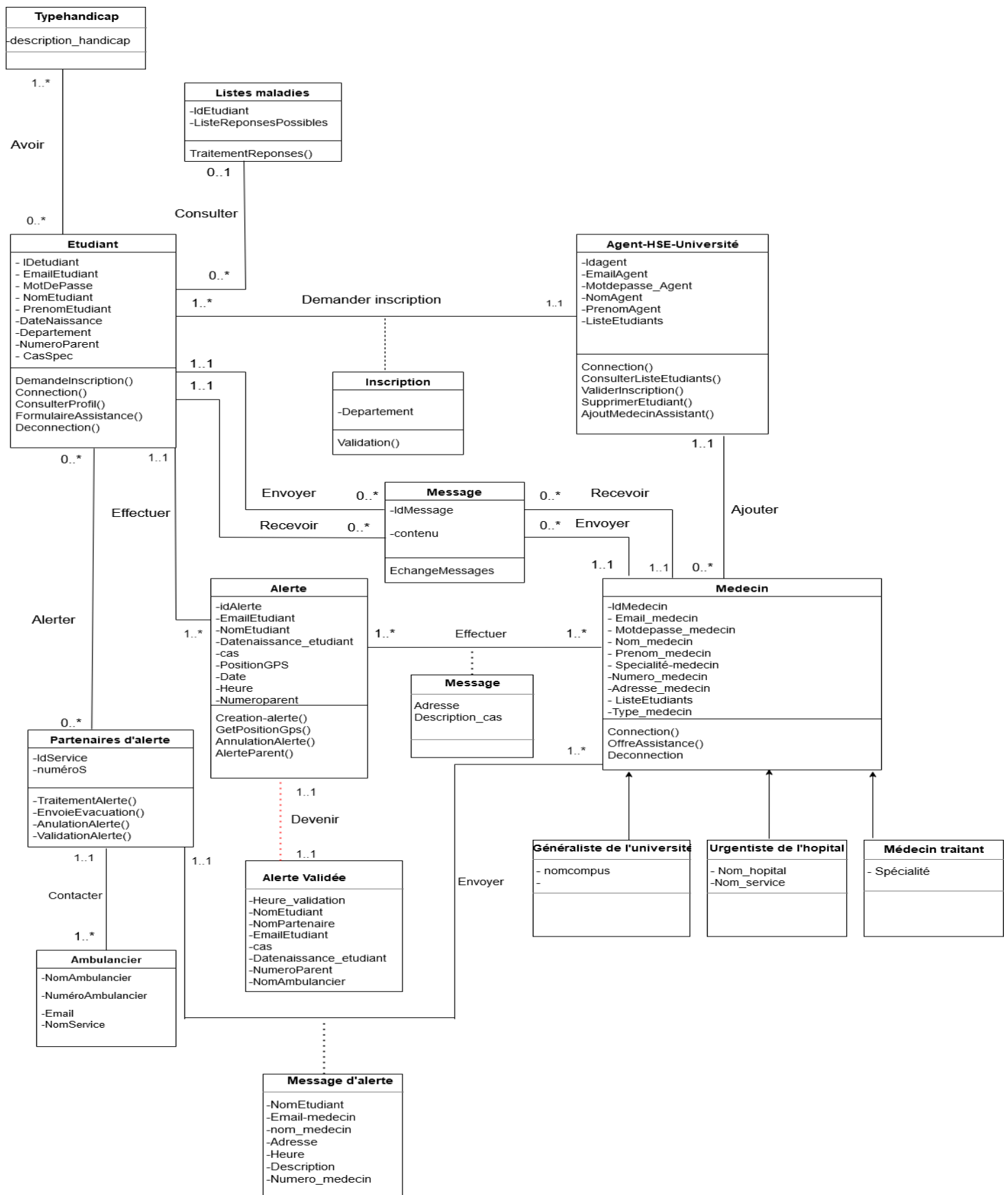
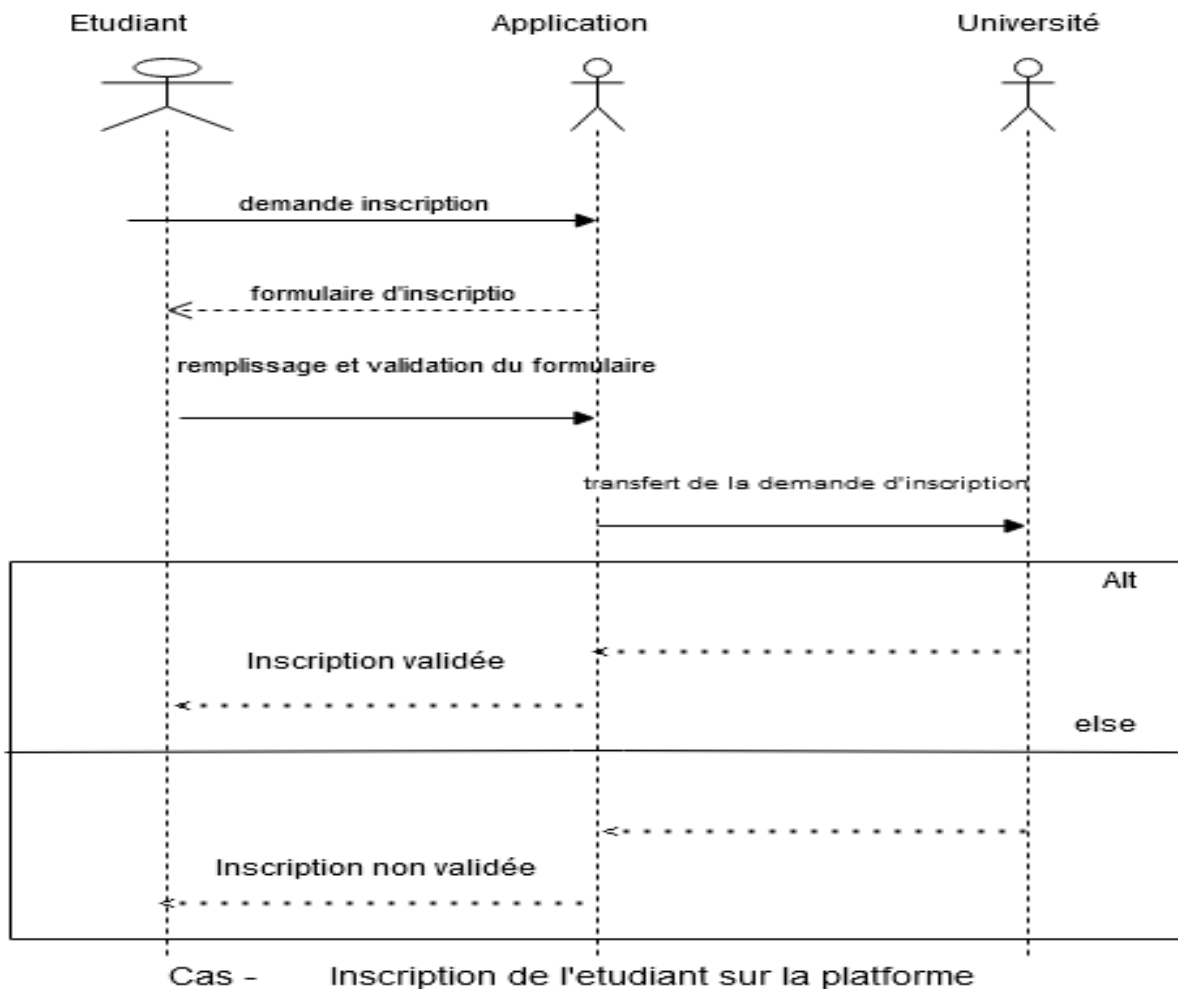


Figure 06 : Diagramme de classe.

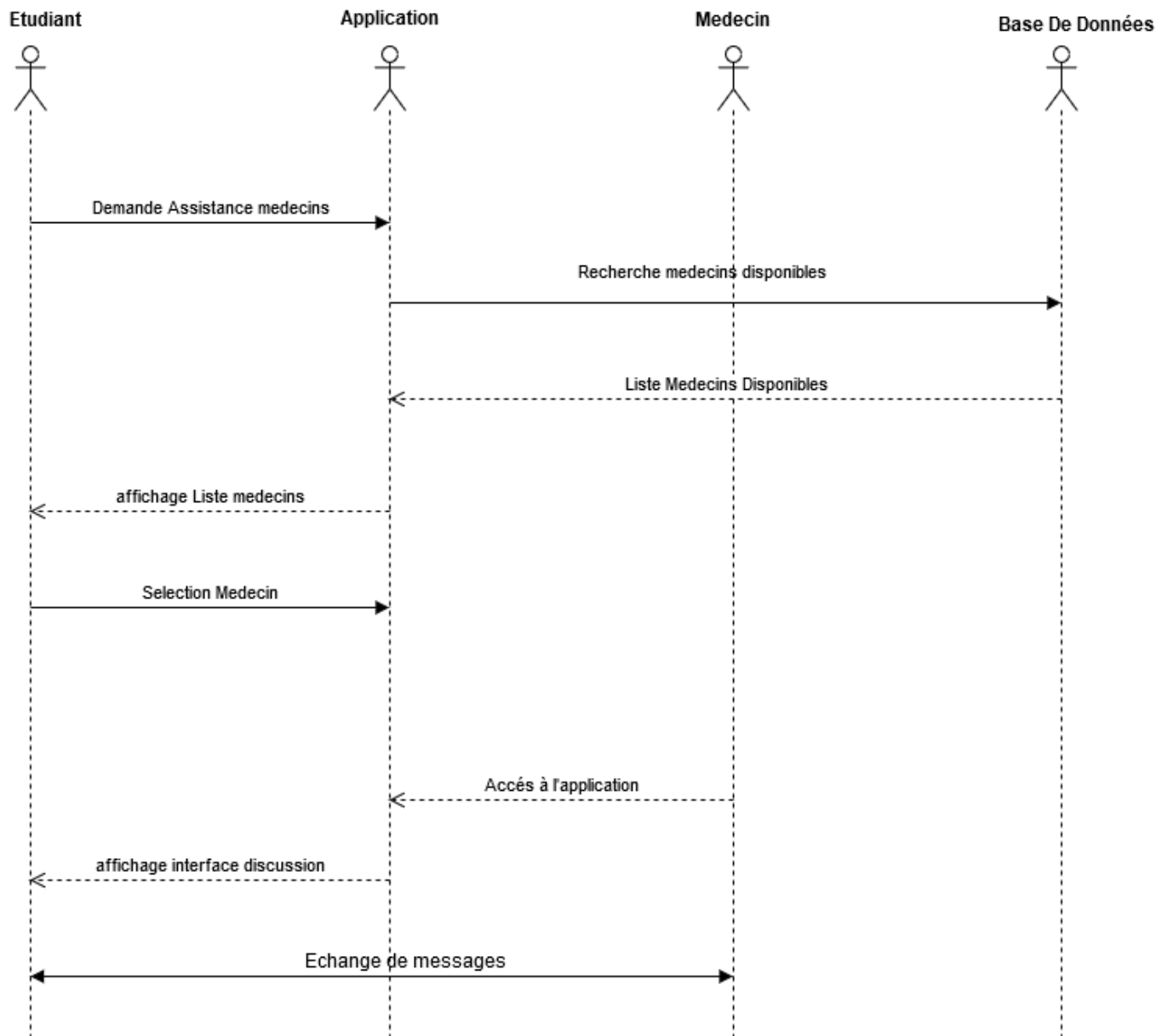
2.3.2.2 Diagramme de séquence de notre cas d'étude

Les diagrammes de séquence, également appelés diagrammes d'événements, illustrent la façon dont des processus interagissent entre eux en montrant les appels entre différents objets dans une séquence. Ces diagrammes ont deux dimensions : verticale et horizontale. Les lignes verticales montrent la séquence de messages et d'appels dans l'ordre chronologique, et les éléments horizontaux montrent les instances d'objets où les messages sont relayés.[38]



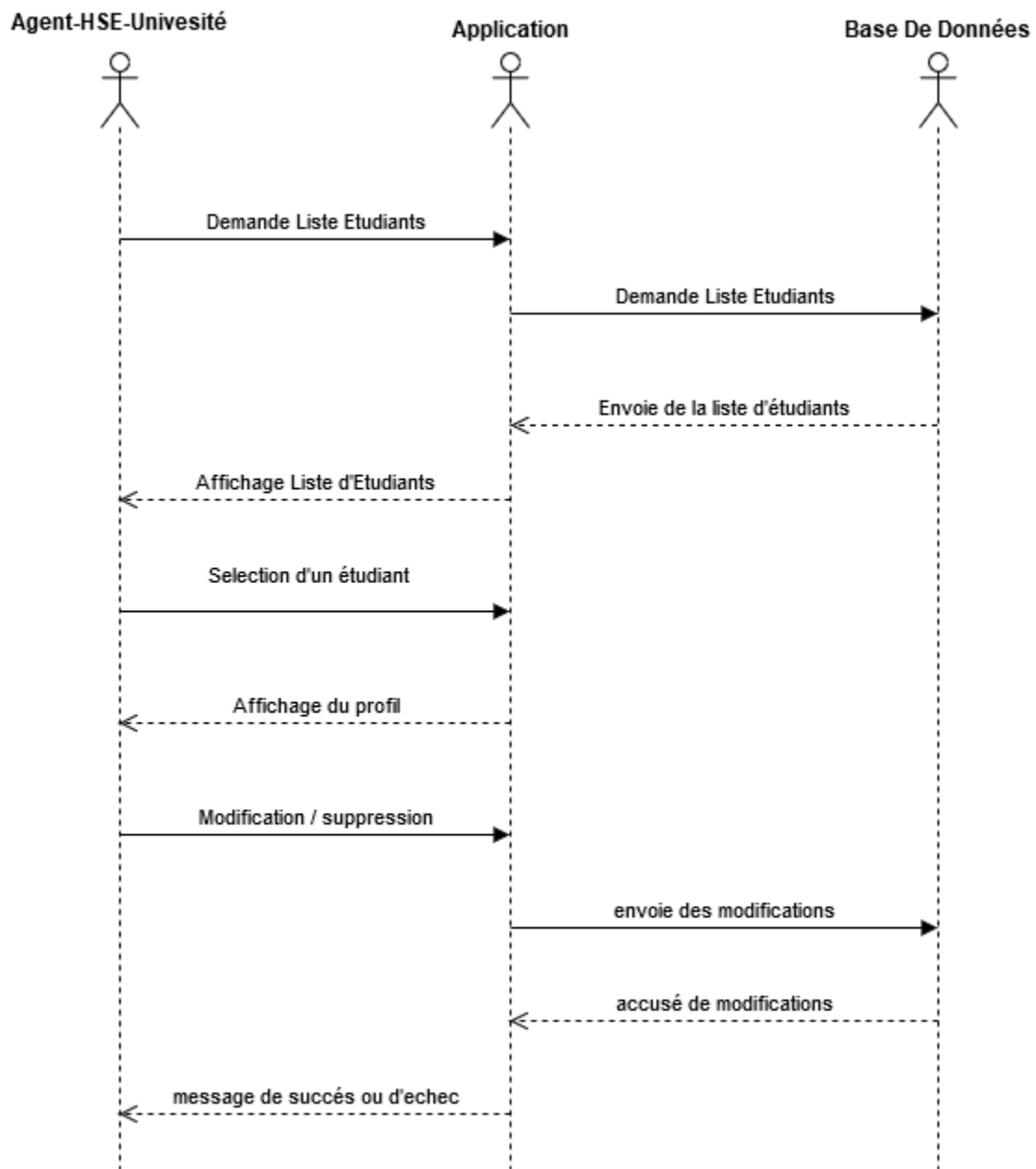
Nous présentons ci-dessous quelques diagrammes de séquences de notre cas d'étude :

Figure 09 : Diagramme de séquence cas : alerter le Partenaire d'alerte.



Cas - Demande Assistance Medecin

Figure 10 : Diagramme de séquence cas : demande assistance médecin.



Cas - Manipulation sur la liste d'étudiants

Figure 11 : Diagramme de séquence cas : manipulation sur la liste d'étudiants.

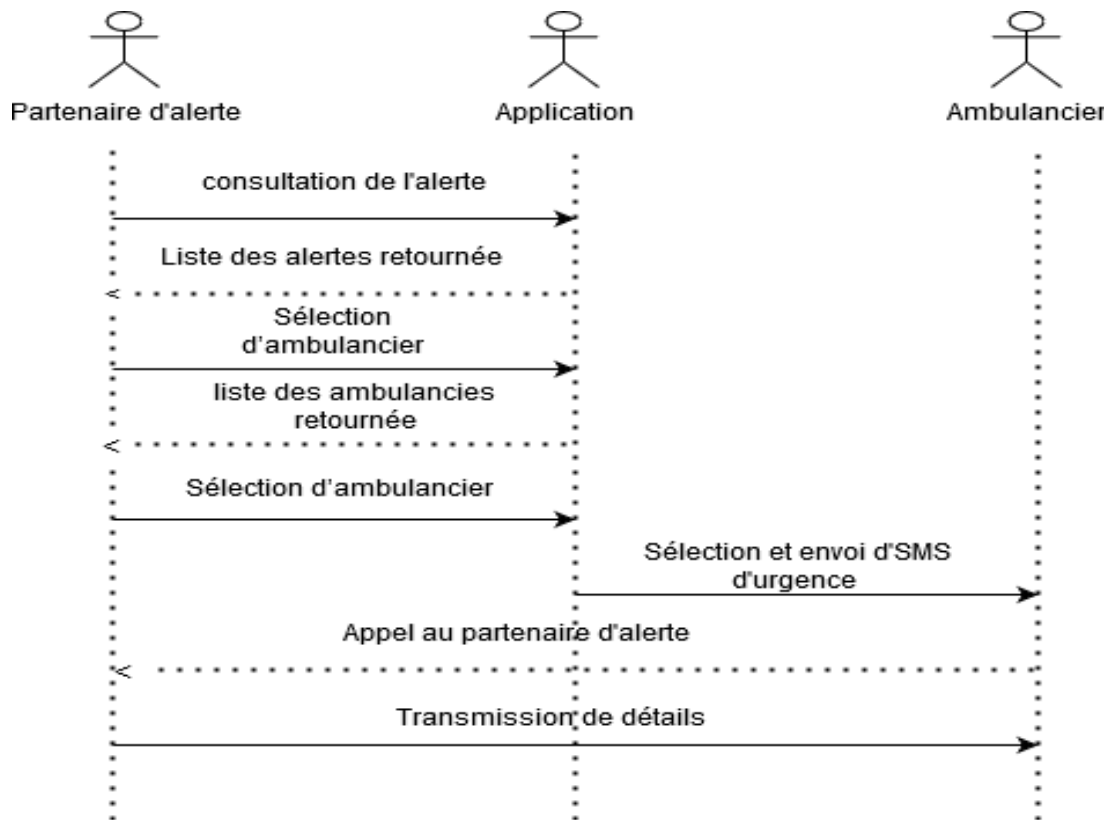


Figure 12 : Diagramme de séquence cas : Contacter un ambulancier.

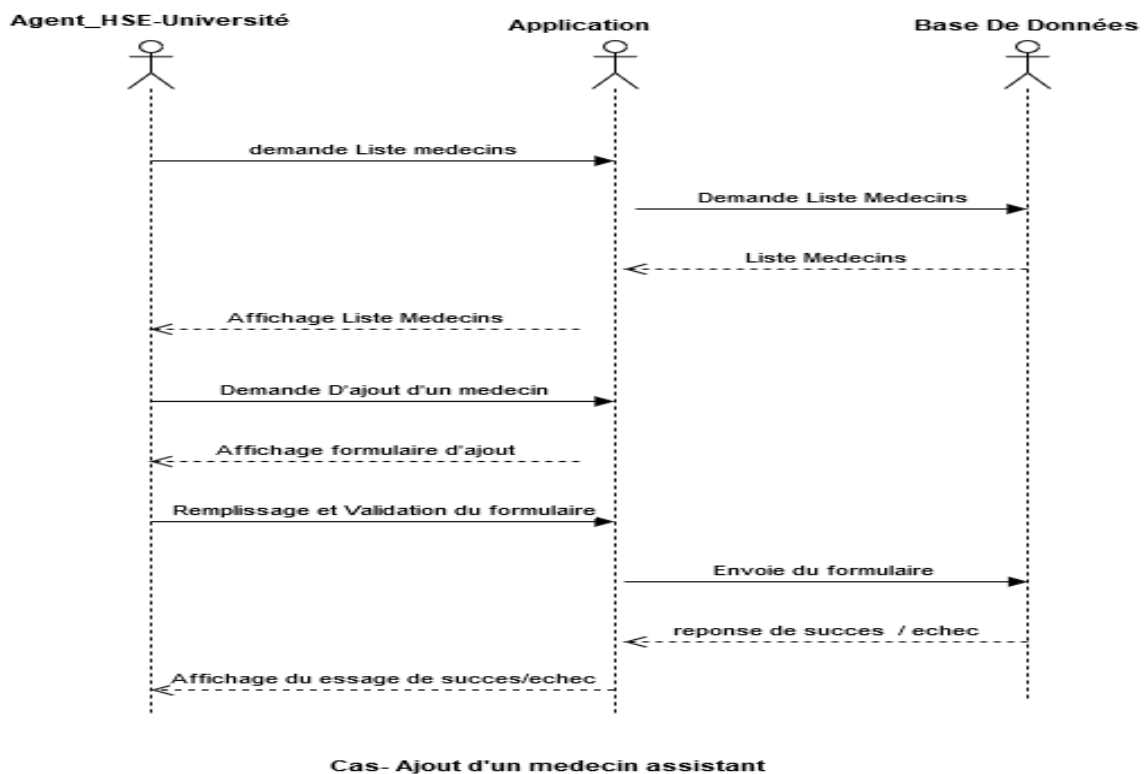
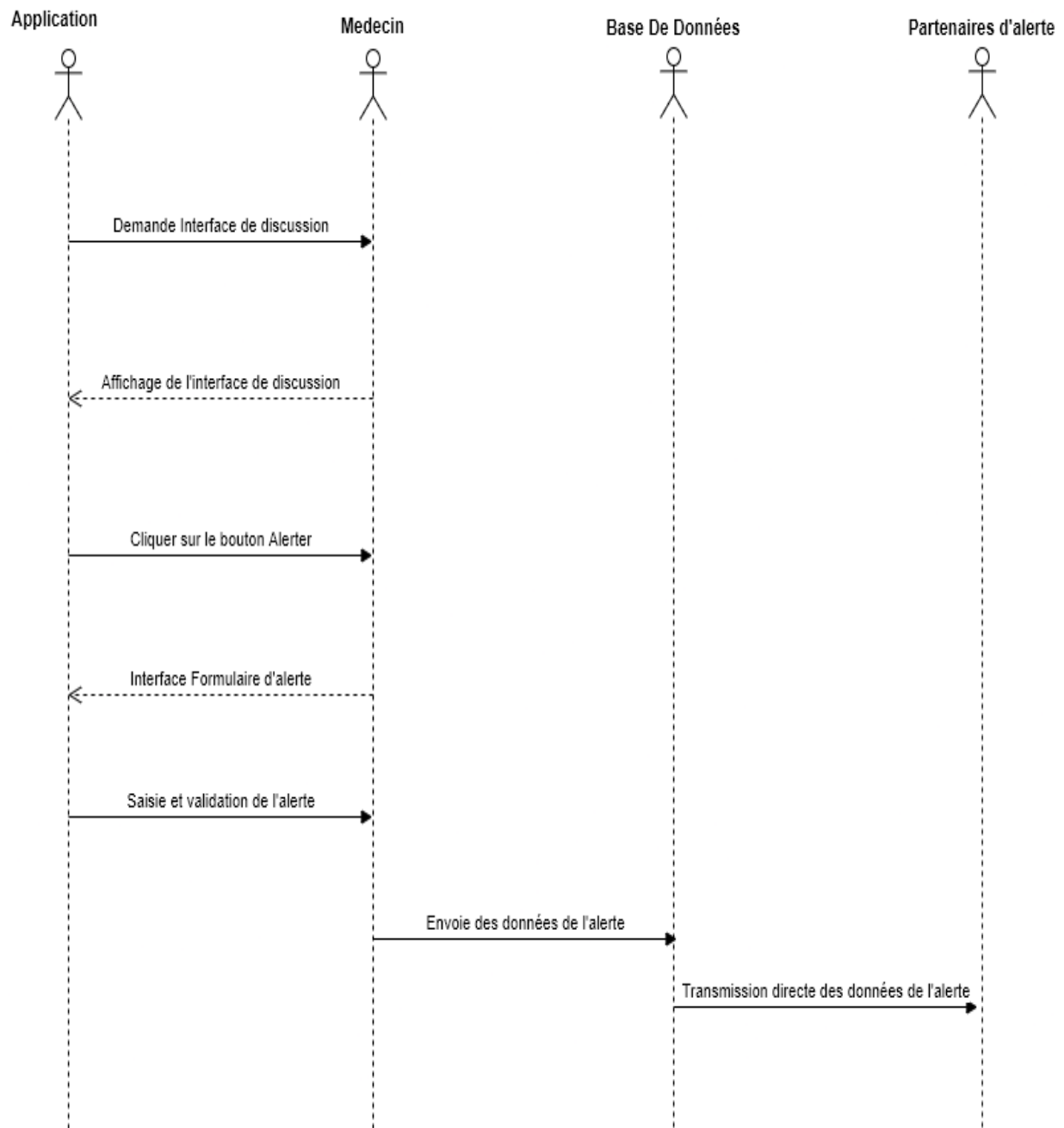


Figure 13 : Diagramme de séquence cas : ajouter un médecin.



Cas - Envoi d'un message d'alerte par le médecin

Figure 14 : Diagramme de séquence : Envoi de message d'alerte par le médecin.

Conclusion

Dans ce chapitre, nous avons présenté l'analyse et la conception de notre solution, en élaborons quelques diagrammes utilisant le formalisme UML (**Unified Modeling Language**). Nous avons défini les besoins de notre client, et détaillé les besoins fonctionnelles et non fonctionnelles que doit contenir notre solution. Et nous avons tracé deux diagrammes UML à savoir le diagramme de contexte et le diagramme de cas d'utilisations.

Au niveau de la conception, nous avons défini l'architecture globale et détaillée de notre cas d'étude, nous avons élaboré les diagrammes de classes participantes des cas d'utilisation de notre système et quelques diagrammes de séquences. Cela a permis d'identifier les différents concepts du domaine : classes, attributs, opérations, etc.

Le chapitre suivant est consacré à la phase de réalisation de notre application, à la présentation des différentes interfaces, ainsi que les outils utilisés durant sa réalisation.

Chapitre IV : Implémentation et réalisation

Introduction

L'implémentation est la phase la plus importante après celle de la conception. Le choix des outils de développement influence énormément sur le coût en temps de programmation, ainsi que sur la flexibilité du produit à réaliser. Cette phase consiste à transformer le modèle conceptuel établi précédemment en des composants logiciels formant notre système.

Dans ce chapitre, nous allons commencer par la description de l'environnement de travail puis à dégager et élaborer les composants de notre système.

1. Les langages de programmation et outils utilisés

1.1 Flutter

Flutter est un kit de développement de logiciel d'interface utilisateur open source créé par Google. Il permet de développer des applications pour Android, iOS, Linux, Mac, Windows, Google Fuchsia et le Web à partir d'une seule base de code.

La première version de Flutter était connue sous le nom de code 'Sky' et fonctionnait sur le système d'exploitation Android. Flutter a été dévoilé lors du Summit des développeurs Dart 2015.[39]



Flutter se positionne différemment par rapport aux frameworks multiplateformes existants, notamment avec l'obtention d'une application native possédant une interface utilisateur n'utilisant pas de composants natifs, ni de webview. En effet tous les composants de votre interface utilisateur sont construits par le moteur graphique de Flutter écrit en C++ et dessiné par les équipes de Google, en étroite relation avec les équipes de Material Design si vous souhaitez utiliser le design material.[40]

Flutter pourra se lancer sur un marché autre que le mobile tout en gardant **un seul code source** et une application toujours native, c'est-à-dire que votre application Flutter web sera compilée JavaScript et conservera le même rendu que votre application mobile. Il ne

vous restera qu'à adapter votre interface utilisateur pour avoir un rendu correspondant à une application web.

Le fonctionnement est assez impressionnant et aucun changement n'est nécessaire au niveau de votre code pour déployer votre application sur toutes les plateformes. Flutter veut donc se positionner comme un vrai framework multiplateforme.

1.2 Dart

Dart est un langage orienté Web open source, sous licence BSD, proposé par Google, très impliqué dans le développement Web. Il s'agit d'un langage orienté objet simple, facile à prendre en main. Il s'exécute dans une Machine Virtuelle (VM) aussi bien du côté client (sur le navigateur) que du côté serveur. Il a pour objectif de détrôner le langage JavaScript, qui est le plus utilisé. Cependant, pour l'instant Dart n'est supporté par aucun navigateur. C'est pour cette raison que Google propose de le compiler en JavaScript pour qu'il puisse être utilisé. De plus, bien qu'étant très utilisé, JavaScript comporte de nombreux défauts qui rendent les programmes difficiles à maintenir et vulnérables. [41]



1.3 Firebase

Firebase est le nom d'une plateforme mobile de Google qui facilite la création de backend à la fois scalable et performant. En d'autres termes, il s'agit d'une plateforme qui permet de développer rapidement des applications pour mobile et pour le web.[42]

L'objectif de la création de firebase.google.com en 2011 par James Tamplin et Andrew Lee est d'éviter aux professionnels et aux particuliers de s'engager dans un processus complexe de création et de maintenance d'une architecture serveur.[42]



De plus, la plateforme peut être exploitée par plusieurs utilisateurs en même temps sans connaître un quelconque bug. La praticité est également au rendez-vous grâce à ses fonctionnalités intuitives. Depuis le rachat de la plateforme par Google en 2014, Firebase SDKs a connu de nombreuses améliorations et n'a de cesse de satisfaire ses utilisateurs.[42]

1.4 Cloud Firestore

Cloud Firestore est une base de données NoSQL flexible et évolutive pour le développement mobile, Web et serveur de Firebase et de Google Cloud Platform. Comme Firebase Realtime Database, il maintient vos données synchronisées entre les applications clientes via des écouteurs en temps réel et offre une prise en charge hors ligne pour le mobile et le Web afin que vous puissiez créer des applications réactives qui fonctionnent indépendamment de la latence du réseau ou de la connectivité Internet. Cloud Firestore offre également une intégration transparente avec d'autres produits Firebase et Google Cloud Platform, y compris Cloud Functions.[43]

Les applications iOS, Android et Web peuvent accéder directement via des SDK natifs. Cloud Firestore est également disponible dans les SDK natifs Node.js, Java, Python, Unity, C++ et Go, en plus des API REST et RPC.[43]

1.5 Base de données à temps réel firebase (Firebase Real Time Database)

Une base de données en temps réel est un système de base de données qui utilise un traitement en temps réel pour gérer les charges de travail dont l'état change constamment. Cela diffère des bases de données traditionnelles contenant des données persistantes, généralement non affectées par le temps.[44]

La base de données temps réel de firebase est une base de données hébergée dans le cloud. Les données sont stockées au format JSON et synchronisées en temps réel avec chaque client connecté. Lorsque vous créez des applications multiplateformes avec leurs SDK iOS, Android et JavaScript, tous vos clients partagent une instance de base de données en temps réel et reçoivent automatiquement des mises à jour avec les données les plus récentes.[44]

1.6 Git

Git est de loin le système de contrôle de version le plus largement utilisé aujourd'hui. Git est un projet open source avancé, qui est activement maintenu. À l'origine, il a été développé en 2005 par Linus Torvalds, le créateur bien connu du noyau du système d'exploitation Linux. De plus en plus de projets de développement reposent sur Git pour le contrôle de version, y compris des projets commerciaux et en open source. Les développeurs qui utilisent Git sont bien représentés dans le pool de talents disponible, et la solution fonctionne bien sur une vaste gamme de systèmes d'exploitation et d'environnements de développement intégrés (IDE).[45]



Par sa structure décentralisée, Git illustre parfaitement ce qu'est un système de contrôle de version décentralisé (DVCS). Plutôt que de consacrer un seul emplacement pour l'historique complet des versions du logiciel comme c'était souvent le cas dans les systèmes de contrôle de version ayant fait leur temps, comme CVS (système de version décentralisé) et Subversion (également connu sous le nom de SVN), dans Git, chaque copie de travail du code est également un dépôt qui contient l'historique complet de tous les changements.[46]

En plus d'être décentralisé, Git a été conçu pour répondre à trois objectifs : performance, sécurité et flexibilité.

1.7 GitKraken

Est un client Git (outil de gestion de versions surtout utilisé par les développeurs) assez récent, compatible avec macOS, Linux et Windows et gratuit pour un usage non-commercial. Il a reçu sa première mise à jour majeure avec une 2.0 qui se concentre en premier lieu sur les performances. Tout est plus rapide dans cette version et certaines tâches sont même deux fois plus rapides.[47]

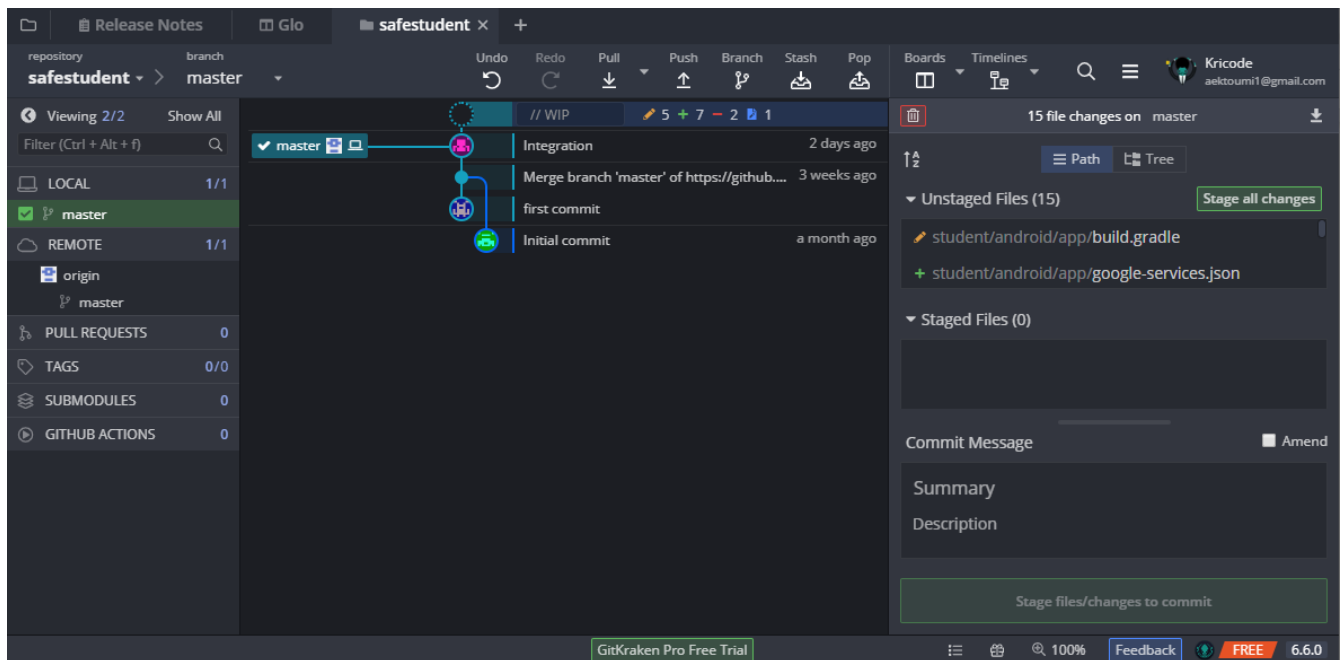


Figure 15: Interface gitkraken de notre projet.

1.8 Visual Studio Code

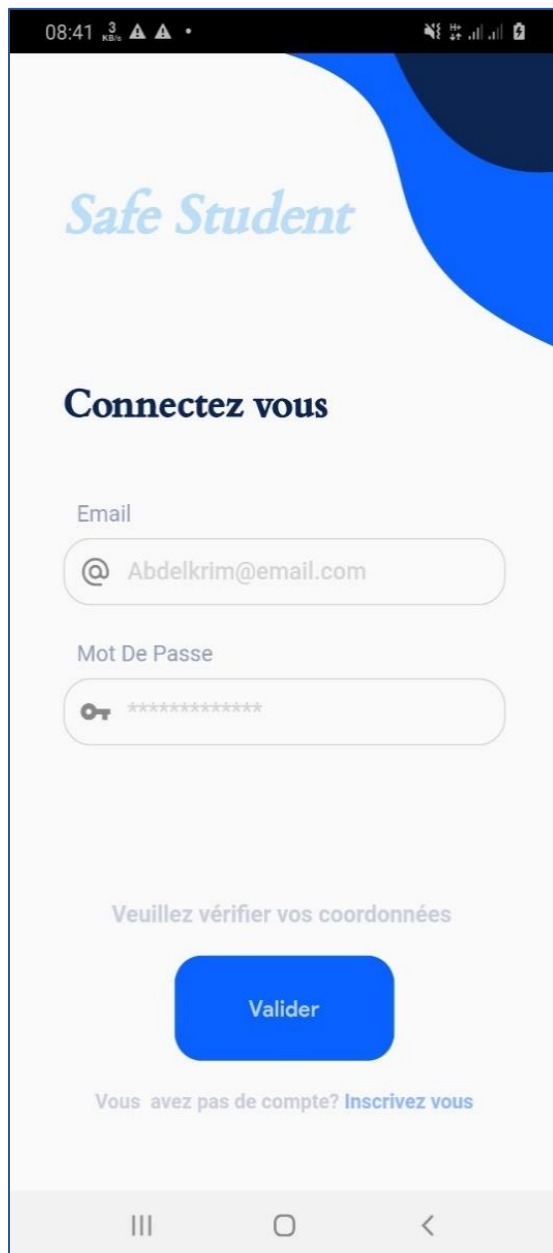
Visual studio code ou VS Code est un éditeur de code développé par Microsoft en 2015. Contrairement à ce à quoi Microsoft a eu l'habitude de nous habituer durant des années, il est l'un de ces premiers produits open source et gratuit, et surtout disponible sur les systèmes d'exploitation Windows, Linux et Mac. Vs code est développé avec le framework Electron et conçu principalement pour développer des projets avec **Javascript**, **Node.js** ou encore **TypeScript**.^[48]

Décider de changer d'éditeur de code n'est pas toujours simple. On possède souvent une solution à laquelle nous sommes habitués, qui nous convient la majorité du temps et on a peur de se retrouver perdu et de perdre par la même occasion en productivité. Pourtant, Visual Studio Code rassure la majorité des nouveaux utilisateurs dès les premières heures d'utilisation. Facile à installer, à comprendre, à utiliser et rapide, il dispose d'une interface graphique responsive et customisable via des thèmes déjà installés.^[49]

Quel que soit le langage : Javascript, PHP, JAVA, C#, C++, flutter ou autres, VS code permet de développer soit via les fonctionnalités par défaut (pour le HTML, CSS, Javascript, Typescript...) ou en ajoutant des extensions disponibles selon les besoins de chacun.

2. La description de notre application

2.1 Interface Authentification et demande d'inscription :



08:41 3 KB/s

Safe Student

Connectez vous

Email

@ Abdelkrim@email.com

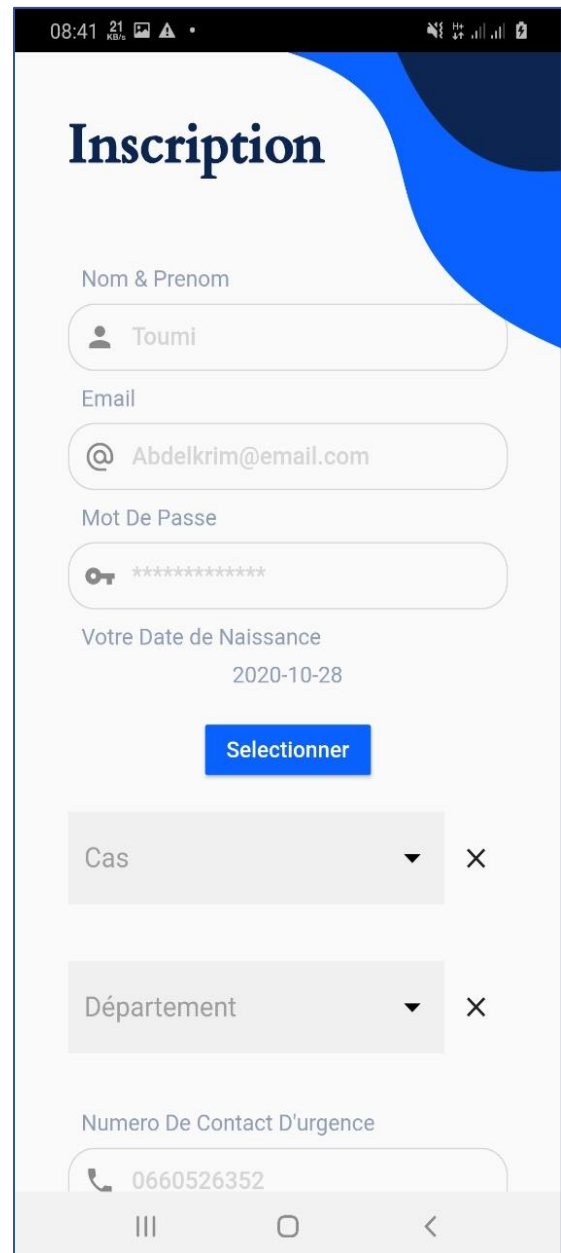
Mot De Passe

Veuillez vérifier vos coordonnées

Valider

Vous avez pas de compte? [Inscrivez vous](#)

Figure 16 : Interface authentification



08:41 21 KB/s

Inscription

Nom & Prenom

Toumi

Email

@ Abdelkrim@email.com

Mot De Passe

Votre Date de Naissance

2020-10-28

Selectionner

Cas

Département

Numero De Contact D'urgence

0660526352

Figure 17 : Interface demande d'inscription

2.2 Interface Agent HSE

2.2.1 Interface d'Accueil

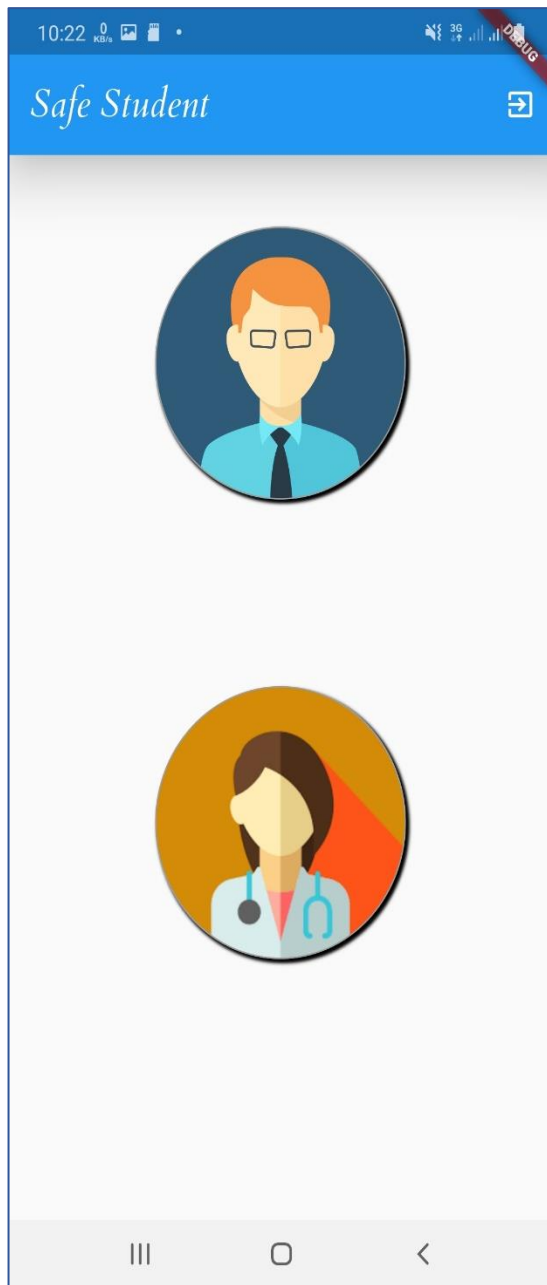


Figure 18 : Interface d'accueil

2.2.2 Interface Agent HSE (Liste Demandes)

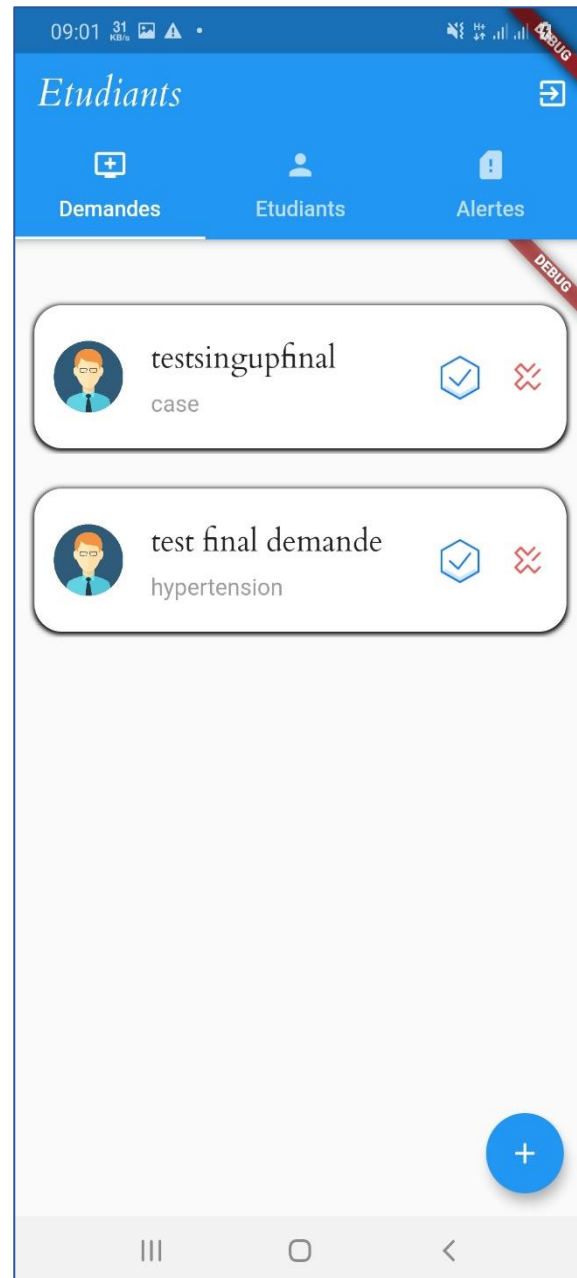


Figure 19 : Interface Agent HSE (Liste Demandes)

2.2.3 Interface Details de la demande

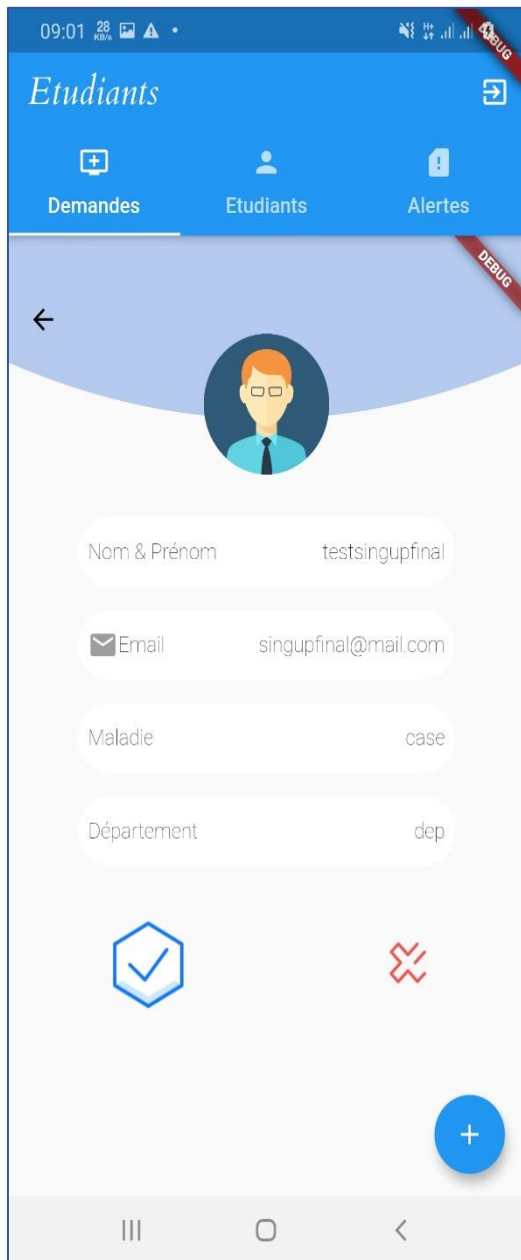


Figure 20 : Interface Details de La Demande

2.2.4 Interface Liste D'étudiants



Figure 21 : Interface Liste D'étudiants

2.2.5 Interface Details Etudiant

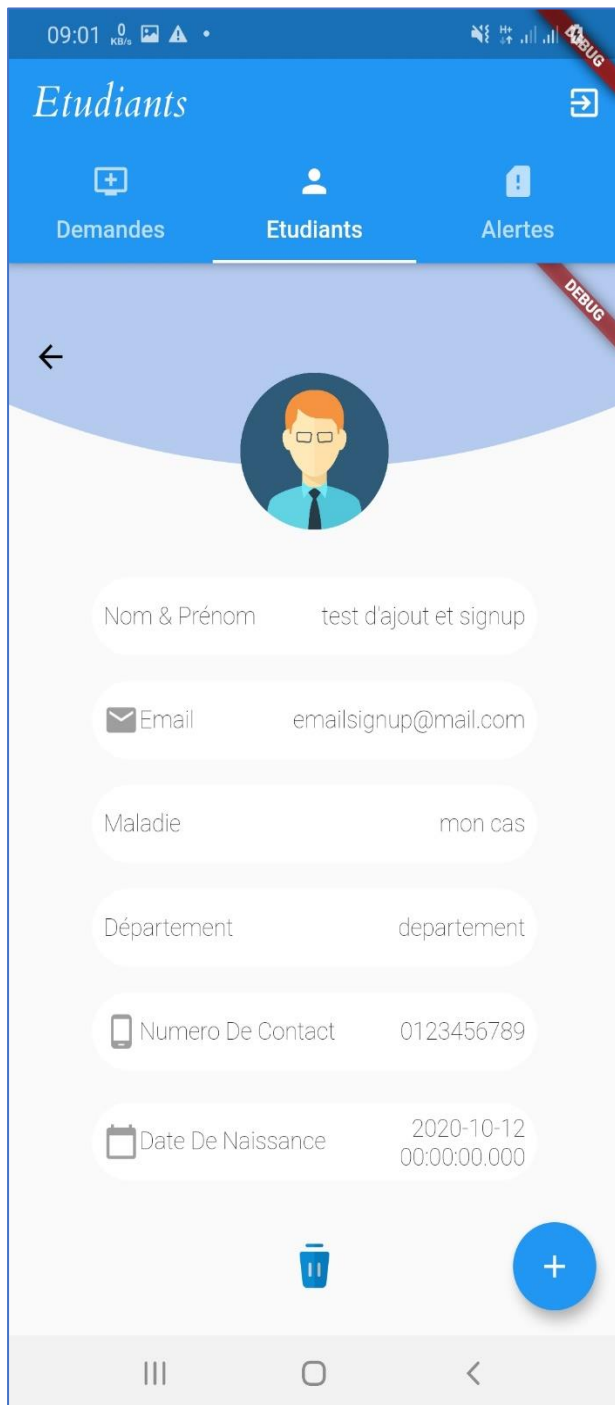


Figure 22 : Interface Details Etudiant

2.2.6 Interface Liste Des Alertes Validées

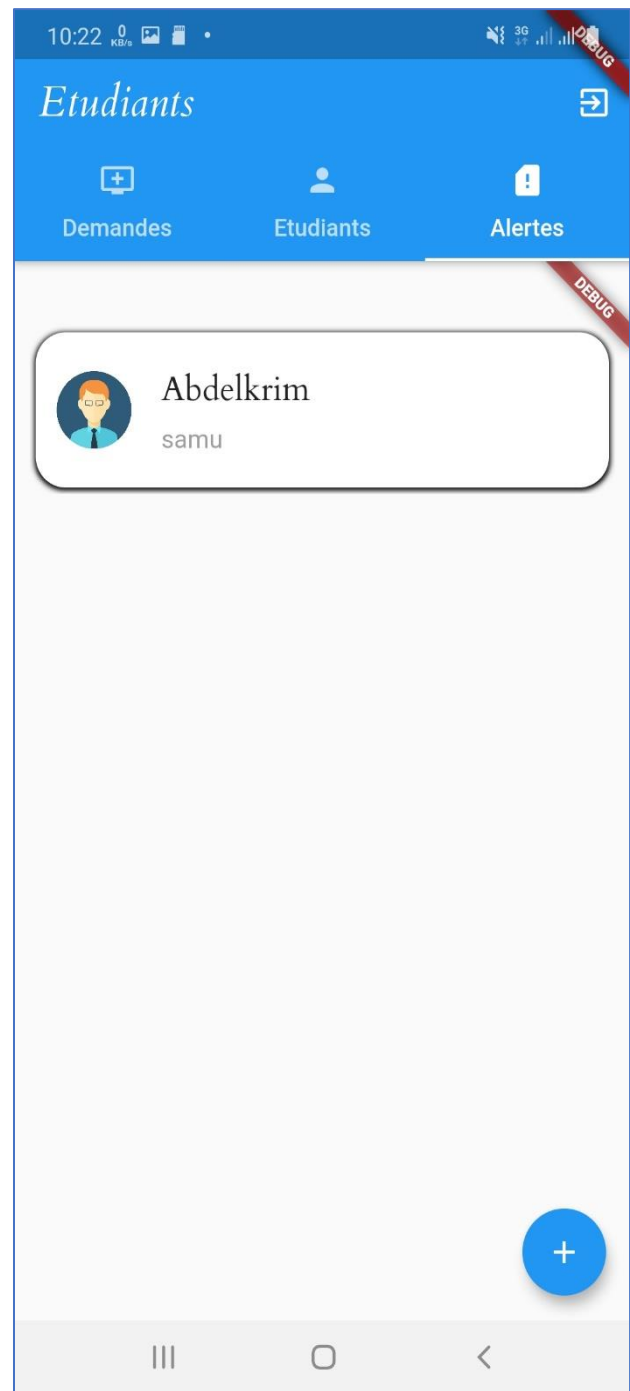


Figure 23 : Interface Liste Des Alertes Validées

2.2.7 Interface Details de l'Alerte Validée

The screenshot shows the 'Etudiants' app interface. At the top, there's a blue header with the title 'Etudiants' and a share icon. Below the header is a navigation bar with three tabs: 'Demandes', 'Etudiants', and 'Alertes'. The 'Alertes' tab is selected. Below the navigation bar is a dark blue curved banner with a back arrow on the left and a 'DEBUG' label on the right. In the center of the banner is a circular profile picture of a man with orange hair and glasses. Below the banner, there are several white rounded rectangular boxes containing the following information: 'Temps' with the value '2020-10-28 10:19:23.015334', 'Nom' with the value 'Abdelkrim', 'Evacuateur' with the value 'samu', 'Email' with the value 'formulaire@mail.com', 'Maladie' with the value 'cas', 'Date De Naissance' with the value '2020-10-23 22:12:47.018013', and 'Numero De Contact' with the value '0657146957'. At the bottom right of the screen is a blue circular button with a white plus sign. The bottom of the screen shows the Android navigation bar.

Figure 24 : Interface Details de l'Alerte Validée

2.2.8 Interface Ajout Etudiant

The screenshot shows the 'Ajouter un Etudiant' app interface. At the top, there's a blue header with a back arrow and the title 'Ajouter un Etudiant'. Below the header is a light gray form with the following fields: 'Nom & Prenom' with the value 'Toumi', 'Email' with the value 'Abdelkrim@email.com', 'Mot De Passe' with the value 'Password', and 'Votre Date de Naissance' with the value '2020-10-28'. Below these fields is a blue button labeled 'Selectionner'. Below the button are two dropdown menus: 'Cas' with a downward arrow and an 'X' icon, and 'Département' with a downward arrow and an 'X' icon. Below the dropdown menus is a field for 'Numero De Contact D'urgence' with the value '0660526352'. The bottom of the screen shows the Android navigation bar.

Figure 25 : Interface Ajout Etudiant

2.2.9 Interface Liste De Médecins

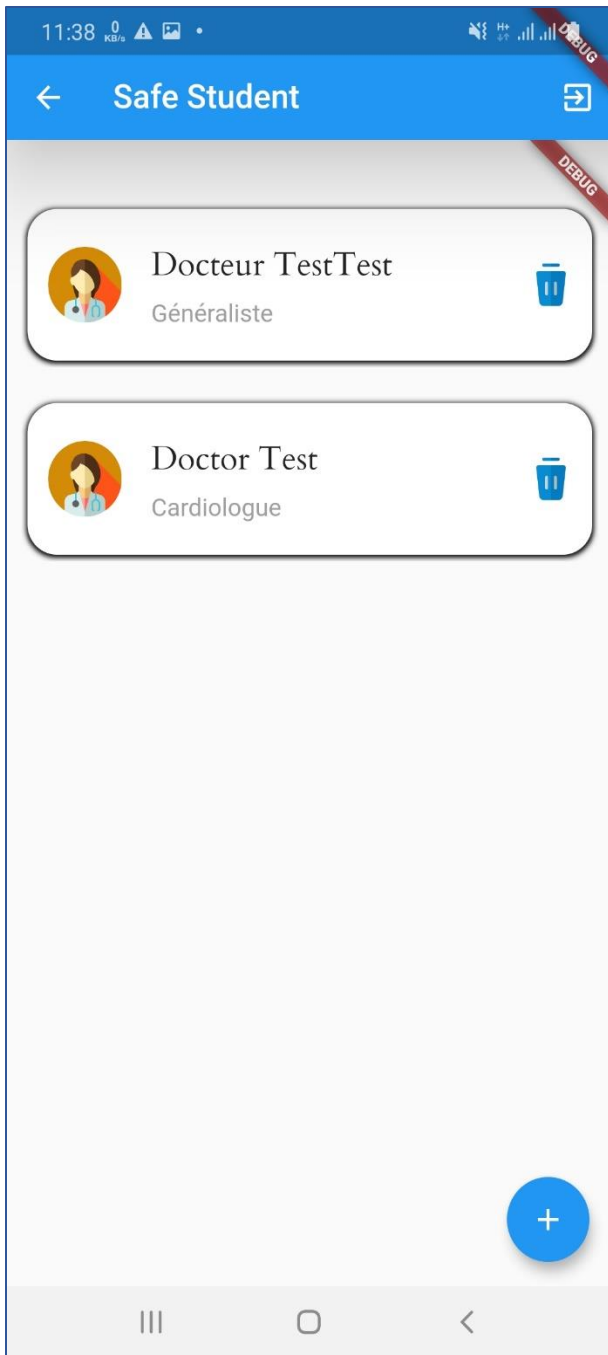


Figure 26 : Interface Liste De Médecins

2.2.10 Interface Détails du Médecin

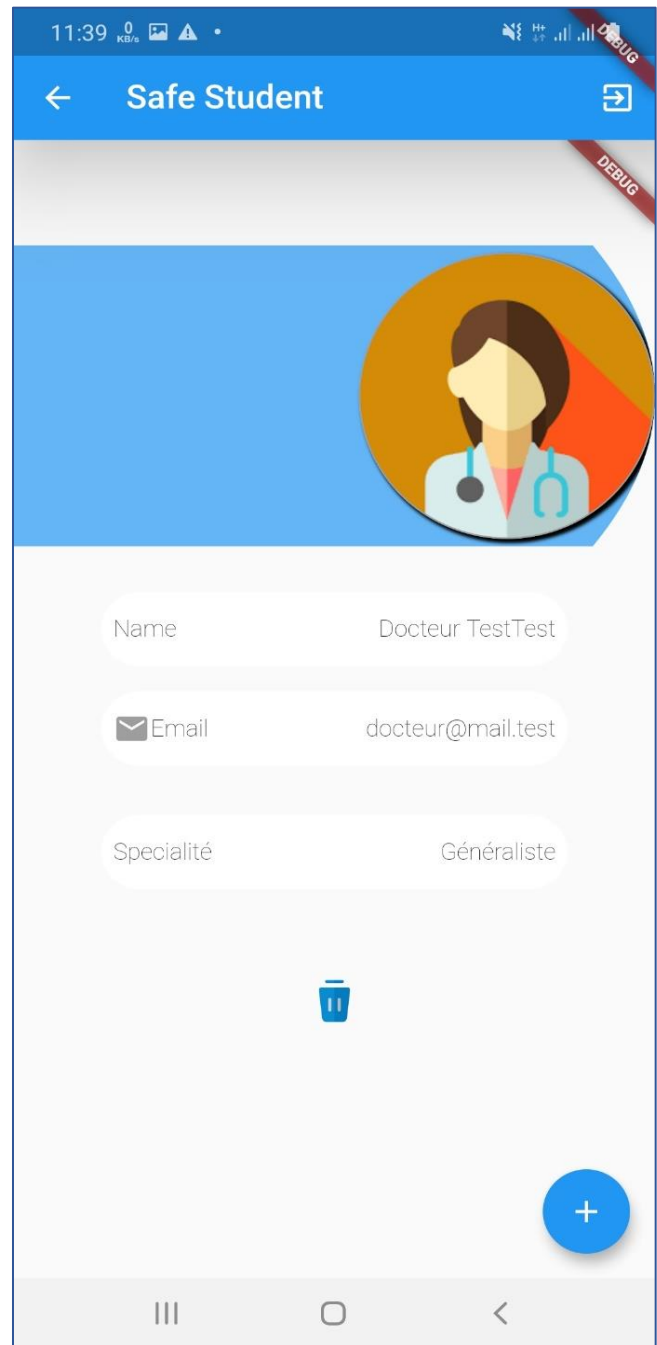


Figure 27 : Interface Détails Du Médecin

2.2.11 Interface Ajout d'un Médecin

12:42 0 KB/s 4G

Ajoutez Un Medecin

Nom & Prenom

Toumi

Email

Abdelkrim@email.com

Mot De Passe

Password

Spécialité

Adresse

Cite 100 lgts.....

Numero Du Medecin

0660526352

III

Figure 28 : Interface Ajout d'un Médecin

2.3 Interface Etudiant Inscris

2.3.1 Interface Accueil Etudiant

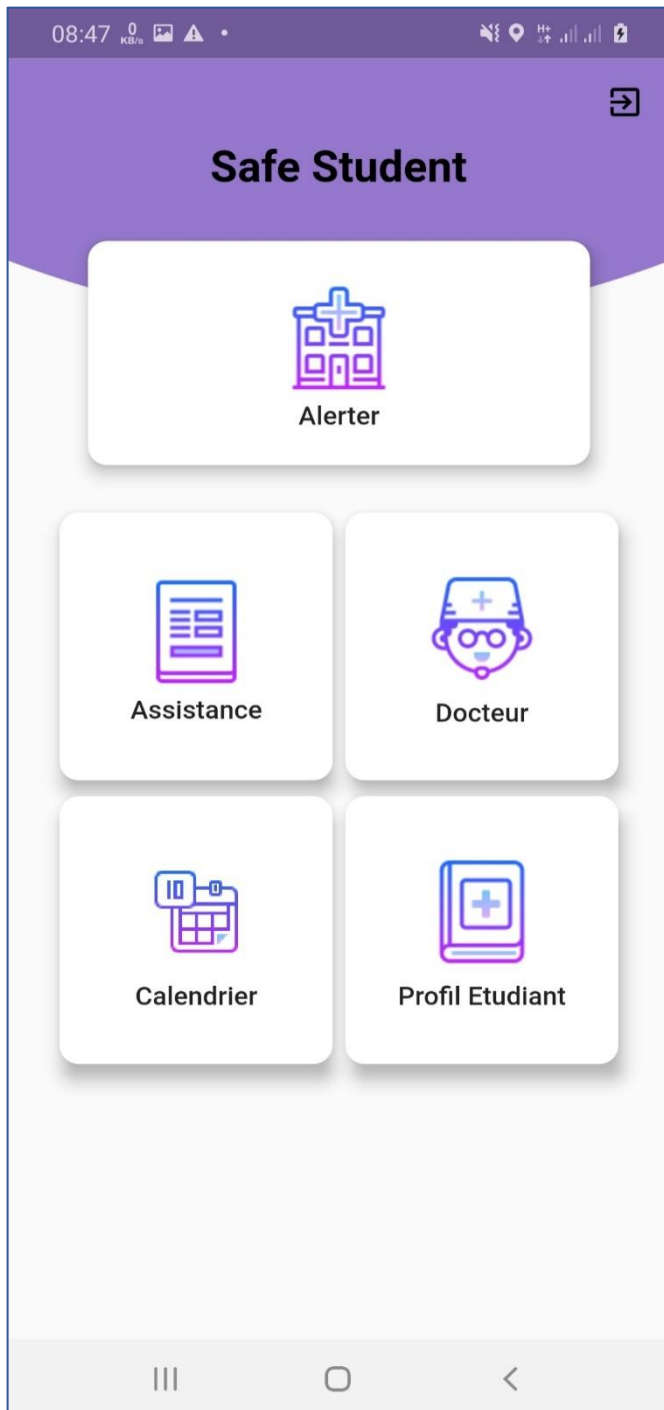


Figure 29 : Interface Accueil Etudiant

2.3.2 Interface Profil Etudiant

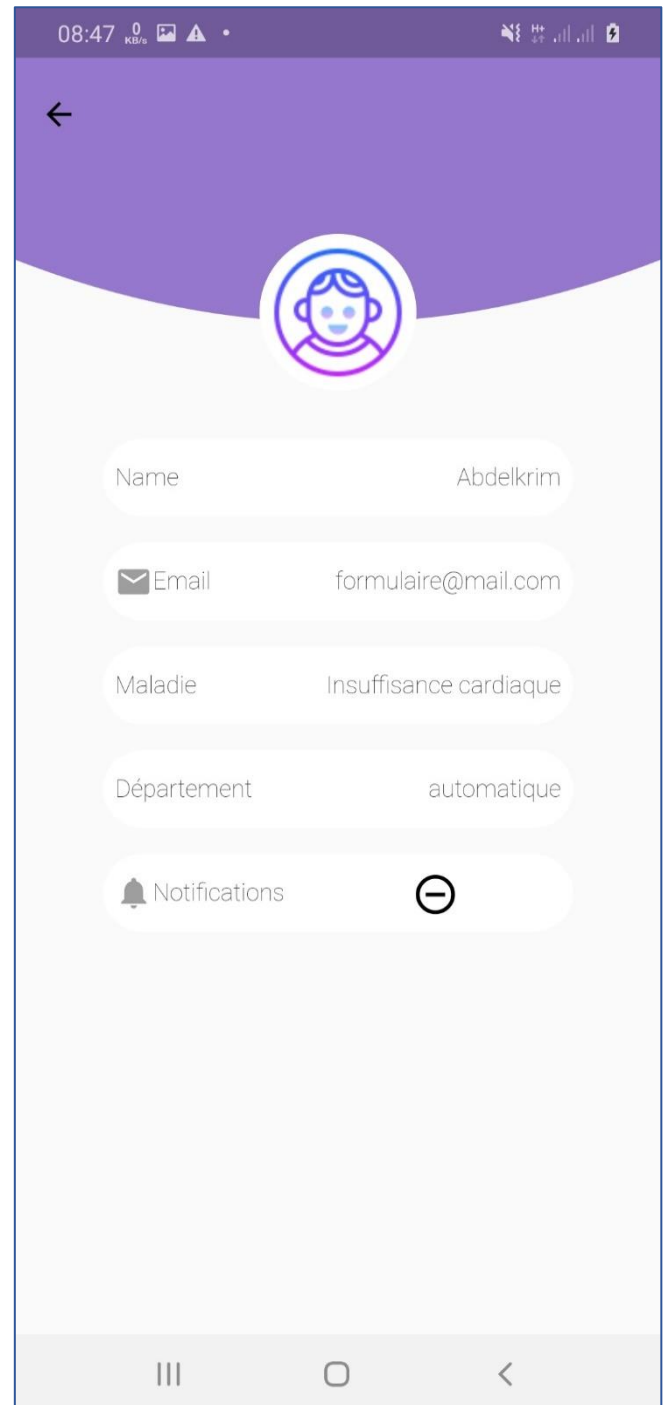


Figure 30 : Interface Profil Etudiant

2.3.3 Interface Calendrier Etudiant

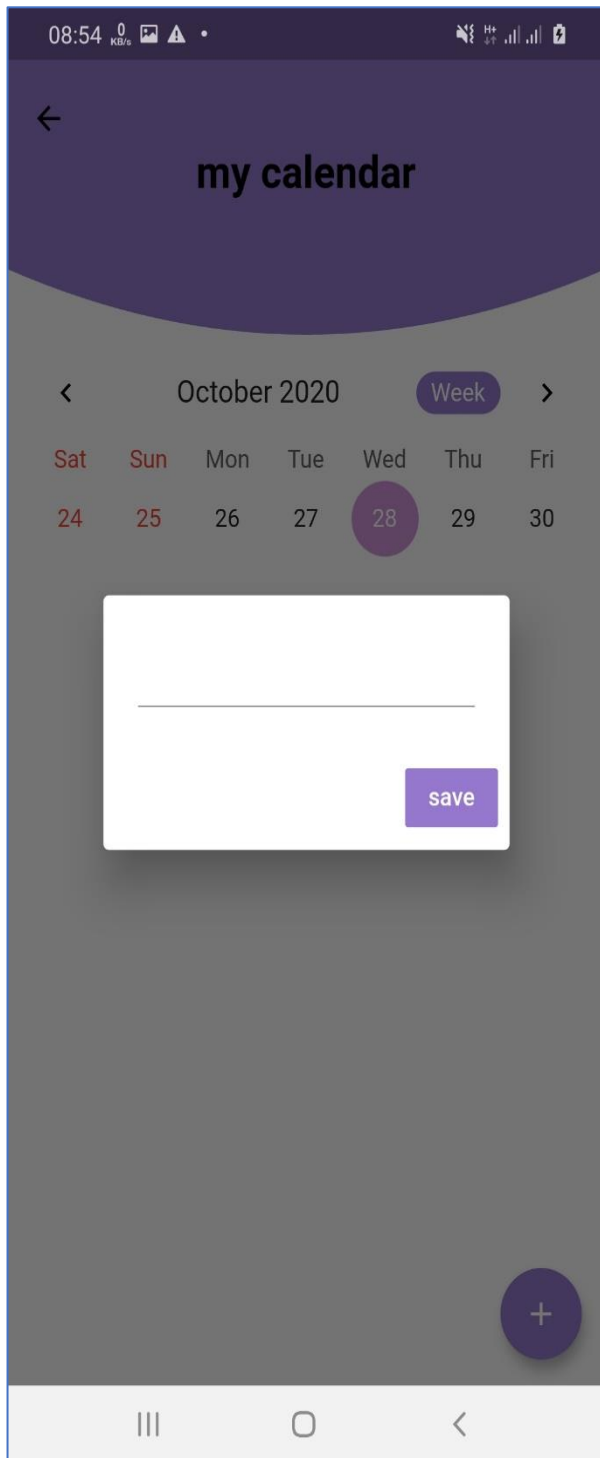


Figure 31 : Interface Calendrier Etudiant

2.3.4 Interface Validation d'Alerte



Figure 32 : Interface Validation d'Alerte

2.3.5 Interface de Liste des maladies

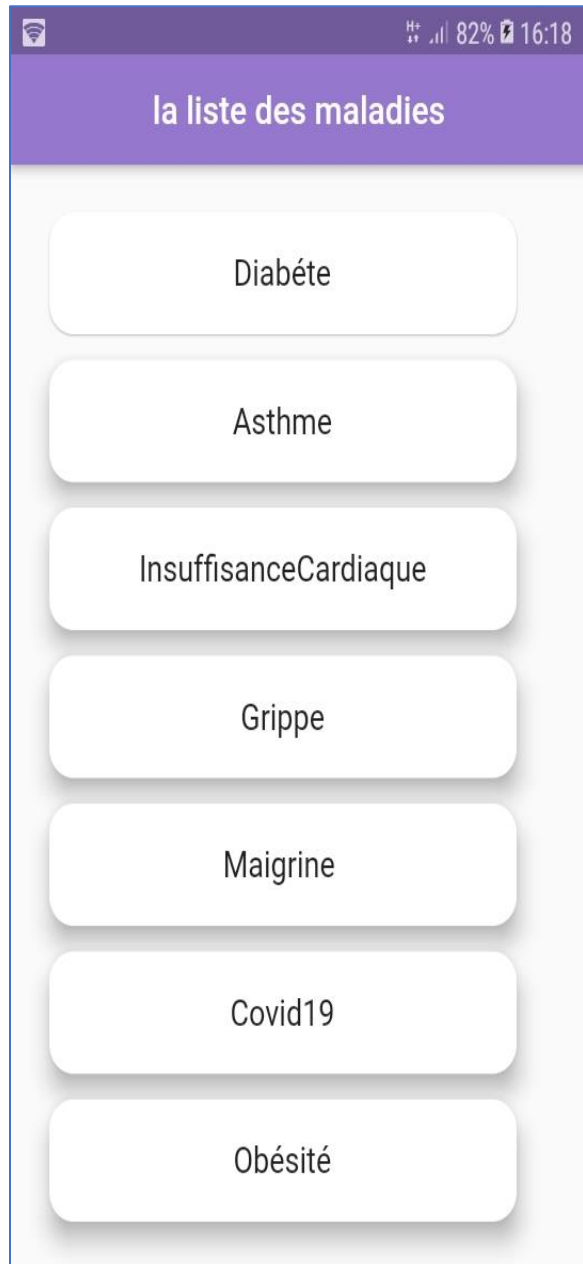


Figure 33 : Interface de Liste des maladies

2.3.6 Interface d'information sur la maladie

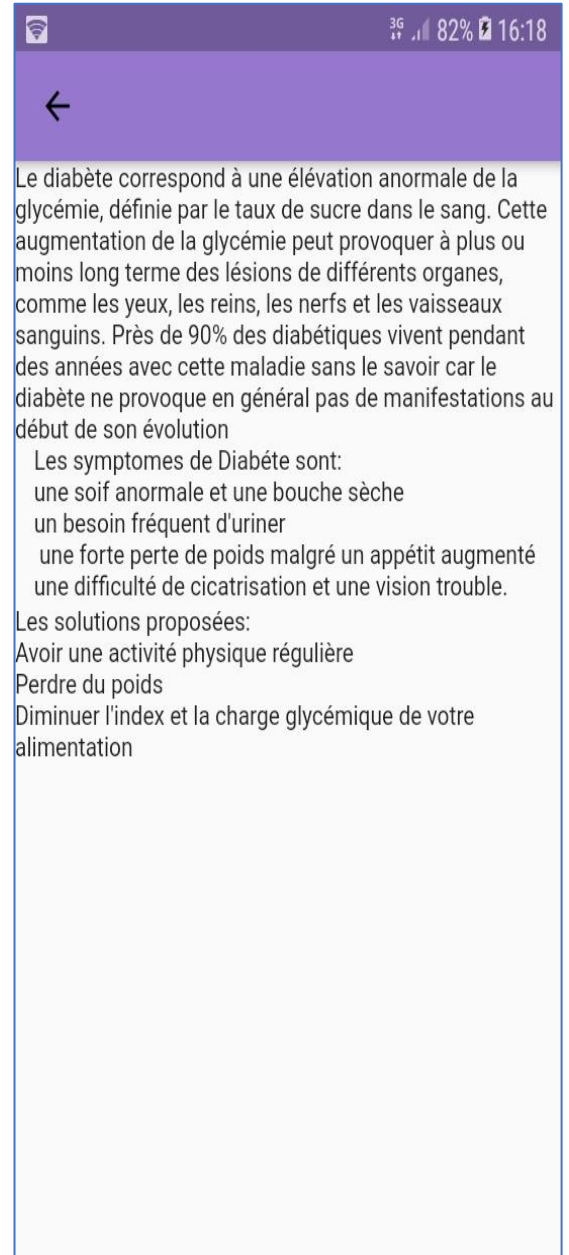


Figure 34 : Interface d'information sur la maladie

2.4 Interface Messagerie : Cette interface est la même pour l'étudiant et les médecins, et elle est la seule interface accédée par les médecins.

2.4.1 Interface liste des contacts

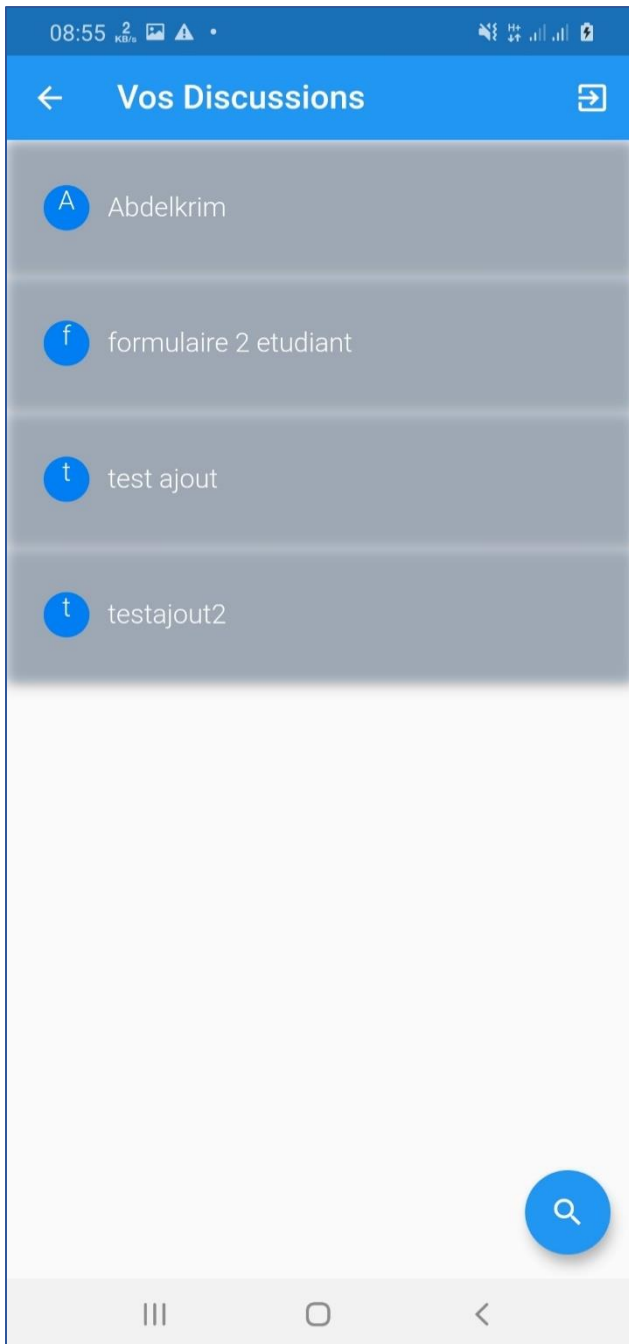


Figure 35 : Interface liste des contacts

2.4.2 Interface recherche de Contact

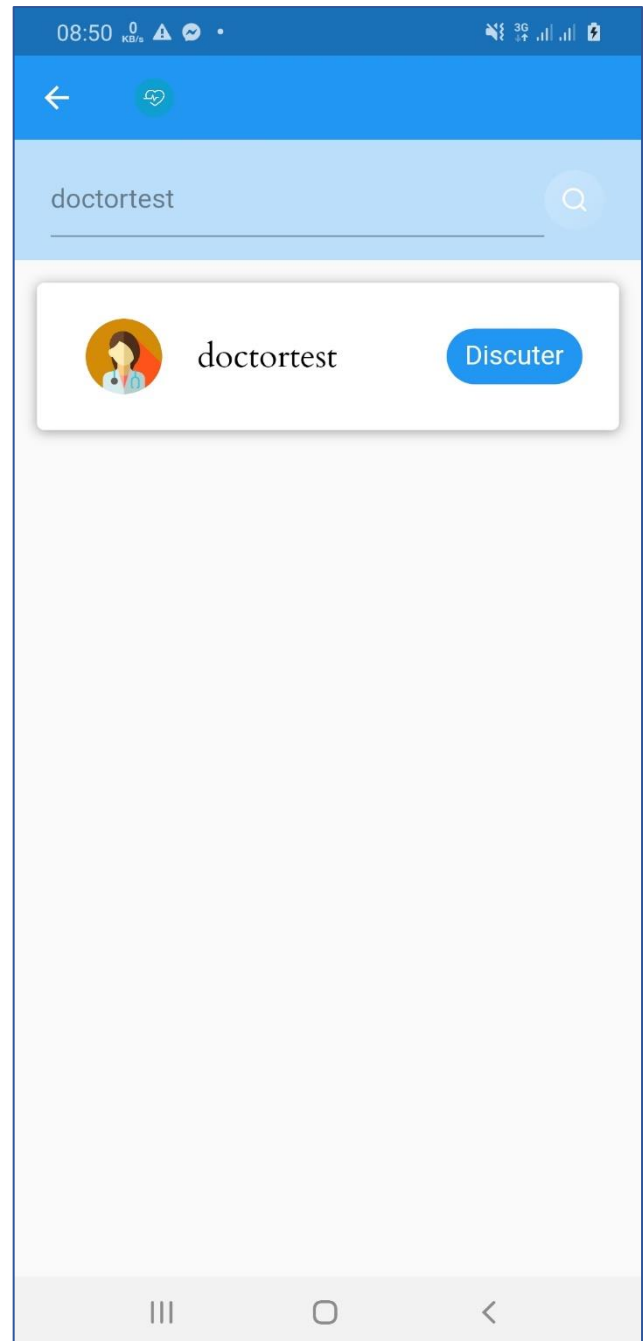


Figure 36 : Interface recherche de Contact

2.4.3 Interface Discussion



Figure 37 : Interface Discussion

2.4.4 Interface message d'alerte textuelle

(Concerne uniquement le médecin)

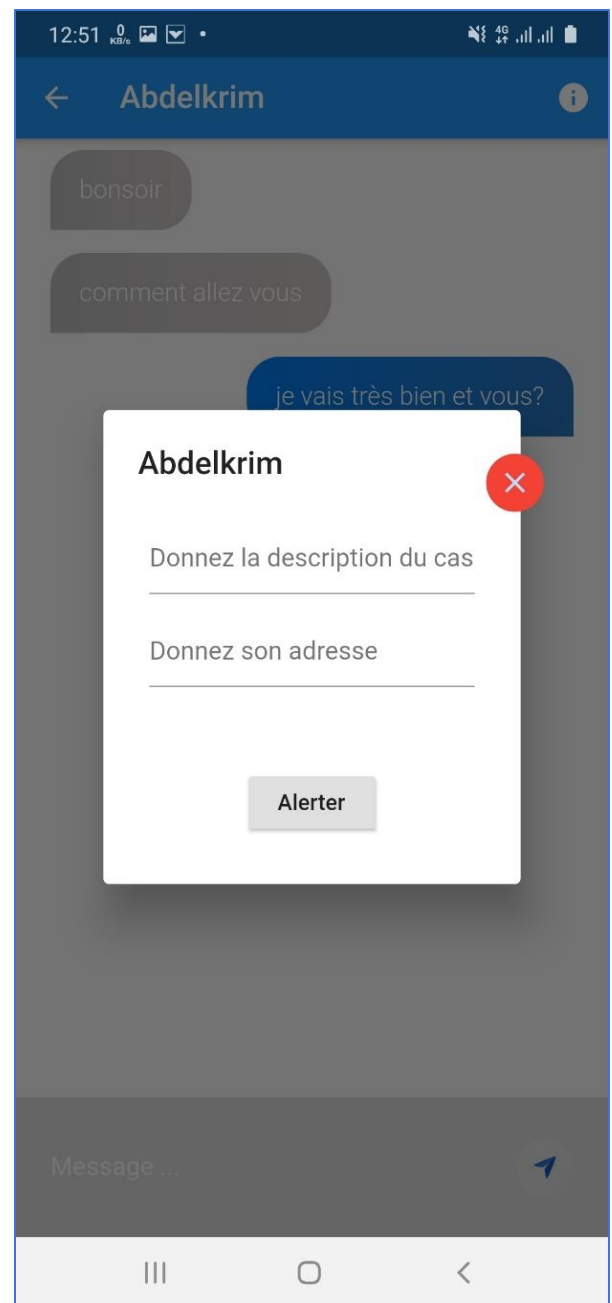


Figure 38 : Interface de message d'alerte

2.5 Interface des Partenaires d'alerte : Dans notre application, nous avons défini deux partenaires d'alertes ; le SAMU et La Protection civile qui ont accès aux mêmes fonctionnalités.

2.5.1 Interface du Samu

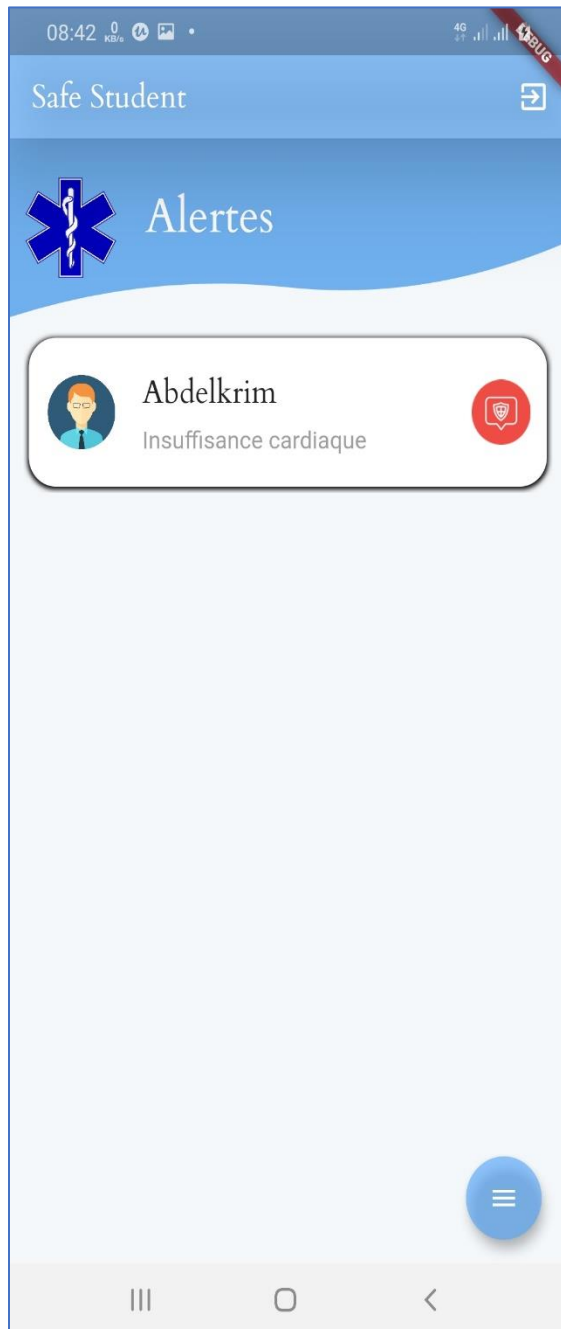


Figure 39: Interface du Samu

2.5.2 Interface de la Protection Civile

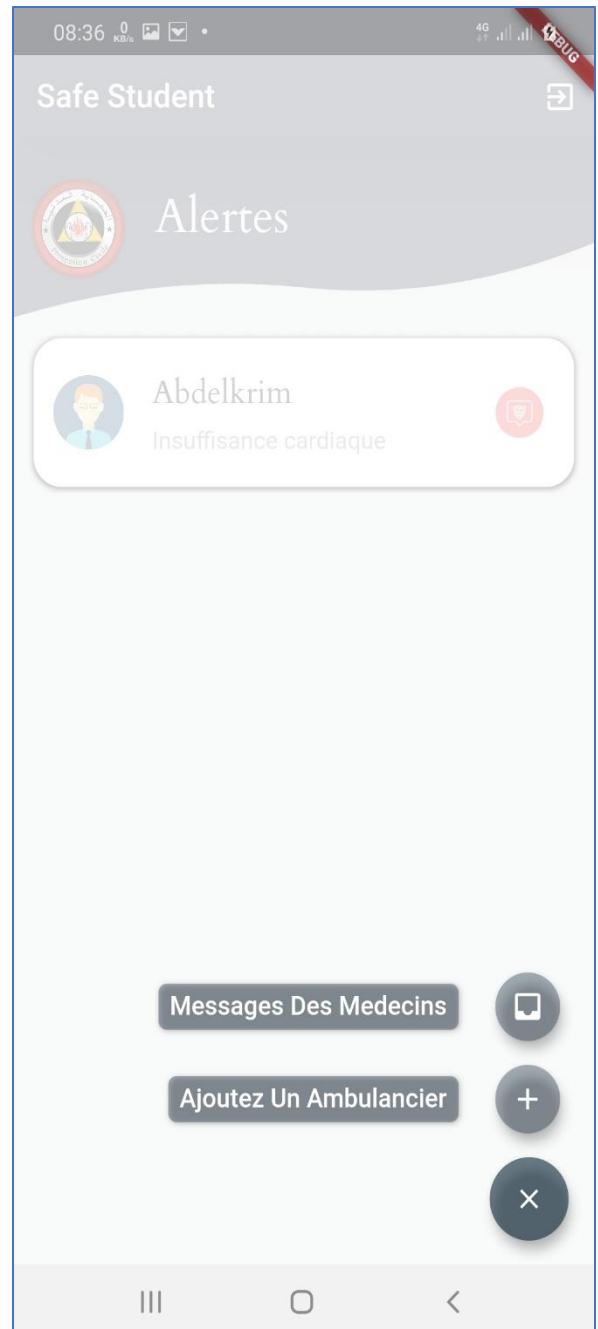
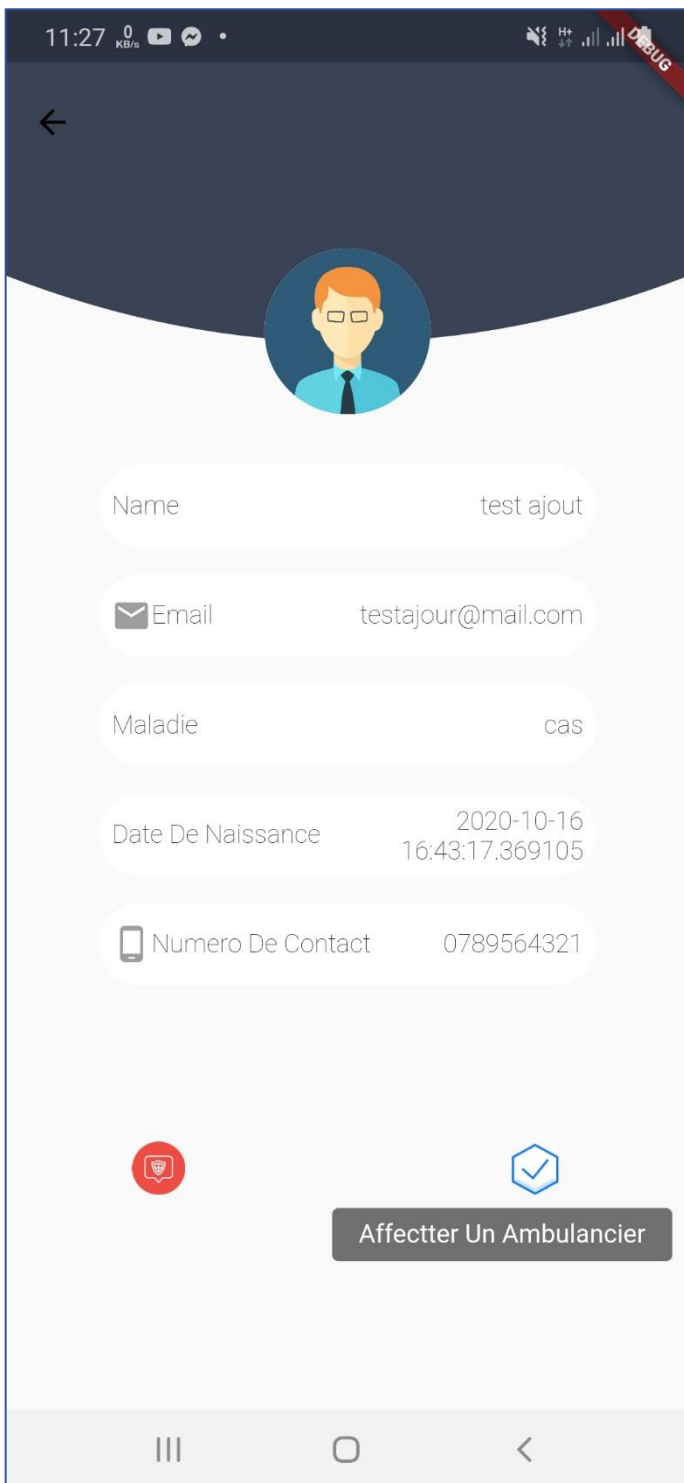



Figure 40 : Interface de la Protection Civile

2.5.3 Interface Details D'une Alerte



11:27 0 KB/s

←





Name test ajout

Email testajour@mail.com

Maladie cas

Date De Naissance 2020-10-16 16:43:17.369105

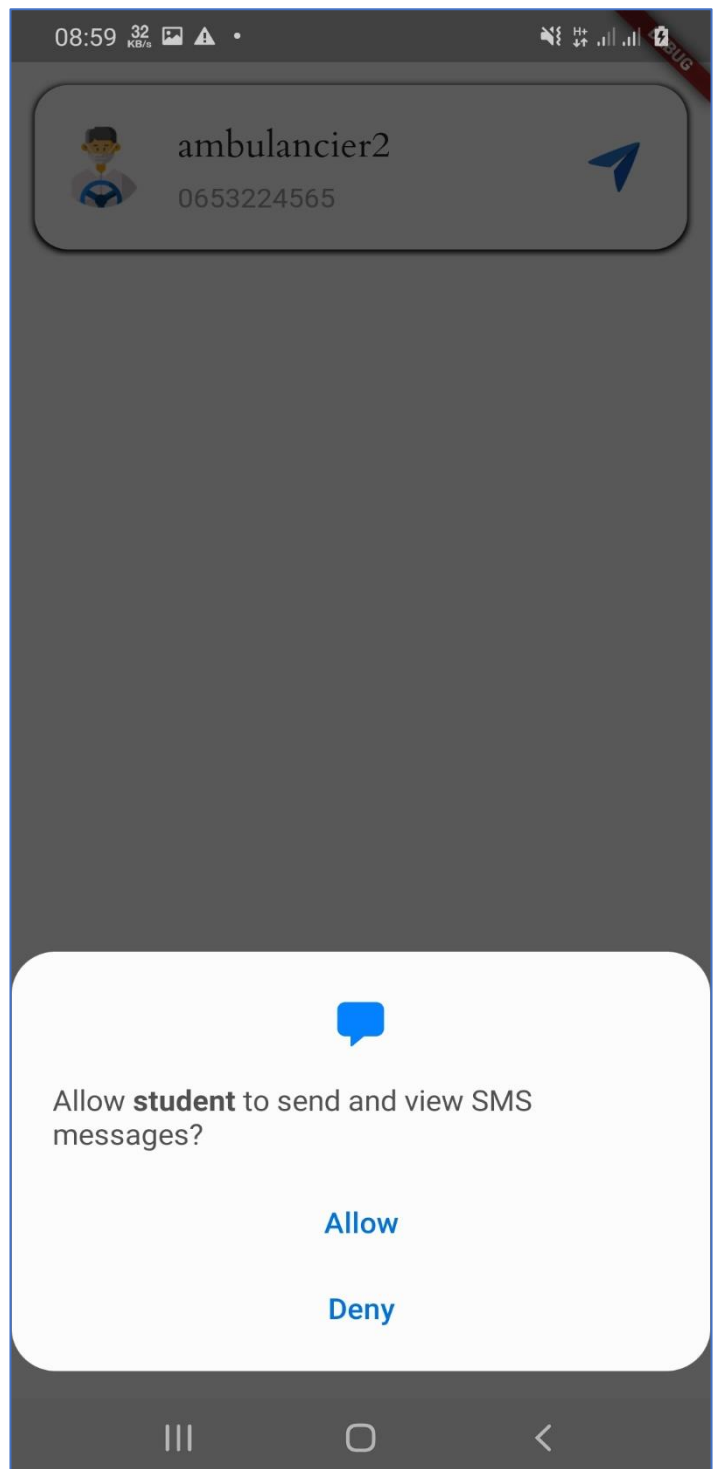
Numero De Contact 0789564321


Affecter Un Ambulancier


Figure 41 : Interface Details D'une Alerte

2.5.4 Interface Affectation d'un Ambulancier



08:59 32 KB/s

 ambulancier2 0653224565



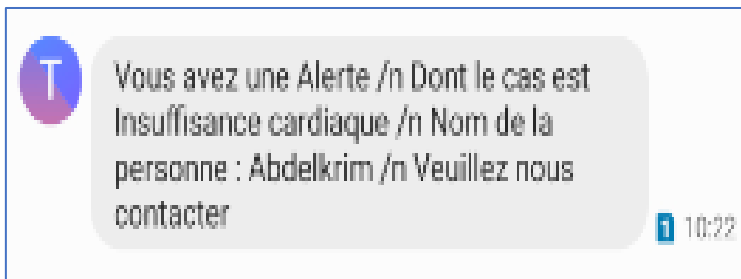
Allow **student** to send and view SMS messages?

Allow

Deny

Figure 42 : Interface d'affectation ambulancier

2.5.5 Exemple d'un SMS envoyé à un Ambulancier



2.5.6 Interface Ajout D'ambulancier

A screenshot of a mobile application interface titled "Ajoutez Un Ambulancier". The interface features a dark blue header with a circular logo on the left. Below the header, there are three input fields for user information: "Nom & Prenom" (filled with "Boubchir Yasnima"), "Email" (filled with "Yasmina@email.com"), and "Numero De Telephone" (filled with "0657146957"). Each field has a corresponding icon (person, email, and phone). At the bottom of the form, there are two buttons: a blue button with a person icon and a plus sign, and a red button with a crossed-out 'X' icon. The bottom of the screen shows a standard Android navigation bar with three icons: a square, a circle, and a triangle.

Figure 43 : Exemple d'envoi d'SMS

Figure 44 : Interface Ajout D'ambulancier

2.5.7 Interface de la carte GoogleMaps :

Elle permet de montrer la géolocalisation de L'alerte.



Figure 45 : Interface de géolocalisation d'alerte

2.5.8 Interface « fiche message reçu »

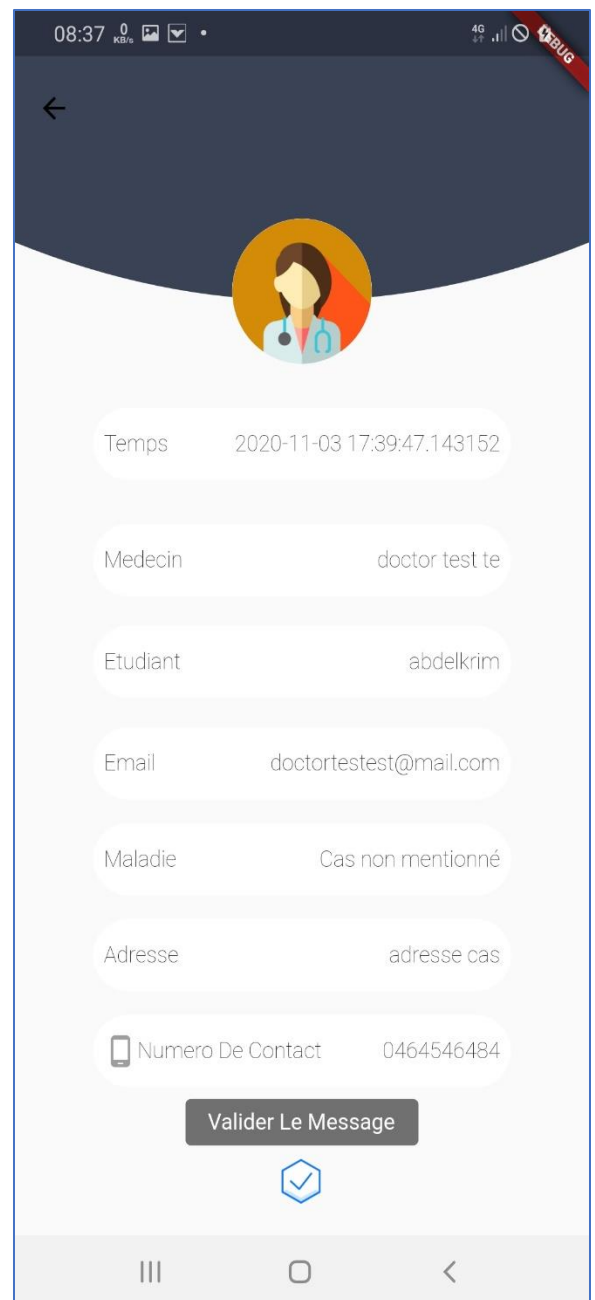


Figure 46 : Interface « fiche message reçu »

2.5.9 Interface liste des alertes :

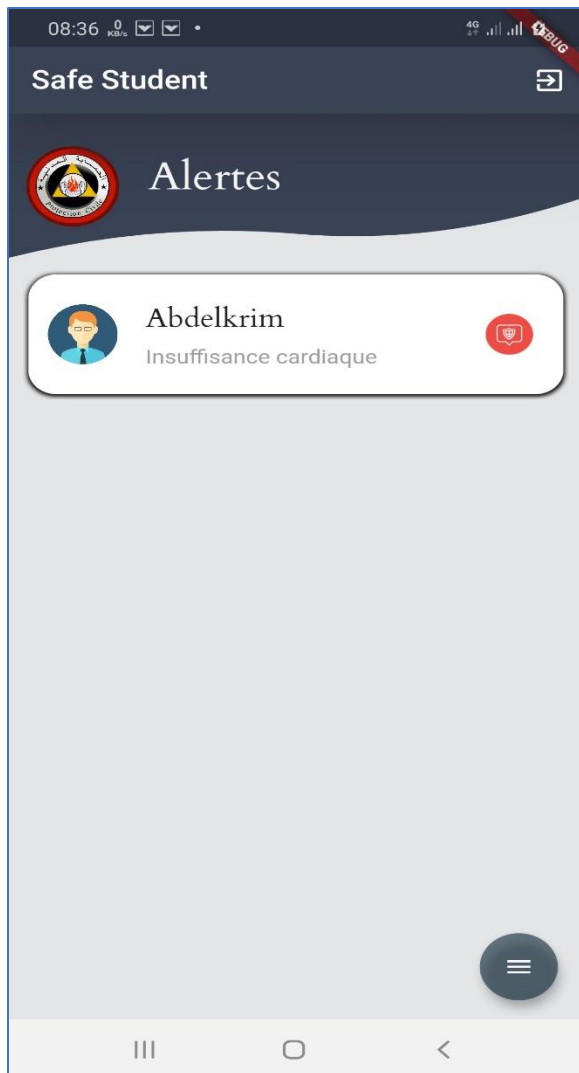


Figure 47 : Interface liste des alertes.

2.5.10 Interface liste des messages reçus :

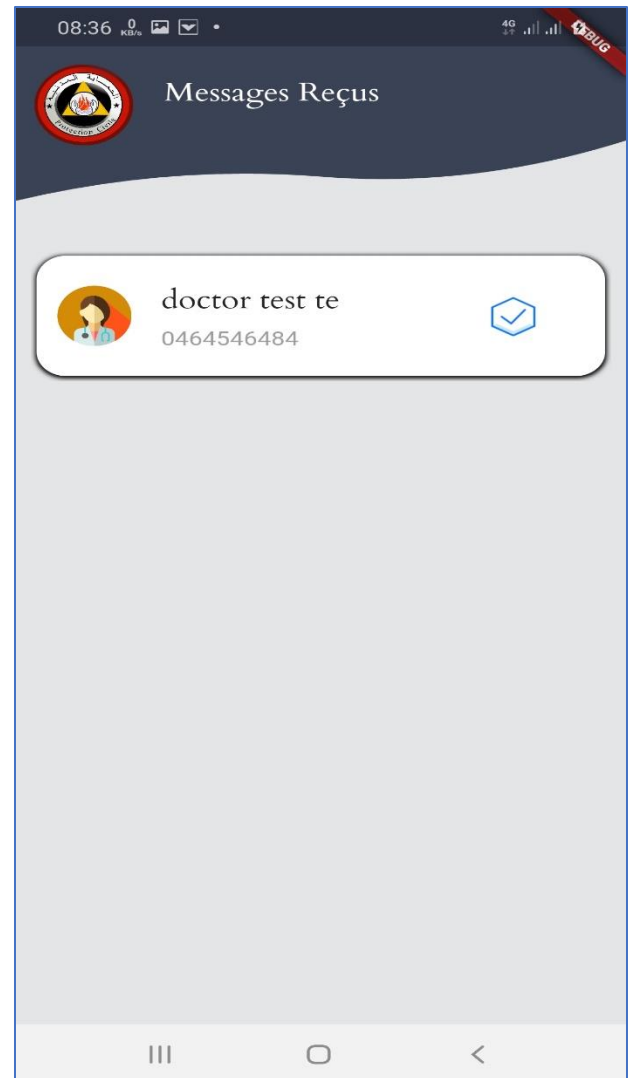


Figure 48 : Interface liste des messages reçus.

Conclusion

Dans ce dernier chapitre, nous avons présenté la dernière phase de la conception de notre application à savoir la réalisation, ou nous avons cité et défini les outils logiciels que nous avons utilisé pour développer notre solution, et nous avons aussi présenté notre application à travers leurs interfaces utilisateurs.

Conclusion générale

Dans le cadre de notre projet de fin d'études, notre objectif était de développer une assistance médicale multiplateforme destinée aux étudiants a maladies chroniques et éventuellement les étudiants à besoins spécifiques afin de leurs offrir l'aide en cas d'urgence en alertant le SAMU ou la Protection civile ; et permettre une communication rapide avec l'étudiant et des médecins ; accessible via leurs smartphones, tablette, Pc dans tous les systèmes d'exploitation.

L'application a été modélisée en utilisant le langage UML qui nous a permis de concevoir cette dernière ; et la réalisation a été faite en utilisant une combinaison de technologies et de langages.

Notre solution a abouti à une application qui fonctionne sur Android et IOS et qui répond aux attentes des utilisateurs qui permet l'évacuation de l'étudiant en cas d'urgence qui se fera avec l'alerte de SAMU ou de la Protection Civile, ainsi que la possibilité de communiquer avec un médecin afin de traiter son cas de maladie et répondre à ces questions grâce au système de messagerie en temps réel.

Comme perspectives, nous avons l'intention de terminer l'implémentation de certaines fonctionnalités comme la création d'une application de pointage automatique dédiée à l'ambulancier qui sera automatisé avec le système de service SAMU et celui de la Protection Civile ; elle reste en développement dans le but d'avoir un système utilisable par l'ensemble des étudiant en besoin. Elle peut être améliorée comme suit :

- Connecter l'université réellement avec le service SAMU et la Protection Civile.
- Ajout d'autres partenaires d'alerte comme les services la police en cas d'agression d'un étudiant par exemple.
- Faire fonctionner l'application sur tous les systèmes d'exploitation existants.
- Connecter l'application au serveur de l'université.
- Ajouter d'autres fonctionnalités comme l'alerte aux parents de l'étudiant évacué.
- Ajouter une assistance adaptative selon le type handicap en cas d'étudiants à besoins spécifiques comme l'affectation d'outils (Ajouter un ChatBot).
- Avoir une application dédiée pour chaque type d'utilisateur afin de profiter de tous les avantages que pourrait offrir notre plateforme (comme un système de notifications).

La réalisation de ce travail nous a été d'une grande utilité dans la mesure où elle nous a permis d'approfondir nos connaissances des technologies dans le domaine de développement multiplateforme, le développement d'applications en temps réel, le génie logiciel, les architectures logicielles etc.

- ❖ [1] Segal, Troy. “Inside Decision Support Systems-DSS.” *Investopedia*, Investopedia, 29 Jan 2020, www.investopedia.com/terms/d/decision-support-system.asp ; consulté le 04 juin 2020.
- ❖ [2]” *Nature News*, Nature Publishing Group, 6 Feb. 2020, www.nature.com/articles/s41746-020-0221-y ;consulté le 18 juin 2020.
- ❖ [3] “Decision Support System.” *Wikipedia*, Wikimedia Foundation, 10 Apr. 2020, en.wikipedia.org/wiki/Decision_support_system ; consulté le 18 juin 2020.
- ❖ [4] Sutton, Reed T., et al. “An Overview of Clinical Decision Support Systems: Benefits, Risks, and Strategies for Success ; consulté le 18 juin 2020.
- ❖ [5] Sugumaran, R., and J. Degroote. *Spatial Decision Support Systems: Principles and Practice*. CRC, 2011; consulté le 18 juin 2020.
- ❖ [6] “Demande D'assistance Informatique.” *Wikipedia*, Wikimedia Foundation, 9 June 2020, fr.wikipedia.org/wiki/Demande_d'assistance_informatique ; consulté le 20 juin 2020.
- ❖ [7] “Assistance.” *Wikipedia*, Wikimedia Foundation, 20 June 2016, fr.wikipedia.org/wiki/Assistance; consulté le 20 juin 2020.
- ❖ [8] “Services D'assistance.” *Wikipedia*, Wikimedia Foundation, 9 June 2020, fr.wikipedia.org/wiki/Services_d'assistance ; consulté le 20 juin 2020.
- ❖ [9] “Téléconseiller.” *Wikipedia*, Wikimedia Foundation, 8 Aug. 2020, fr.wikipedia.org/wiki/Téléconseiller ; consulté le 22 juin 2020.
- ❖ [10] “Consultant.” *Wikipedia*, Wikimedia Foundation, 9 Aug. 2020, fr.wikipedia.org/wiki/Consultant; consulté le 22 juin 2020.

- ❖ [11] Etudiant.gouv.fr. “La Santé Sur Les Campus.” *Etudiant.gouv.fr*,
www.etudiant.gouv.fr/cid96662/la-sante-sur-les-campus.html ; consulté le 24 juin 2020.
- ❖ [12] VibeThemes. “Responsable Hygiène Sécurité Et Environnement.” *Université Virtuelle Privée Du Gabon*, 12 Apr. 2020, www.univga-edu.ga/course/hygiene-securite-et-environnement/; consulté le 24 juin 2020.
- ❖ [13] “Ordinateur : Choisir Les Bons Logiciels De Messagerie Instantanée.” *Ooreka.fr*,
ordinateur.ooreka.fr/comprendre/messagerie-instantanee ; consulté le 24 juin 2020
- ❖ [14] *Google Search*, Google, www.google.com/search?client=firefox-b-d&q=usages+de+la+messagerie+instantanéé ; consulté le 24 juin 2020.
- ❖ [15] Asal, Hakan. “What Is Cross-Platform Software?” *Medium*, Medium, 17 July 2019,
medium.com/@hakanasal51/what-is-cross-platform-software-38ee57b7304a ; consulté le 29 juin 2020.
- ❖ [16] Bobology.com. “What Is Cross-Platform Software?” *Bobology.com*, 7 Feb. 2020,
www.bobology.com/public/What-is-CrossPlatform-Software.cfm ; consulté le 29 juin 2020.
- ❖ [17] “Conception D'un Logiciel Multiplateforme Pour La Gestion D'un Restaurant Moderne.” *Memoire Online*, www.memoireonline.com/07/19/10847/m_Conception-dun-logiciel-multiplateforme-pour-la-gestion-dun-restaurant-moderne5.html ; consulté le 29 juin 2020.
- ❖ [18] “Les Événements Du Club Décision DSI En Vidéos Et En Photos.” *IT Research – L'information IT Des Décideurs*, www.itresearch.fr/multi-plateforme/ ; consulté le 30 juin 2020.
- ❖ [19] Pierrat, Olivier. “Le Développement Multi-Plateforme : Enjeux, Promesses Et Réalité.” *Journaldunet.com*, JDN, 28 Feb. 2012, www.journaldunet.com/web-tech/developpeur/1029855-le-developpement-multi-plateforme-enjeux-promesses-et-realite/ consulté le 30 juin 2020.

- ❖ [20] SUPINFO - Ecole Informatique - Formation en Informatique - Paris, Lyon. “Articles - Étudiants SUPINFO.” *Le Développement Mobile Multiplateforme | SUPINFO, École Supérieure D'Informatique*, www.supinfo.com/articles/single/3967-developpement-mobile-multiplateforme ; consulté le 1 juillet 2020.
- ❖ [21] Développer Une Application Hybride, fr.goodbarber.com/creer-app/49-developper-application-hybride/ ; consulté le 1 juillet 2020.
- ❖ [22] “Overview.” *Architectural Overview of Cordova Platform - Apache Cordova*, cordova.apache.org/docs/en/latest/guide/overview/index.html ; consulté le 2 juillet et 2020
- ❖ [23] Loïc Knuchel. “Découvrir Ionic Framework - Loïc Knuchel .Loïc Knuchel, 26 Jan. 2015, loicknuchel.fr/blog/2015/01/26/decouvrir-ionic-framework ;consulté le 3 juillet 2020.
- ❖ [24] Profexorgeek. “What Is Xamarin? - Xamarin.” *Xamarin | Microsoft Docs*, docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin ;Consulté le 3 juillet 2020.
- ❖ [25] SUPINFO - Ecole Informatique - Formation en Informatique - Paris, Lyon. “Articles - Étudiants SUPINFO.” *Le Développement Mobile Multiplateforme | SUPINFO, École Supérieure D'Informatique*, www.supinfo.com/articles/single/3967-developpement-mobile-multiplateforme ; consulté le 4 juillet 2020.
- ❖ [26] “React Native & Le Développement Mobile.” *Easy Partner*, 25 Feb. 2020, www.easypartner.fr/blog/pourquoi-utiliser-react-native/ ; consulté le 04 juillet 2020.
- ❖ [27] Technodor Dr. “C'est Quoi Flutter ?” *Medium*, Flutter : La Communauté Francophone, 17 June 2020, medium.com/flutter-la-communauté-francophone/cest-quoi-flutter-4b9efb30da0c consulté le 04 juillet 2020.
- ❖ [28] “Formation UML 2, Processus De Développement.” *AgoraTIC, Centre De Formation Informatique - Le Meilleur De La Formation Professionnelle*,

www.agoratic.com/formation/formation-uml-2-processus-de-developpement.php ; consulté le 13 juillet 2020.

- ❖ [29] Gestion de projet – MERISE - UML - SGBDR ORACLE - AGL – Management des systèmes d’information. “Quelle Démarche Suivre ?” *OpenClassrooms*, openclassrooms.com/fr/courses/2035826-debutez-lanalyse-logicielle-avec-uml/2049091-quelle-demarche-suivre ; consulté le 13 juillet 2020.
- ❖ [30] “Comment Faire Un Cahier Des Charges ?” *Cahiers Des Charges*, cahiersdescharges.com/comment-faire-cahier-des-charges/ ; consulté le 14 juillet 2020.
- ❖ [31] Nihel, Moussi. “Chapitre 2 : Spécification Et Analyse Des Besoins.” *Academia.edu*, [www.academia.edu/33346693/Chapitre_2_Spécification_et_analyse_des_besoins](https://www.academia.edu/33346693/Chapitre_2_Sp%C3%A9cification_et_analyse_des_besoins) ; consulté le 14 juillet 2020.
- ❖ [32] “Système D'Information De Gestion.” *Le Diagramme De Contexte [Système D'Information De Gestion]*, gainfolemans.github.io/SGBD/co/contexte.html ; consulté le 16 juillet 2020.
- ❖ [33] “Diagramme De Cas D'utilisation.” *Wikipedia*, Wikimedia Foundation, 4 Jan. 2020, fr.wikipedia.org/wiki/Diagramme_de_cas_d'utilisation ; consulté le 16 juillet 2020.
- ❖ [34] AUDIBERT, Laurent. “UML 2 - De L'apprentissage à La Pratique.” *Developpez.com*, laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation ; consulté le 20 juillet 2020.
- ❖ [35] Lydie dBousquet cour: GL - 2 Architecture logicielle” “http://membres-lig.imag.fr/dubousquet/docs/2.4_Architecture.pdf ; consulté le 20 juillet 2020.
- ❖ [36] “Architecture Logicielle.” *Wikipedia*, Wikimedia Foundation, 9 Feb. 2020, fr.wikipedia.org/wiki/Architecture_logicielle; consulté le 22 juillet 2020.

- ❖ [37] Yann-Gaël Guéhéneuc © Guéhéneuc, 2012 Département de génie informatique et de génie logiciel École Polytechnique de Montréal “Architectures – Partie 1/2 “: “
[http://www.ptidej.net/courses/log4430/winter12/slides/Cours%201%20-%20Concepti
on%20architecturale%201.pdf](http://www.ptidej.net/courses/log4430/winter12/slides/Cours%201%20-%20Concepti%20on%20architecturale%201.pdf) ; consulté le 22 juillet 2020.
- ❖ [38] Gérard, Pierre, and IUT de Villetaneuse. *UML Cours 1 : Diagrammes De Classes : Associations*, lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours01.html ; consulté le 25 juillet 2020.
- ❖ [39] “Flutter (Software).” *Wikipedia*, Wikimedia Foundation, 4 Oct. 2020, [en.wikipedia.org/wiki/Flutter_\(software\)](https://en.wikipedia.org/wiki/Flutter_(software)). Consulté le 4 octobre 2020.
- ❖ [40] mobile, Hugo ExtratIngénieur Concepteur Développeur. “Flutter : Tout Le Monde En Parle, Mais Pourquoi ?” *Business*, 3 Mar. 2020, www.technologies-ebusiness.com/solutions/pourquoi-flutter. Consulté le 05 octobre 2020.
- ❖ [41] Kopec, David. “Les Meilleurs Livres Dart.” *Developpez.com*, dart.developpez.com/livres/.
- ❖ [42] Alcouffe. “A Quoi Sert Firebase, La Plateforme Mobile De Google?” *Junto*, 28 Apr. 2020, junto.fr/blog/firebase/. Consulté le 05 octobre 2020.
- ❖ [43] “Cloud Firestore Firebase.” *Google*, Google, firebase.google.com/docs/firestore. Consulté le 05 octobre 2020.
- ❖ [44] “Firebase Realtime Database.” *Google*, Google, firebase.google.com/docs/database. Consulté le 10 octobre 2020.
- ❖ [45] “Real-Time Database.” *Wikipedia*, Wikimedia Foundation, 27 June 2020, en.wikipedia.org/wiki/Real-time_database; Consulté le 10 octobre 2020.
- ❖ [46] Atlassian. “Qu'est-Ce Que Git : Devenez Un pro De Git Grâce à Ce Guide.” *Atlassian*, www.atlassian.com/fr/git/tutorials/what-is-git ; Consulté le 12 octobre 2020.

- ❖ [47] Furno |, Nicolas. “Développeurs : GitKraken 2.0 Est Deux Fois plus Rapide.” *MacGeneration*, www.macg.co/logiciels/2017/01/developpeurs-gitkraken-20-est-deux-fois-plus-rapide-97211 ; Consulté 12 octobre 2020.
- ❖ [48] “Documentation for Visual Studio Code.” *RSS*, Microsoft, 14 Apr. 2016, code.visualstudio.com/docs.Etienne ; consulté le 12 octobre 2020.
- ❖ [49] Fakher, et al. “Visual Studio Code.” *Le Blog Technique Webnet*, 23 Oct. 2019, blog.webnet.fr/visual-studio-code/ ; consulté le 12 octobre 2020.

Logiciel de gestion des services d'assistance

est un logiciel applicatif qui permet le suivi de l'activité des services d'assistance dans des organisations vouées à ce type d'activité (centre d'assistance, cellules d'assistance réparties, centre d'appel, ...).

Un centre d'assistance

est un service chargé de répondre aux demandes d'assistance émanant des utilisateurs de produits ou de services et constitue le point de contact principal entre la direction des systèmes d'information et les utilisateurs.

UML

C'est l'acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ». La notation UML est un **langage visuel** qui permet de représenter graphiquement les besoins des utilisateurs, constitué d'un ensemble de schémas, appelés des **diagrammes**, qui donnent chacun une vision différente du projet à traiter.

UML nous fournit donc des diagrammes pour **représenter** le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc.

UML n'est pas une méthode, mais peut être employé dans tout le cycle de développement, indépendamment de la méthode.

Packages utilisés durant le codage de notre application

Firestore_core : fournit des API pour gérer vos instances d'application Firestore et vos informations d'identification. Ce plugin est requis par tous les plugins FlutterFire. la version actuelle est 0.5.0+1.

Firestore_storage : conçu pour vous aider à stocker et diffuser rapidement et facilement du contenu généré par l'utilisateur, tel que des photos et des vidéos.

Location : Ce plugin pour Flutter gère la localisation sur Android et iOS. Il fournit également des rappels lorsque l'emplacement est modifié et obtenir l'emplacement actuel de l'utilisateur.

Google_maps_flutter : permet d'intégrer Google Maps dans les applications iOS, Android et Web. Il s'agit d'un wrapper de google_maps_flutter pour mobile et de google_maps pour le Web.

Cloud_firestore : Cloud Firestore est une base de données de documents NoSQL qui vous permet de stocker, de synchroniser et d'interroger facilement des données pour vos applications mobiles et Web, à l'échelle mondiale.

Firebase_auth : L'authentification Firebase fournit des API faciles à utiliser pour authentifier les utilisateurs auprès de votre application. Il prend en charge l'authentification à l'aide de mots de passe, de numéros de téléphone, de fournisseurs d'identité fédérés populaires tels que Google, Facebook et Twitter, etc.

Shared_Preferences : L'utilisation principale de Shared Preferences est de sauvegarder les préférences de l'utilisateur, les paramètres, peut-être les données (si elles ne sont pas trop grandes) de sorte que la prochaine fois que l'application est lancée, ces informations peuvent être récupérées et utilisées.

PermissionHandler : Ce plugin fournit une API multiplateforme (iOS, Android) pour demander des autorisations et vérifier leur statut. Vous pouvez également ouvrir les paramètres de l'application de l'appareil afin que les utilisateurs puissent accorder une autorisation.

Provider : un package qui facile à utiliser qui fonctionne avec les widgets de bas niveau permettant aux widgets de fournir des données et des services à leurs descendants.

SMS : c'est un plugin pour l'envoi de SMS et MMS sur Android et iOS. Si vous envoyez à plus d'une personne, il sera envoyé sous forme de MMS. Sur iOS, si le numéro est un iPhone et que iMessage est activé, il sera envoyé en tant qu'iMessage.

Geoflutterfire : est une bibliothèque open-source qui vous permet de stocker et d'interroger un ensemble de clés en fonction de leur emplacement géographique. En son cœur, GeoFlutterFire stocke simplement les emplacements avec des clés de chaîne. Son principal avantage, cependant, est la possibilité de récupérer uniquement ces clés dans une zone géographique donnée - le tout en temps réel.