

## Table des matières

|  |           |
|--|-----------|
| <b>INTRODUCTION GÉNÉRALE .....</b>                         | <b>4</b>  |
| <b>CHAPITRE 1 : CONCEPTS GÉNÉRAUX.....</b>                 | <b>5</b>  |
| <b>I. LES APPLICATIONS MOBILES .....</b>                   | <b>5</b>  |
| <b>II. LES TYPES D'APPLICATIONS MOBILES .....</b>          | <b>5</b>  |
| 1. Application native.....                                 | 5         |
| 2. Application web.....                                    | 5         |
| 3. Application hybride .....                               | 6         |
| <b>III. CARACTÉRISTIQUES DES APPLICATIONS MOBILES.....</b> | <b>6</b>  |
| <b>IV. LES DIFFÉRENTES PLATEFORMES.....</b>                | <b>6</b>  |
| 1. Android.....  | 6         |
| 2. iOS .....   | 6         |
| 3. Windows Phone 7.....                                    | 6         |
| <b>V. LE SYSTÈME D'EXPLOITATION ANDROID.....</b>           | <b>7</b>  |
| 1. Les outils.....   | 7         |
| 2. Les exemples de code.....                               | 7         |
| 3. La documentation .....                                  | 8         |
| 4. Les API .....   | 8         |
| <b>VI. CONCEPTS DE LA MUTUELLE MOBILE.....</b>             | <b>8</b>  |
| 1. Quelques définitions liées à la mutuelle .....          | 8         |
| 2. Types d'opérations .....                                | 8         |
| 3. Types d'utilisateurs.....                               | 11        |
| <b>BILAN .....</b>   | <b>11</b> |
| <b>CHAPITRE 2 : ANALYSE.....</b>                           | <b>12</b> |
| <b>I. ANALYSE DES BESOINS.....</b>                         | <b>12</b> |
| 1. Analyse des besoins fonctionnels.....                   | 12        |
| 2. Analyse des besoins non fonctionnels.....               | 12        |
| <b>II. LES ACTEURS DU SYSTÈME.....</b>                     | <b>13</b> |
| <b>III. QUELQUES DIAGRAMMES D'ANALYSE.....</b>             | <b>13</b> |
| 1. Diagramme de contexte .....                             | 13        |

|  |   |           |
|--|---|-----------|
| 2.   | Diagramme de package .....  | 14        |
| 3.   | Diagramme de cas d'utilisation .....  | 14        |
| 4.   | Diagramme de classe.....  | 21        |
| 5.   | Diagramme d'activité .....  | 22        |
| 6.   | Diagramme d'état transition .....   | 25        |
| BILAN .....                                      |   | 26        |
| <b>CHAPITRE 3 : CONCEPTION .....</b>             |   | <b>27</b> |
| I.   | ARCHITECTURE DE LA SOLUTION.....  | 27        |
| 1.   | Architecture de la solution .....   | 27        |
| 2.   | Architecture logicielle.....  | 27        |
| 3.   | Architecture de sécurité .....  | 28        |
| II.  | QUELQUES DIAGRAMMES DE CONCEPTION.....  | 28        |
| 1.   | Diagramme de classe technique.....  | 28        |
| 2.   | Diagramme de séquence technique.....  | 29        |
| BILAN .....                                      |   | 42        |
| <b>CHAPITRE 4 : IMPLÉMENTATION ET TEST .....</b> |   | <b>43</b> |
| I.   | OUTILS UTILISÉS ET TECHNOLOGIES .....   | 43        |
| II.  | RÉSULTATS OBTENUS.....  | 43        |
| 1.   | Les interfaces de bienvenu, connexion et accueil .....  | 43        |
| 2.   | La sidebar .....  | 47        |
| 3.   | Les membres, les responsable des opérations, les types d'aides, la configuration, la session, les exercices, les dettes, les aides et la deconnexion..... | 48        |
| 4.   | La navbar .....   | 52        |
| <b>CONCLUSION GÉNÉRALE .....</b>                 |   | <b>54</b> |
| <b>RÉFÉRENCES .....</b>                          |   | <b>55</b> |

## Table des figures

|  |    |
|--|----|
| Figure 2: diagramme de contexte  | 13 |
| Figure 3: diagramme de package   | 14 |
| Figure 4: diagramme de cas d'utilisation du membre                                     | 15 |
| Figure 5: diagramme de cas d'utilisation coté responsable des opérations               | 18 |
| Figure 6: diagramme de classe  | 22 |
| Figure 7: diagramme activité session   | 23 |
| Figure 8: diagramme activité authentification de l'utilisateur                         | 24 |
| Figure 9: diagramme activité création du membre  | 24 |
| Figure 10: diagramme état transition session   | 25 |
| Figure 11: diagramme état transition exercice  | 25 |
| Figure 12: diagramme état transition de membre   | 26 |
| Figure 13: diagramme état transition emprunt   | 26 |
| Figure 14: architecture logicielle   | 27 |
| Figure 15: diagramme de séquence technique   | 29 |
| Figure 16: diagramme de séquence technique s'authentifier                              | 30 |
| Figure 17: diagramme de séquence technique ajouter responsable des opérations          | 31 |
| Figure 18: diagramme de séquence technique ajouter membre                              | 32 |
| Figure 19: diagramme de séquence technique ajouter type d'aide                         | 33 |
| Figure 20: diagramme de séquence technique modifier la configuration                   | 34 |
| Figure 21: : diagramme de séquence technique modifier profil                           | 35 |
| Figure 22: diagramme de séquence technique suppression d'un responsable des opérations | 35 |
| Figure 23: diagramme de séquence technique suppression de membre                       | 36 |
| Figure 24: diagramme de séquence technique valider une aide                            | 36 |
| Figure 25: diagramme de séquence technique valider emprunt                             | 37 |
| Figure 26: diagramme de séquence technique valider remboursement                       | 37 |
| Figure 27: diagramme de séquence technique valider contribution                        | 38 |
| Figure 28: authentification  | 39 |
| Figure 29 : diagramme de séquence technique demander une aide                          | 40 |
| Figure 30 : diagramme de séquence technique demander une contribution                  | 40 |
| Figure 31 : diagramme de séquence technique demander un emprunt                        | 41 |
| Figure 32 : diagramme de séquence technique modifier profil                            | 41 |
| Figure 33: interface de bienvenue  | 44 |
| Figure 34: interface de login  | 45 |
| Figure 35: interface d'accueil   | 46 |
| Figure 36: interface sidebar   | 47 |
| Figure 37: interface d'aide  | 48 |
| Figure 38: interface de dettes   | 49 |
| Figure 39: interface session   | 50 |
| Figure 40: interface liste des membres   | 51 |
| Figure 41: interface responsable des opérations  | 52 |
| Figure 42: La navbar   | 53 |

## INTRODUCTION GÉNÉRALE

Avec l'indispensabilité de la communication et de l'information dans notre vie quotidienne et grâce à la généralisation des téléphones portables tactiles à écrans large ainsi qu'au développement des logiciels et des réseaux, la recherche d'information est accessible du bout des doigts par l'intermédiaire de téléphone ou pads à tout moment et en tous lieux. Il est important pour tout individu de se tenir au courant de ce qui se passe autour de lui et d'avoir accès aux informations dont il a besoin, quand il en a besoin plus facilement et plus rapidement.

Les applications mobiles sont capables de satisfaire un large éventail de besoins. Elle avoir une rapidité d'accès à l'information ainsi que la possibilité de profiter de plusieurs fonctionnalités novatrices, ludique et intuitives qui repoussent les limites de l'expérience utilisateur comme : l'appareil photo, vidéo, le GPS... en plus elles offrent une interface plus spécifique et plus harmonieuse qui les rendre plus aisé que les sites mobiles dans leur utilisation. Pour cette raison l'idée de développement une application devient utile et conviviale.

Dans le domaine de la finance, très souvent une personne a des difficultés pour gérer son épargne, ses remboursements, sa mutuelle, .... Donc il est important pour des développeurs de se pencher sur la résolution de ce problème. Il s'agit au cours de notre projet d'étudier et de développer une application sous Android comme une solution simple permettant de gérer sa mutuelle en quelques clics.

Ce document est organisé en cinq chapitres ; le premier présente les différentes notions utilisées dans le développement des applications mobiles à savoir, le système d'exploitation Android, la mutuelle et les bases de données et nous présentons notre méthodologie c'est-à-dire comment nous avons modélisé le problème à résoudre et la méthode de gestion de projet dont on fait usage. Le deuxième chapitre porte sur l'analyse du projet à travers une analyse des besoins, une détermination des acteurs du système, une modélisation de quelques diagrammes d'analyse (contexte, package, cas d'utilisation, classe et activité, état transition). Le chapitre trois décrira la conception du système à travers une présentation de l'architecture de la solution, quelques diagrammes de conception (diagramme de classe technique, diagramme de séquence technique). Nous passerons ensuite au chapitre quatre qui est une présentation de l'implémentation et les tests à travers les outils et les technologies utilisés ainsi que des résultats obtenus.

## CHAPITRE 1 : CONCEPTS GÉNÉRAUX

Avec l'évolution de la stratégie mobile et la tendance actuelle à l'utiliser de plus en plus dans la vie quotidienne, nombreux sont les secteurs d'activité à l'exploiter pour se développer. Les mutuelles sont des sociétés de personnes à but non lucratif organisant la solidarité entre leurs membres, et dont les fonds proviennent principalement des cotisations des membres. Dans ce chapitre nous présentons les différentes notions de l'environnement Android et ses fonctionnalités.

### I. LES APPLICATIONS MOBILES

Une application mobile est une application embarquée qui est sensé s'exécuter sur une plateforme mobile à l'instar des Smartphones et les tablettes. L'application doit être téléchargée par l'utilisateur pour devenir exécutable à partir du système d'exploitation du téléphone exactement comme un logiciel ordinaire qui s'installe sur un ordinateur classique. Les applications mobiles ne sont pas transférables d'un système d'exploitation à l'autre en raison du langage informatique utilisé : si l'on télécharge une application Android, l'on ne pourra l'utiliser que sur un téléphone portable Android. Reflet du succès commercial et technique de l'iPhone, une grande majorité des applications mobiles furent créées à l'origine pour le téléphone mobile d'Apple. En ce sens, les applications mobiles se regroupent en plusieurs séries suivant des critères basiques :

- **Applications fonctionnant sans internet**
- **Applications exigeant une connexion**
- **Applications connectées**
- **Applications interagissant avec les autres équipements de Smartphone**
- **Applications interagissant avec d'autres mobinautes**

### II. LES TYPES D'APPLICATIONS MOBILES

Techniquement parlant, il y a trois types d'application mobile que tout utilisateur peut rencontrer :

#### 1. Application native

Il s'agit d'application conçue pour une grande partie de systèmes d'exploitation fiables par les Smartphones en se référant à un langage particulier à chacun d'eux. Le développement de l'application native nécessite le recours à la mémoire du Smartphone sans omettre les options reliées au système d'exploitation en question. De cette façon, le résultat se résume dans l'aboutissement à des applications mobiles avec des fonctions plus professionnelles, développées et performantes au même niveau que les applications en HTML5/CSS3 et les applications hybrides.

#### 2. Application web

Toute application conçue avec HTML et CSS de plus opérationnelle sur navigateur internet pour un Smartphone est appelée application web. Peu importe la marque du Smartphone, l'on peut accéder à l'application web par le biais de son navigateur et donc pas besoin de la télécharger. Vu qu'elle ne tient pas en compte les divergences persistantes entre les systèmes d'exploitation et les marques de Smartphone, l'application web manque d'ergonomie et de plus elle ne se sert pas de la mémoire du Smartphone ce qui la place en infériorité par rapport à l'application native.

### 3. Application hybride

Il s'agit d'une application mobile qui fusionne entre les caractéristiques de l'application web (développement en HTML 5) et celles de l'application native. De cette manière, l'application mobile sera accessible sur toutes les plateformes d'application. Ce type d'application mobile minimise les charges et la durée de son développement même si cela sera au détriment du perfectionnement et de la qualité qui caractérise l'application native. Notons que les applications hybrides sont accessibles exclusivement sur iPhone et Android.

## III. CARACTÉRISTIQUES DES APPLICATIONS MOBILES

- **Une application mobile est plus rapide**
- **Une application offre plus de possibilités**
- **C'est plus difficile à développer**
- **Il faut développer pour chaque plateforme**
- **On gagne plus avec une application**
- **Le déploiement est plus compliqué, plus long**
- **Pas besoin d'être connecté pour utiliser mon application**

## IV. LES DIFFÉRENTES PLATEFORMES

Pour connaître le leader dans le marché des Smartphones et déterminer le système le plus répondant aux besoins de notre application, nous allons présenter brièvement chacun de ces systèmes et décrire leurs avantages et inconvénients :

### 1. Android

Android est un système d'exploitation ouvert (Open Source) pour tablettes tactiles, terminaux mobiles et TV connectées. Il a été conçu en 2007, par la société Android, une startup rachetée par Google. C'est un système d'exploitation fondé sur un noyau Linux disponible grâce à une licence Apache. Android inclut tous les utilitaires requis par un constructeur pour le mettre en œuvre dans un téléphone portable. Il est proposé à tous les fabricants de téléphones mobiles pour faciliter son adoption.

### 2. iOS

Anciennement iPhone OS, est le système d'exploitation mobile développé par Apple pour l'iPhone. Reconnu pour sa fluidité, son ergonomie et son intuitivité : c'est le système d'exploitation le plus abouti à ce jour. Il dispose du portail App Store, qui avec un catalogue de plus de 200 000 applications, s'est imposé comme une référence parmi les kiosques d'applications mobiles. Après Android, iOS est le deuxième système d'exploitation le plus répandu. Il équipe uniquement les « iPhone », c'est-à-dire les Smartphones de la marque Apple.

### 3. Windows Phone 7

Dernier né des systèmes d'exploitation, Windows Phone 7 se différencie de ses concurrents par son système de hub qui regroupe tous les éléments d'une même catégorie (contacts, photos, jeux...). Par exemple, lorsque vous vous rendez sur le blog contacts, vous pourrez avoir accès aux derniers appels, messages et photos Facebook.... Windows Phone 7 est très exigeant sur les spécifications des

Smartphones : la mémoire et le processeur doivent ainsi atteindre des seuils minimums de performance. Des caractéristiques qui rendent la navigation fluide et agréable.

## V. LE SYSTEME D'EXPLOITATION ANDROID

Nous avons décrit grosso modo les différents systèmes d'exploitation utilisés par les applications mobiles. Dans cette section, nous allons mettre le point sur la plateforme qu'on a choisi pour développer notre application. Nous avons choisi le système d'exploitation Android pour les raisons suivantes :

- Android est une plateforme Open Source, puissante, moderne, sûre et ouverte. Grâce à l'ouverture du code source et des APIs, les développeurs obtiennent la permission d'intégrer, d'agrandir et de modifier les composants existants. Les utilisateurs peuvent adapter les applications à leur besoin.
- Android est basé sur le noyau Linux. Alors, y a plusieurs avantages comme une grande mémoire, la gestion de processus, le modèle de sécurité, le soutien de bibliothèque partage, etc.
- Le SDK de l'Android offre complètement les APIs pour développer l'application sur Android.

Grâce au « Android Développeur Challenge » que Google fournira 10 millions de dollars en prix, aucune condition pour les applications sur la plateforme Android, les développeurs ont eu l'opportunité de gagner beaucoup d'argent.

Exploiter une nouvelle plate-forme n'est jamais été une chose aisée. C'est pourquoi Google fournit, en plus du système d'exploitation, un kit de développement (Software Développment Toolkit ou SDK). Ce SDK est un ensemble d'outils qui permet aux développeurs et aux entreprises de créer des applications. Il est disponible gratuitement sur le site de Google. Le SDK Android est composé de plusieurs éléments afin d'aider les développeurs à créer et à maintenir des applications :

- Des outils
- Des exemples de code
- De la documentation.
- Des API (interfaces de programme d'application).

### 1. Les outils

Le SDK est livré avec un certain nombre d'outils couvrant différents aspects du cycle de développement d'une application Android. Le kit de développement propose une boîte à outils complète pour les tâches de compilation, de débogage, de génération de code AIDL et de la signature de l'application, etc. L'émulateur Android : c'est un téléphone virtuel qui permet de tester les applications qui sont en train de se développer. Il est lancé par la commande "emulator". Celle-ci prend en paramètre l'image AVD (Android Virtual Device) qui sera montée en mémoire. Il a des limitations par exemple : il n'est pas capable de supporter le Bluetooth ainsi qu'il ne permet pas le teste des applications de réalité augmentée.

### 2. Les exemples de code

Le kit de développement est accompagné d'un certain nombre d'exemples illustrant les possibilités du SDK Android. Parmi ces exemples, on peut citer : un jeu du serpent et le projet qui couvre l'utilisation de plusieurs exemples de l'API Android comme les alarmes, les notifications et les menus.

### 3. La documentation

La documentation du SDK Android est scindée en deux parties bien distinctes :

- Le guide du développeur qui est disponible en HTML (Hypertext Markup Language) dans le répertoire du SDK qu'on vient d'installer.
- La documentation des API au format java doc est également située dans le répertoire docs et accessible grâce au chemin débutant du répertoire d'installation.

### 4. Les API

Android offre plusieurs API (Application Programming Interface) pour alléger le travail de programmation.

## VI. CONCEPTS DE LA MUTUELLE MOBILE

### 1. Quelques définitions liées à la mutuelle

#### a. Trésorerie

La trésorerie est générée par les épargnes des différents adhérents et est mis à disposition lors des différentes sessions pour d'éventuels emprunts.

#### b. Fond social

Il permet de venir en aide aux adhérents qui doivent faire face à un événement grave ou exceptionnel. Le plus souvent, ce fonds permet d'intervenir de façon ponctuelle sur un financement de dépenses médicales ou sur la participation de frais d'obsèques. Ce fond est entretenu par les contributions obligatoires et les inscriptions des adhérents.

#### c. Exercice et Session

Un exercice représente un cycle des activités de la mutuelle sur un an. Un exercice est caractérisé par l'année et comporte 12 sessions (une par mois). En effet, la session est effectuée un jour d'un mois et c'est durant une session qu'on effectue toutes les opérations de la mutuelle (épargne, aide, emprunt, contribution ...). Cependant une demande d'aide ou d'emprunt peut être faite hors de la session mais ne sera validée ou refusée que pendant la prochaine session.

### 2. Types d'opérations

De nombreuses opérations constituent le fonctionnement de la mutuelle et en tant qu'application à laquelle on doit faire confiance, il faut expliquer clairement comment l'application traite ces opérations.

#### a. Inscription

Afin d'utiliser les services offerts par mutuelle, chaque utilisateur (membre ou responsable des opérations) doit se connecter. Ceci a l'avantage de fournir à l'utilisateur une vue personnalisée.

Lors de la première installation de l'application, seul l'utilisateur responsable des opérations peut se connecter avec un email et un mot de passe qui lui ont été communiqués par les développeurs de l'application. L'utilisateur responsable des opérations se connecte avec ces identifiants initiaux, modifie son profil en changeant



également cet email et ce mot de passe initial par un mot de passe fort de son choix, qu'il utilisera à partir de ce moment.

Pour chaque membre, la première connexion se fait avec l'email et le mot de passe qui lui ont été donnés par l'responsable des opérations. Le responsable des opérations inscrit les nouveaux membres de la mutuelle en saisissant des informations concernant le membre en question : nom, email, mot de passe, adresse dans la section "Membres" de l'application. Pour chaque nouveau membre ajouté, un montant d'inscription est versé par le membre qui est ajouté au fond social de la mutuelle. Lors de la première connexion, le membre doit utiliser l'email et le mot de passe qui lui ont été donnés par l'responsable des opérations. Après s'être connecté, le membre peut modifier son profil ainsi que l'email et le mot de passe initiaux pour en créer de nouveaux qu'il utilisera pour les connexions suivantes.

Chaque fois qu'un membre ou un responsable des opérations se connecte, en entrant son email et son mot de passe, une requête de type "post" est envoyée au backend avec l'email et le mot de passe comme "payload" de la requête. Lorsqu'une correspondance est trouvée dans la base de données, un token est renvoyé avec la réponse, ce token est conservé dans le cache de stockage de l'application mutuelle et est utilisé pour identifier l'utilisateur, l'utilisateur reste connecté aussi longtemps que ce token est dans le cache. Lorsque l'utilisateur se déconnecte de l'application enspy mutuelle, le token est supprimé du cache. Il lui sera alors présenté la page de connexion la prochaine fois qu'il lance l'application.

#### b. Contribution

Enspy mutuelle a deux types de contributions : les contributions obligatoires et les contributions pour les aides (facultatives) qui augmentent toutes deux les fonds sociaux.

##### i. Contributions obligatoires

Les contributions obligatoires doivent être versées par tous les membres de la mutuelle lors de chaque session. Le montant de la cotisation obligatoire est fixé par n'importe quel responsable des opérations dans l'option "configurations de la mutuelle" de l'application. Il s'agit d'une source fondamentale de fonds sociaux pour la mutuelle.

##### ii. Contribution facultatives (contribution d'aide)

Les membres de la mutuelle peuvent également contribuer lorsque d'autres membres ont besoin d'aide. Un responsable des opérations peut créer un type d'aide (exemple : un membre malade, le mariage d'un membre, tout problème d'aide qui se présente), choisir le membre concerné par l'aide, préciser la date limite des contributions et valider l'aide. Tous les membres de la mutuelle reçoivent la notification contenant les détails de l'aide spécifiée par l'responsable des opérations. Tous les membres de la mutuelle sont informés des personnes qui ont payé pour l'aide en question, du montant qu'elles ont contribué, de la date de leur paiement et de l'responsable des opérations qui a enregistré leur don.

#### c. Aide

Enspy mutuelle existe pour aider ses membres. Différents types d'aide (Mariage d'un membre, décès, naissance, et autres) peuvent être créés par un responsable des opérations dans l'option "aides" de l'application. Il spécifie le montant à donner pour chaque aide créée. Par défaut, une partie de l'argent des fonds sociaux sera prise pour soutenir un scénario d'aide qui se présente ; des contributions optionnelles peuvent également être ouvertes par un responsable des opérations. Les membres

donnent ce qu'ils veulent, lorsque la date limite de don est atteinte, l'aide est fermée et le montant collecté est remis à la personne concernée par l'aide.

#### *d. Epargne*

L'épargne est réalisée dans le but d'obtenir des intérêts de la part de ceux qui empruntent. Pour chaque emprunt effectué, les intérêts à payer pour cet emprunt sont répartis entre ceux qui ont épargné. Le montant que reçoit chaque épargnant dépend de la proportion du montant total épargné qui lui appartient. À la fin de l'exercice, ils reçoivent leur épargne plus les intérêts qu'ils ont gagnés sur l'épargne tout au long de l'exercice.

L'enregistrement des épargnes des utilisateurs se fait pendant les sessions. Il suffit à l'responsable des opérations de choisir le nom du membre qui souhaite épargner de l'argent et de saisir le montant qu'il souhaite épargner dans la section "Epargne" de l'application.

#### *e. Emprunt*

Les membres de la mutuelle qui traversent une crise financière ou qui ont un besoin urgent d'argent peuvent emprunter de l'argent à la mutuelle avec un intérêt sur le paiement. Le taux d'intérêt est fixé par un responsable des opérations.

Pour que l'emprunt d'un membre soit confirmé :

- Le membre doit avoir déjà donné sa contribution mensuelle pendant la session en cours.
- Le membre ne doit pas avoir de dette (s'il a emprunté dans le passé, il doit les avoir entièrement remboursés avec leurs intérêts).
- Le montant que le membre emprunte ne doit pas être supérieur à ce qui se trouve dans la trésorerie.

Un membre peut émettre une demande d'emprunt à partir de son interface, qu'un responsable des opérations voit de son côté. Lors d'une session, l'responsable des opérations saisit simplement le nom du membre et le montant qu'il souhaite emprunter. L'application vérifie que le membre en question respecte les trois points mentionnés ci-dessus avant que son emprunt ne soit enregistré. Chaque emprunt doit être payé avec les intérêts au plus tard à la fin de l'exercice.

Un responsable des opérations peut mettre à jour la date de remboursement des emprunts dans l'option "configurations mutuelles" de l'application. La date de remboursement des emprunts est fixée par défaut à :

- 3mois : pour un montant d'emprunt compris entre 0 frs CFA et 300 000 frs CFA
- 6mois : pour un montant d'emprunt compris entre 300 frs CFA et 600 000 frs CFA
- Fin d'exercice : pour un montant d'emprunt supérieur à 600 000 frs CFA

Lorsqu'un emprunt est effectué, la date limite de remboursement est automatiquement calculée.

#### *f. Remboursement*

Les membres doivent rembourser l'argent emprunté avec les intérêts et avant la date limite. Enspy mutuelle rend l'opération de remboursement simple. Il suffit à l'responsable des opérations d'aller dans la section "remboursement" de l'application, de choisir le nom du membre qui rembourse et d'entrer le montant qu'il a donné. Des vérifications sont effectuées pour s'assurer que le membre en question a

réellement emprunté et que le montant qu'il rembourse n'est pas supérieur à ce qui est attendu de lui. Tous les montants empruntés et les intérêts ne sont pas obligés d'être remboursés une seule fois, ils peuvent être séparés, mais ils doivent être remboursés avant la date limite.

### 3. Types d'utilisateurs

#### a. L'administrateur principal

Dans notre application de mutuelle, l'administrateur principal est celui-là qui ajoute un responsable des opérations, le supprime et peut aussi changer le mot de passe d'un membre en cas de perte.

#### b. Responsable des opérations

Le responsable de la mutuelle peut ajouter des membres, les supprimer, valider les demandes.

#### c. Membre

Le membre est celui-là pour qui l'application est conçue. Il peut jouir de toutes les fonctionnalités offertes par l'application (pour un membre) mais ne peut ni ajouter de membre, ni ajouter d'administrateur des opérations, ni en supprimer.

## BILAN

Dans ce chapitre, nous avons présenté quelques concepts généraux et aussi la façon dont nous avons procédé à l'analyse de la mutuelle mobile.

## CHAPITRE 2 : ANALYSE

Après l'émergence des technologies mobiles beaucoup d'applications ont été développées dans divers domaines. Cependant, nous avons pensé à utiliser ces gadgets pour créer une application pour aider chaque individu gérer sa mutuelle depuis son Smartphone. Ce type d'application métier s'avère très utile non seulement afin de subvenir aux besoins des utilisateurs mais aussi pour les avantages qui présente pour les mutuelles. Il s'agit de :

- Epargner de l'argent pour plus tard en cas de besoin.
- Emprunter de l'argent
- Payer ses dettes
- Prêter de l'argent
- ...

Dans ce chapitre nous présentons comment nous avons procédé à l'analyse de notre application à travers un recueil des besoins (fonctionnels et non fonctionnels), une présentation des acteurs puis quelques diagrammes d'analyse (contexte, package, cas d'utilisation, classe et activité).

### I. ANALYSE DES BESOINS

#### 1. Analyse des besoins fonctionnels

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée / sortie du Système. Dans le cas présent, le système doit permettre :

- S'authentifier
- Demander une aide
- Demander un prêt (emprunter)
- Faire une épargne
- Faire une contribution
- Consulter une session
- Consulter un emprunt
- Consulter une aide
- Consulter un exercice
- Consulter une contribution
- Modifier son profil
- Valider toutes les demandes (aides, emprunt)
- Supprimer un membre
- Ajouter des types d'aide
- Modifier les configurations
- Supprimer un responsable des opérations

#### 2. Analyse des besoins non fonctionnels

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation (langage de programmation, type SGBD, de système d'exploitation...). Dans le cadre

de ce travail, l'application devra être tout d'abord extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités. En plus, l'application devra :

- Tourner en réseau
- Demarrer en un faible intervalle de temps (moins d'une minute)
- Proposer un design ergonomique et accessible
- Disponible tous les jours à toute heure
- Être développée en React Native
- Tenir compte des normes de sécurité
- Respecter la vie privée de l'utilisateur

## II. LES ACTEURS DU SYSTÈME

Dans le cas de notre application, nous avons recensé trois acteurs :

- Le membre
- Le responsable des opérations
- L'administrateur principal

## III. QUELQUES DIAGRAMMES D'ANALYSE

### 1. Diagramme de contexte

Le diagramme de contexte décrit comment les entités externes interagissent avec un système logiciel interne, en l'occurrence ici notre application ENSPY MUTUELLE. Il est principalement utilisé pour aider à comprendre l'étendue d'un système. Le diagramme de contexte est un diagramme de haut niveau, ce qui signifie qu'ils n'entrent pas dans le détail des tenants et aboutissants du système. En tenant compte de tout ceci, voici ci-dessous notre diagramme de contexte :

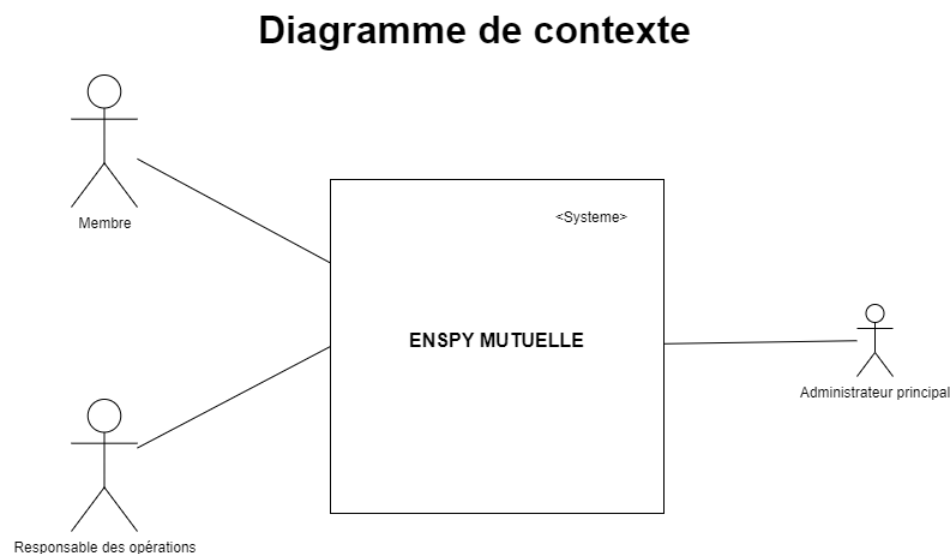


Figure 1: diagramme de contexte

Ici, nos acteurs principaux sont le membre de la mutuelle et le responsable des opérations dont les rôles ont été expliqué dans le chapitre précédent. L'acteur secondaire est l'administrateur principal.

## 2. Diagramme de package

Le diagramme de package est un diagramme structurel utilisé pour montrer l'organisation et l'arrangement de divers éléments de modèle sous forme de packages. Un package est un regroupement d'éléments UML apparentés, tels que des diagrammes, des documents, des classes ou même d'autres packages. Chaque élément est imbriqué dans le package, qui est représenté comme un dossier de fichiers dans le diagramme, puis disposé hiérarchiquement dans le diagramme. Dans le cas de notre mutuelle mobile, nous avons pu organiser notre système en sept grands packages : gestion des exercices, gestion des épargnes, gestion des remboursements, gestion des emprunts, gestion du fond social, gestion des membres, gestion des aides. Ci-dessous est joint notre diagramme de package.

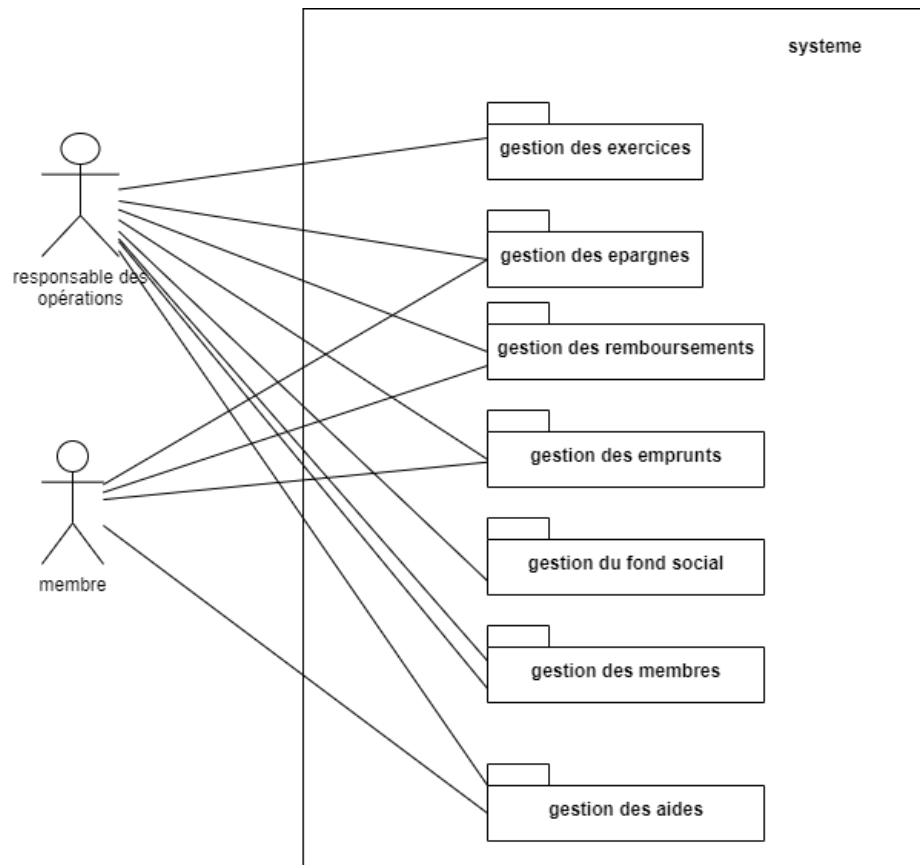


Figure 2: diagramme de package

## 3. Diagramme de cas d'utilisation

Le but d'un diagramme de cas d'utilisation en UML est de démontrer les différentes façons dont un utilisateur peut interagir avec un système. Dans notre cas, nous avons deux diagrammes de cas

d'utilisation pour l'application enspy mutuelle : un diagramme de cas d'utilisation pour le membre, un diagramme de cas d'utilisation pour le responsable des opérations.

a. Diagramme de cas d'utilisation pour le membre

i. Le diagramme

Les cas d'utilisations du membre sont résumés dans le diagramme de cas d'utilisation suivant :

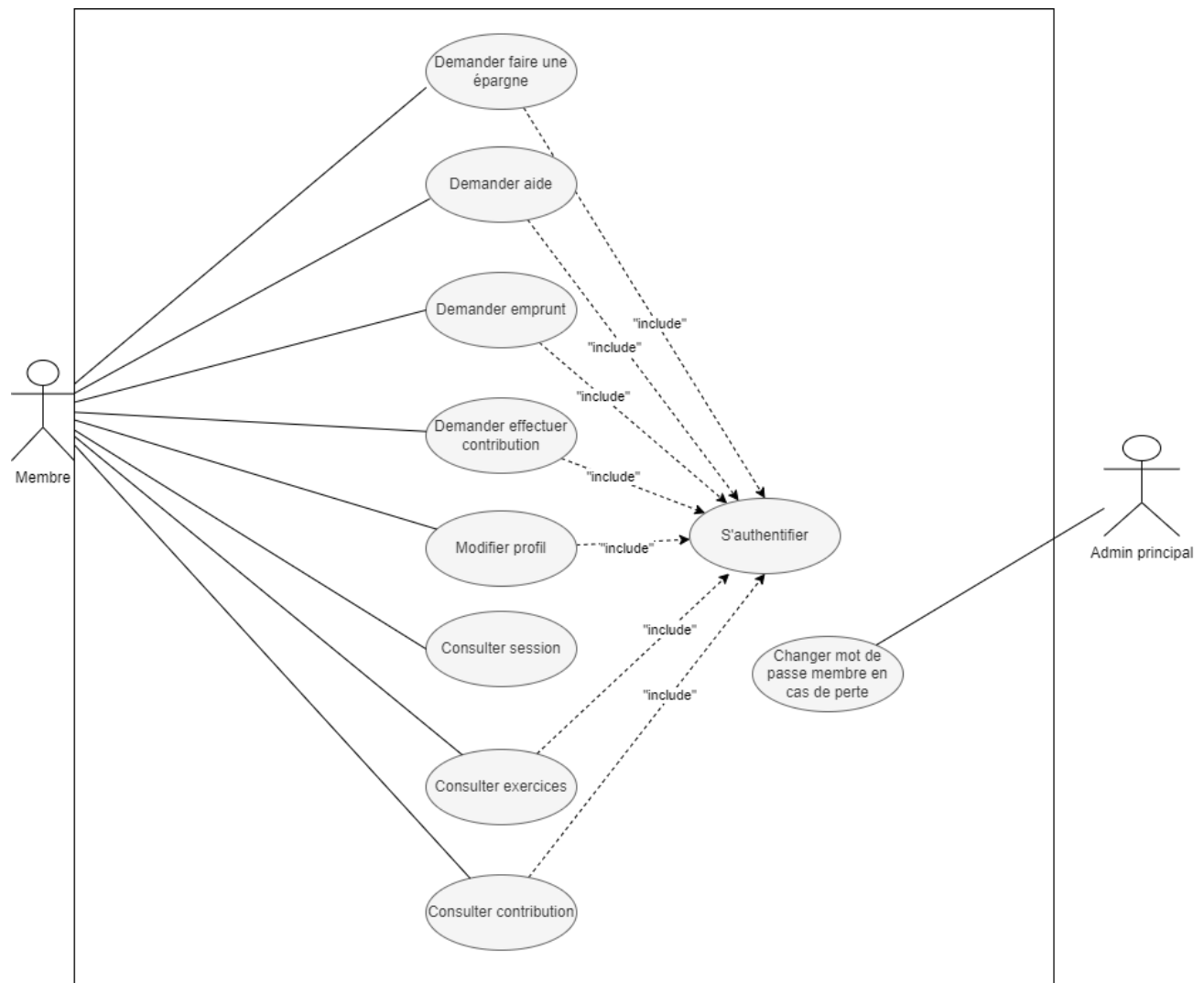


Figure 3: diagramme de cas d'utilisation du membre

ii. La description textuelle

S'authentifier

- **Objectif** : il s'agit pour un membre d'entrer son email et son mot de passe afin de se connecter à l'application.
- **Acteurs**: membre
- **Préconditions** : Le membre doit avoir été ajouté par un admin ou par l'admin principale.

- **Scénario nominal** : Le membre appuie sur le bouton login et est redirigé vers la page d'authentification. Il entre son email et son mot de passe, puis valide. Si les identifiants sont corrects, il a accès à la vue de l'application correspondant à son rôle. Sinon l'authentification échoue.

#### Demander une aide

- **Objectif** : il s'agit pour un membre de demander une aide
- **Acteur** : membre
- **Préconditions** : il doit être connecté
- **Postconditions** : la demande d'aide est effectivement créée.
- **Scénario nominal** : Le membre se rend sur la page d'accueil, puis clique sur le bouton « faire demande d'aide », remplit les champs et valide.

#### Demander un prêt (emprunter)

- **Objectif** : il s'agit pour un membre de demander un prêt (faire un emprunt)
- **Acteur** : membre
- **Préconditions** : il doit être connecté
- **Postconditions** : la demande de prêt est effectivement créée.
- **Scénario nominal** : Le membre se rend sur la page d'accueil, puis clique sur le bouton « faire demande d'emprunt », remplit les champs et valide.

#### Demander effectuer une contribution

- **Objectif** : il s'agit pour un membre de demander à faire une contribution à une aide
- **Acteur** : membre
- **Préconditions** : connaître un responsable des opérations
- **Postconditions** : la demande d'aide est effectivement créée.
- **Scénario nominal** : Le membre demande à l'responsable des opérations de le faire.

#### Modifier son profil

- **Objectif** : modifier son profil
- **Acteur** : Le membre
- **Précondition** : Le membre doit être connecté.
- **Postcondition** : Le profil est effectivement modifié.
- **Scénario nominal** : Le membre se connecte sur l'application de mutuelle mobile, puis clique sur la sidebar, puis sur « profil ». Il entre les nouveaux paramètres dans les champs puis clique sur « sauvegarder la modification ».

#### Consulter exercice

- **Objectif** : Consulter l'exercice en cours et les exercices précédents.
- **Acteur** : Le membre.
- **Précondition** : Le membre doit être connecté.
- **Postcondition** : Les exercices lui sont effectivement présentés.
- **Scénario nominal** : Le membre se connecte sur l'application de mutuelle mobile, puis clique sur la sidebar, puis sur « Exercices ». L'exercice en cours lui est ainsi affiché, tout comme les exercices précédents.



#### Consulter session

- **Objectif** : il s'agit pour un participant de consulter les détails d'une session
- **Acteur** : Un membre
- **Préconditions** : La session doit déjà être terminée et le participant doit être connecter
- **Scénario nominal** :\_Le membre se rend la rubrique sessions. Il consulter la ou les session(s) qu'il souhaite. Il ferme la rubrique

#### Consulter sa contribution

- **Objectif** : il s'agit pour un membre de consulter sa contribution
- **Acteur** : membre
- **Préconditions** : il doit être connecté
- **Postconditions** : la contribution est effectivement affichée.
- **Scénario nominal** :\_Le membre clique sur le bouton contribution de la navbar, puis consulte sa contribution.

#### *b. Diagramme de cas d'utilisation pour le membre*

##### *i. Le diagramme*

Les cas d'utilisations de l'utilisateur des opérations sont résumés dans le diagramme de cas d'utilisation suivant :

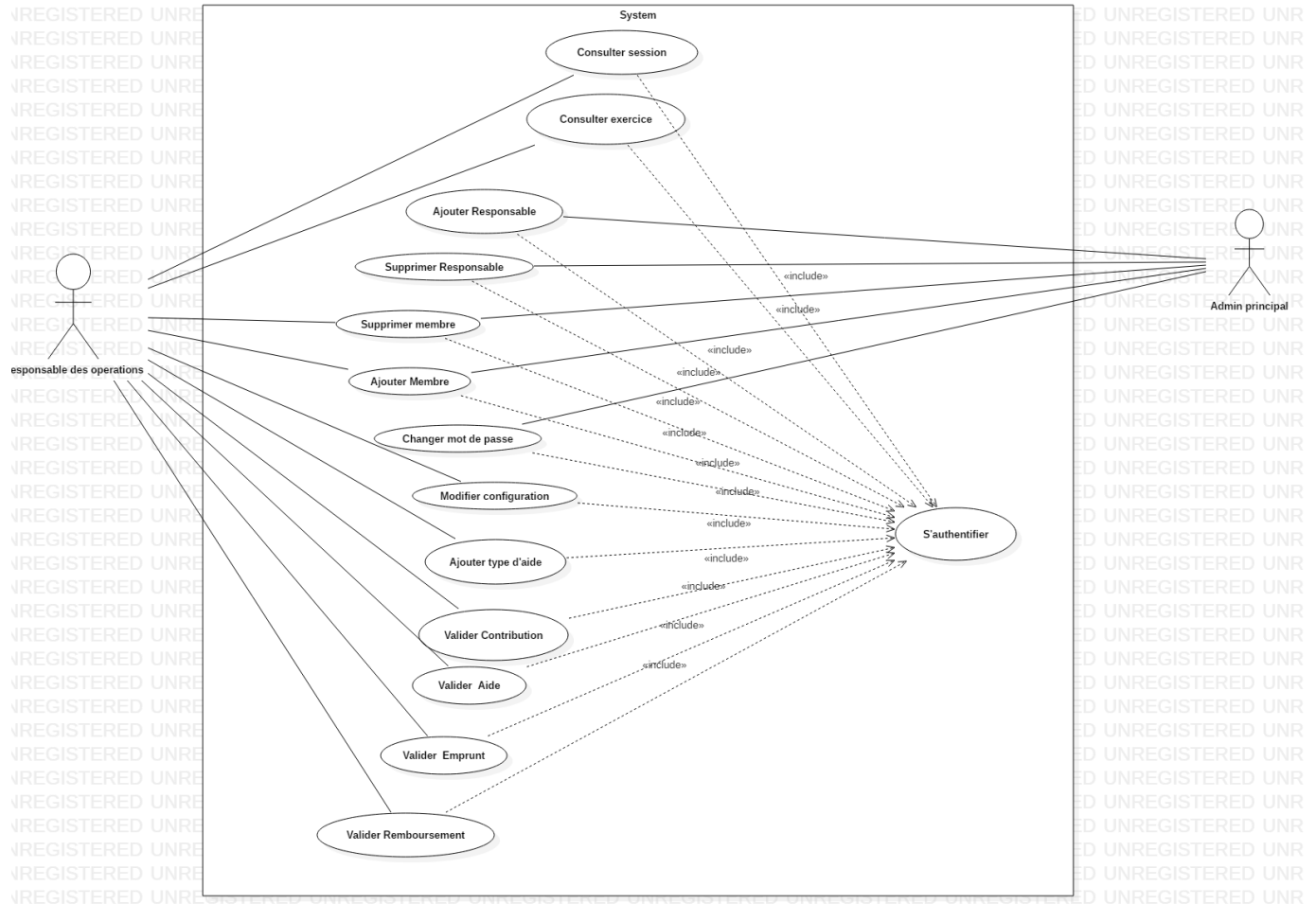


Figure 4: diagramme de cas d'utilisation coté responsable des opérations

## ii. La description textuelle

### S'authentifier

- **Objectif** : il s'agit pour un responsable des opérations d'entrer son email et son mot de passe afin de se connecter à l'application.
- **Acteurs**: responsable des opérations
- **Préconditions** : Le responsable des opérations doit avoir été ajouter par un admin ou par l'admin principale.
- **Scénario nominal** : Le responsable des opérations appuie sur le bouton login et est redirigé vers la page d'authentification. Il entre son email et son mot de passe, puis valide. Si les identifiants sont corrects, il a accès à la vue de l'application correspondant à son rôle. Sinon l'authentification échoue

### Valider une aide

- **Objectif** : il s'agit pour le responsable des opérations de valider une aide qui concerne un membre de la mutuelle
- **Acteur** : responsable des opérations

- **Préconditions** : Le membre concerné doit avoir été enregistré par un admin et avoir soumis sa demande, le fond social doit être suffisant.
- **Postconditions** : L'aide est associée au membre, le fond social est diminué du montant correspondant à l'aide.
- **Scénario nominal** : Le responsable des opérations se rend dans la rubrique aide et clique sur ajouter. Il choisit le membre concerné et le type d'aide. Il ajoute également les contributions personnelles des membres et valide. Si le membre est en règle l'aide est octroyée. Sinon l'aide est refusée

#### Valider un prêt (emprunter)

- **Objectif** : il s'agit pour l'responsable des opérations de valider un emprunt demandé par un membre de la mutuelle.
- **Acteur** : responsable des opérations.
- **Préconditions** : Le membre concerné doit avoir été enregistré par un admin, la trésorerie doit être suffisante, se fait uniquement lors d'une session, la demande doit avoir été faite.
- **Postconditions** : L'emprunt est créé et associé au membre, avec un taux de 1% et un délai de remboursement de 3 mois et la trésorerie est diminuée du montant correspondant à l'emprunt.
- **Scénario nominal** : Le responsable des opérations se rend dans la rubrique emprunt et clique sur ajouter. Il choisit le membre concerné et le montant de l'emprunt. Si le membre est en règle, le montant de l'emprunt est inférieur à la mise du membre alors l'emprunt est octroyé. Sinon l'emprunt est refusé.

#### Valider un remboursement

- **Objectif** : il s'agit pour le responsable des opérations de valider un remboursement sur un emprunt qui concerne un participant de la mutuelle.
- **Acteur** : responsable des opérations
- **Préconditions** : Le membre qui veut rembourser doit initier le remboursement
- **Postconditions** : Le remboursement est valide et va dans la trésorerie
- **Scénario nominal** : Le responsable des opérations reçoit une notification du a l'initiation du remboursement. Il décide de valider le remboursement si les conditions sont remplies. Sinon il annule le remboursement.

#### Ajouter type d'aide

- **Objectif** : il s'agit pour le responsable des opérations d'ajouter une aide possible pour la mutuelle (Maladie, Décès...)
- **Acteur** : responsable des opérations
- **Préconditions** : Le responsable des opérations doit être connecté
- **Postconditions** : L'aide est ajoutée dans les rubriques des aides
- **Scénario nominal** : Le responsable des opérations se rend dans la rubrique des aides. Il clique sur ajouter une nouvelle aide. Il remplit les paramètres de la nouvelle aide. Il valide la création de l'aide

#### Modifier son profil

- **Objectif** : modifier son profil
- **Acteur** : le responsable des opérations
- **Précondition** : Le responsable des opérations doit être connecté.
- **Postcondition** : Le profil est effectivement modifié.
- **Scénario nominal** : Le responsable des opérations se connecte sur l'application de mutuelle mobile, puis clique sur la sidebar, puis sur « profil ». Il entre les nouveaux paramètres dans les champs puis clique sur « sauvegarder la modification ».

#### Modifier la configuration

- **Objectif** : changer les paramètres globaux de l'application
- **Acteur** : Uniquement l'responsable des opérations
- **Précondition** : Le responsable des opérations doit être connecté
- **Postcondition** : Les paramètres de l'application sont effectivement modifiés
- **Scénario nominal** : Le responsable des opérations se connecte sur l'application de mutuelle mobile, puis clique sur la sidebar, puis sur « configuration ». Il entre les nouveaux paramètres dans les champs puis clique sur « sauvegarder la configuration ».

#### Consulter l'exercice

- **Objectif** : Consulter l'exercice en cours et les exercices précédents.
- **Acteur** : Le responsable des opérations.
- **Précondition** : Le responsable des opérations doit être connecté.
- **Postcondition** : Les exercices lui sont effectivement présentés.
- **Scénario nominal** : Le responsable des opérations se connecte sur l'application de mutuelle mobile, puis clique sur la sidebar, puis sur « Exercices ». L'exercice en cours lui est ainsi affiché, tout comme les exercices précédents.

#### Consulter session

- **Objectif** : il s'agit pour un participant de consulter les détails d'une session
- **Acteur** : responsable des opérations
- **Préconditions** : La session doit déjà être terminée et le participant doit être connecté
- **Scénario nominal** : Le responsable des opérations se rend la rubrique sessions. Il consulte la ou les session(s) qu'il souhaite. Il ferme la rubrique

#### Supprimer un membre

- **Objectif** : il s'agit pour le responsable des opérations de changer le statut du membre afin qu'il n'ait plus accès à la mutuelle
- **Acteur** : responsable des opérations
- **Préconditions** : Le membre concerné doit avoir été enregistré par un admin
- **Postconditions** : Le membre ne peut plus accéder à l'application, plus aucune action concernant le membre ne peut être exécutée par le responsable des opérations, les données concernant le membre sont maintenues, seul l'état du membre change.
- **Scénario nominal** : Le responsable des opérations se rend dans la rubrique membre. Il choisit le membre concerné et clique sur le bouton pour supprimer le membre. Si le membre n'a pas d'activités en cours il est supprimé. Sinon la suppression est refusée

#### Supprimer un responsable des opérations

- **Objectif** : Supprimer un responsable des opérations.
- **Acteur** : Exclusivement l'principal.
- **Précondition** : Le responsable des opérations doit être connecté.
- **Postcondition** : Le responsable des opérations sélectionné est effectivement supprimé.
- **Scénario nominal** : L'administrateur principale se connecte avec ses privilèges, ouvre l'interface du responsable puis clique sur « supprimer ».

#### Ajouter un membre

- **Objectif** : Ajouter un membre.
- **Acteur** : Uniquement le responsable des opérations.
- **Précondition** : Le responsable des opérations doit être connecté.
- **Postcondition** : Le ou les membres concerné(s) est/sont effectivement ajouté(s).
- **Scénario nominal** : Le responsable des opérations se connecte à son compte, clique sur la sidebar, clique sur « membre », puis sur l'icône (+) et enfin remplit les champs. Ceci fait, il valide en cliquant sur « add member ».

#### Ajouter un responsable des opérations

- **Objectif** : Ajouter un responsable des opérations.
- **Acteur** : Uniquement l'administrateur.
- **Précondition** : aucune
- **Postcondition** : Le responsable des opérations est effectivement ajouté.
- **Scénario nominal** : L'administrateur se connecte à son compte, clique sur la sidebar, clique sur « responsable des opérations », puis sur l'icône (+) et enfin remplit les champs. Ceci fait, il valide en cliquant sur « add responsable ».

#### 4. Diagramme de classe

Il décrit clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets. Dans notre contexte de mutuelle mobile, notre diagramme de classe est le suivant :

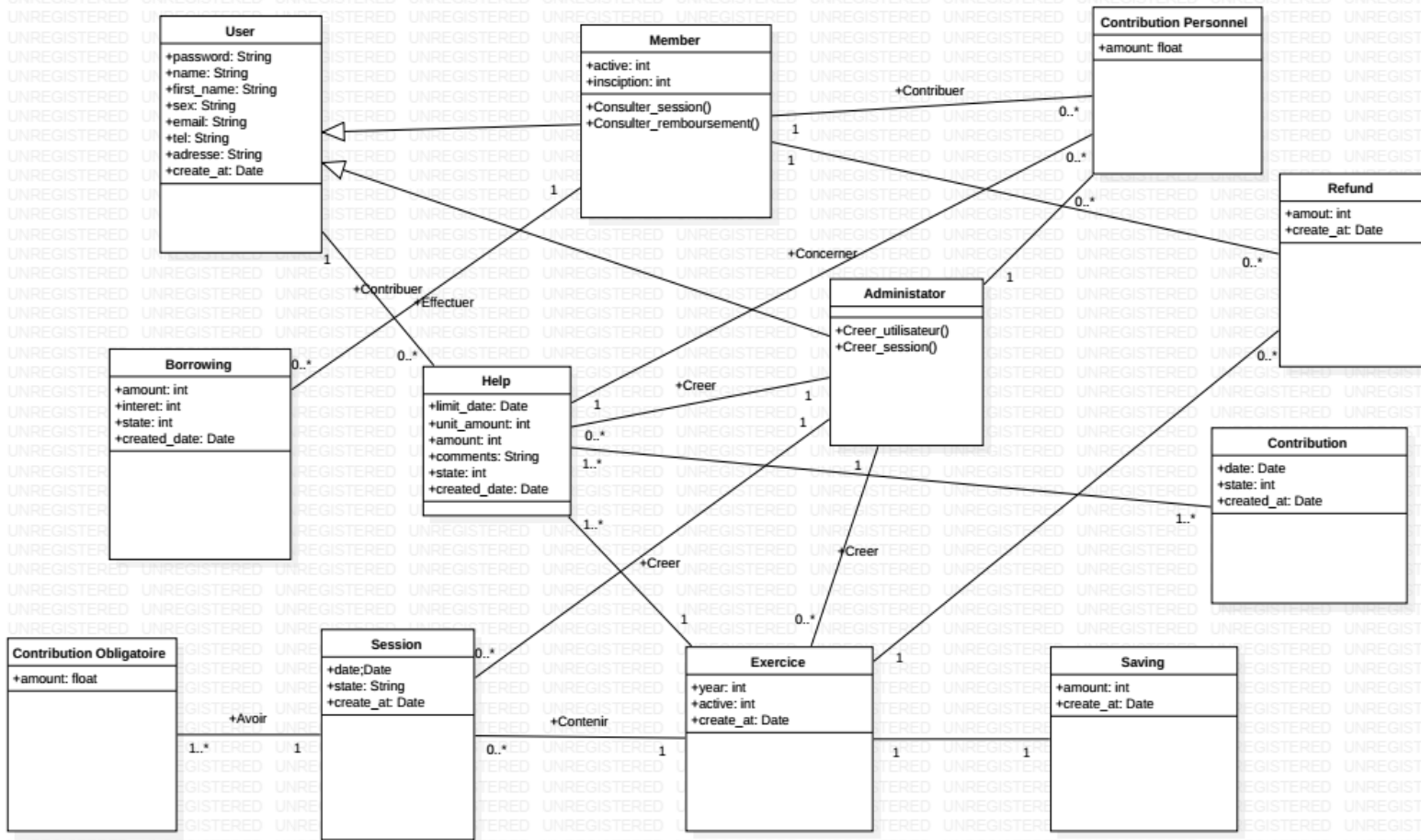


Figure 5: diagramme de classe

## 5. Diagramme d'activité

### a. Session

Nous présentons les étapes qui entrent dans le déroulement d'une session et les différents éléments intervenant :

- L'responsable des opérations crée la session et contrôle toutes les activités dans cette dernière.
- La session comprend les phases suivantes : phase des épargnes, phase des remboursements, phase des emprunts

- Il est également possible d'effectuer des demandes d'aide, des contributions et tout autre activité pendant la session.

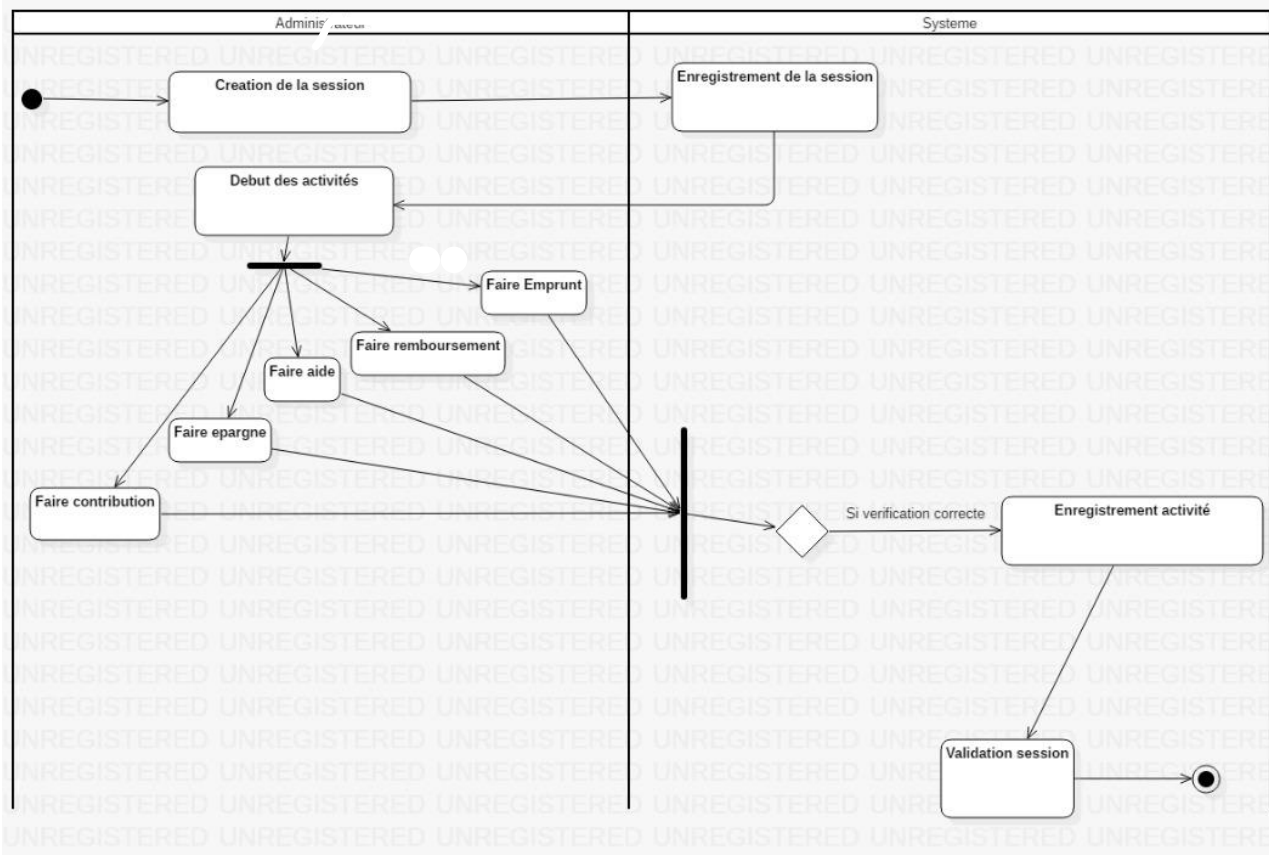


Figure 6: diagramme activité session

#### b. Authentification utilisateur

Nous présentons les étapes qui entrent dans le processus d'authentification d'un utilisateur et les différents éléments intervenant :

- L'authentification d'un utilisateur se fait par l'authentification de son email et son mot de passe.
- L'utilisateur entre ses identifiants
- L'API vérifie qu'ils sont corrects

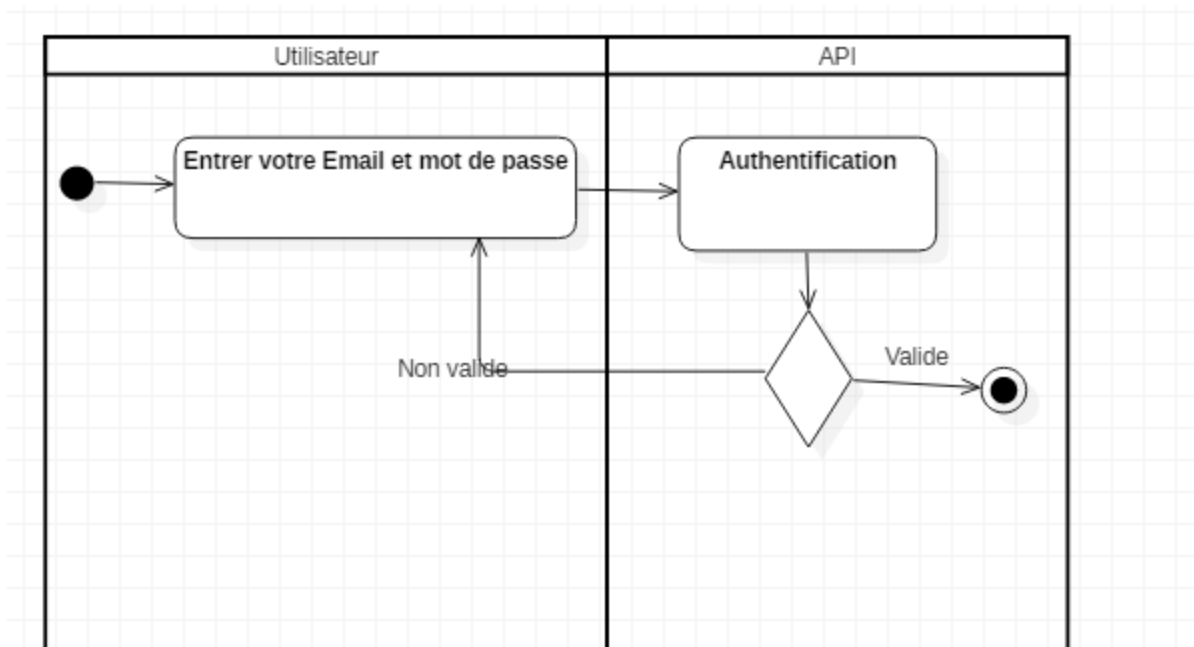


Figure 7: diagramme activité authentification de l'utilisateur

### c. Diagramme activité création membre

Nous présentons les étapes qui entrent dans le processus de création d'un membre et les différents éléments intervenant :

- Le membre est créé par un responsable des opérations
- L'responsable des opérations entre le nom, prénom, Téléphone, email, adresse et mot de passe du membre et valide.

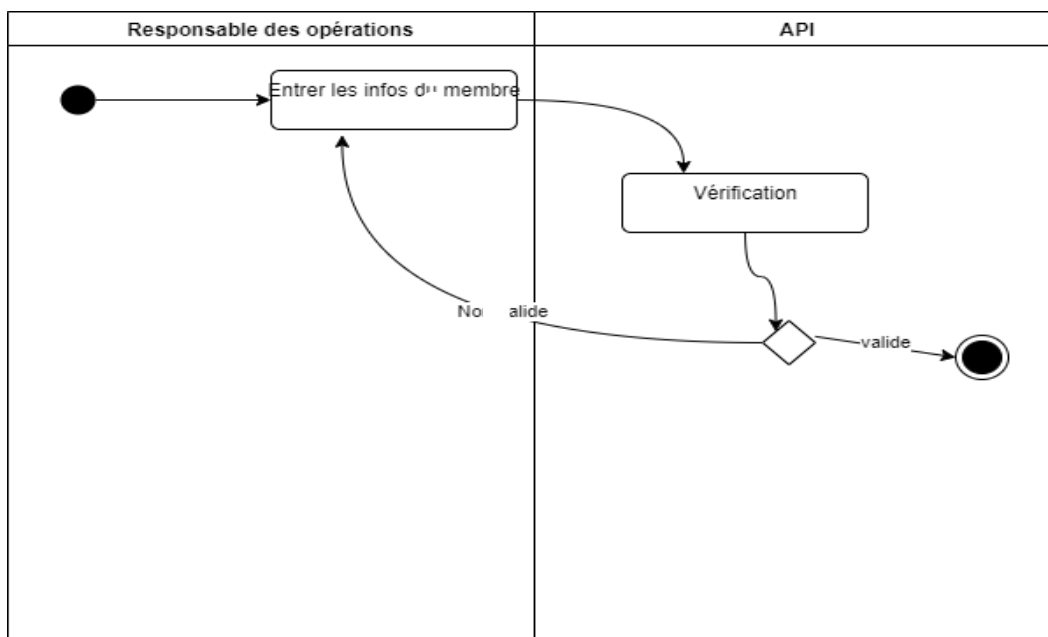


Figure 8: diagramme activité création du membre



## 6. Diagramme d'état transition

Le diagramme d'état transition est généralement utilisé pour décrire le comportement d'un objet en fonction de son état. Un objet réagit différemment à un même événement selon l'état dans lequel il se trouve.

### a. Session

La session peut être résumée en trois états tels que présentés dans le diagramme d'état transition ci-dessous :

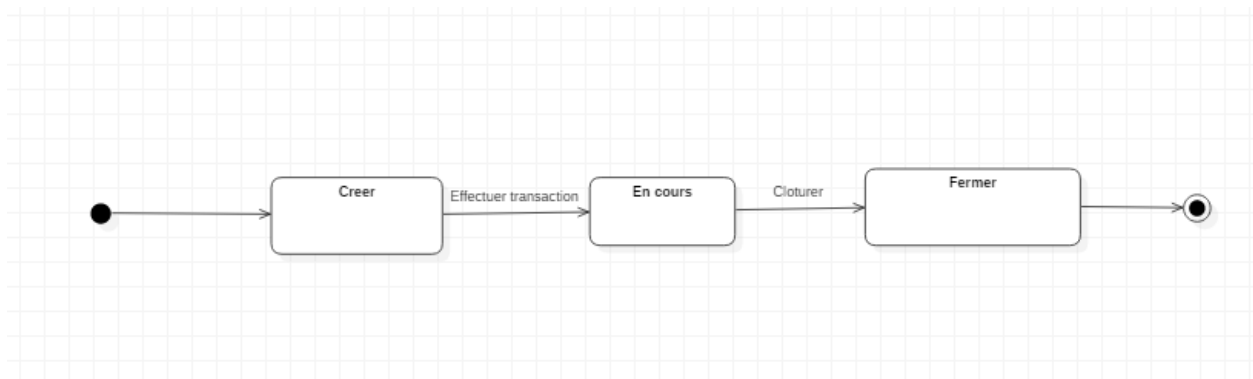


Figure 9: diagramme état transition session

### b. Exercice

L'exercice peut être résumée en trois états tels que présentés dans le diagramme d'état transition ci-dessous :

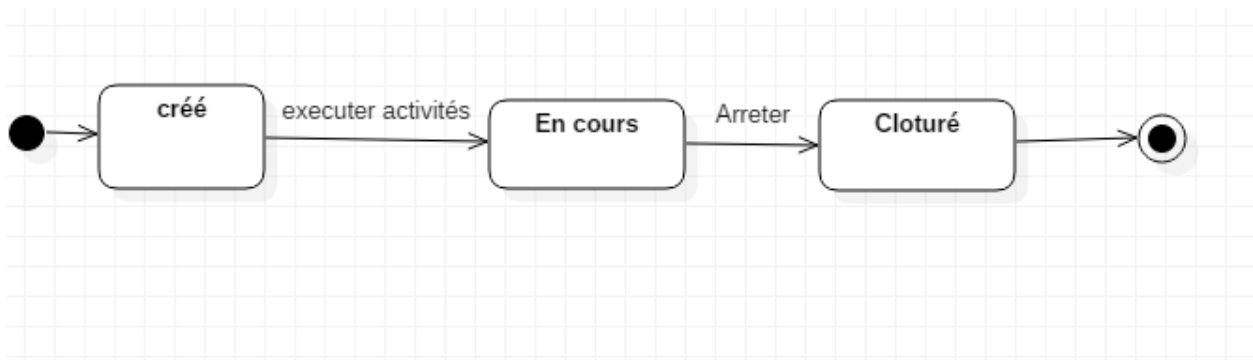


Figure 10: diagramme état transition exercice

### c. Membre

Les états du membre peuvent être résumés en trois états tels que présentés dans le diagramme d'état transition ci-dessous :

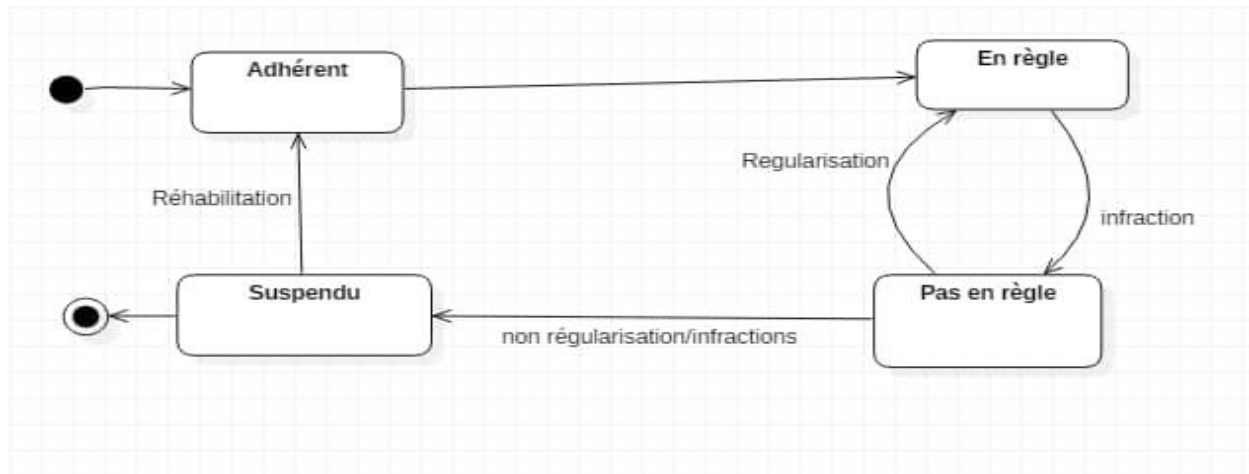


Figure 11: diagramme état transition de membre

#### d. Emprunt

L'emprunt peut être résumé en trois états tels que présentés dans le diagramme d'état transition ci-dessous :

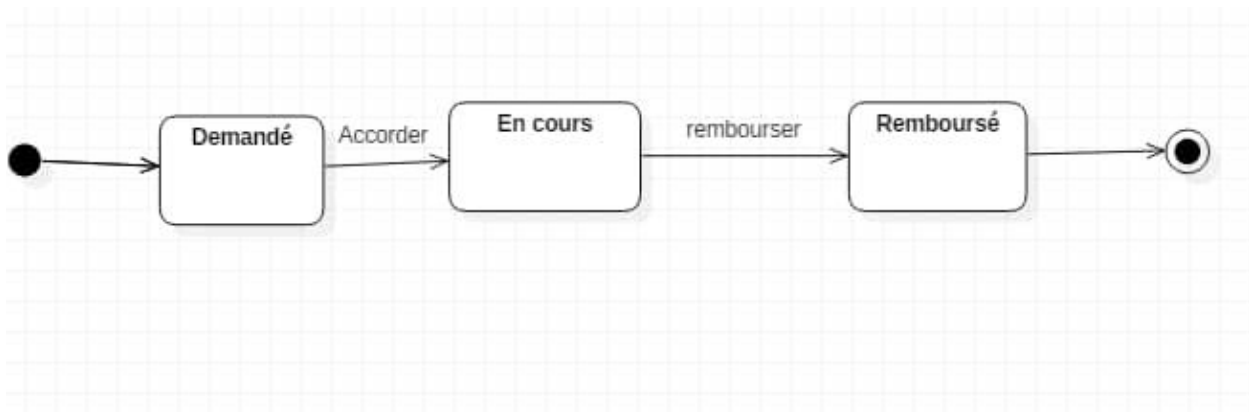


Figure 12: diagramme état transition emprunt

## BILAN

Dans ce chapitre, nous avons présenté la façon dont nous avons fait l'analyse de notre application.

## CHAPITRE 3 : CONCEPTION

Dans ce chapitre, nous entrerons un peu plus en dans la conception à proprement parlé de notre application Enspy mutuelle. Nous présenterons tout d'abord l'architecture de la solution (architecture de la solution, l'architecture logicielle et l'architecture de sécurité) , puis quelques diagrammes de conception (diagramme de classe technique, diagramme de séquence technique).

### I. ARCHITECTURE DE LA SOLUTION

#### 1. Architecture de la solution

Le diagramme de déploiement permet de représenter l'architecture physique supportant l'exploitation du système. Nous avons adopté pour notre solution une architecture physique constituée des éléments suivants :

- **Un Téléphone Portable** : Le téléphone mobile héberge l'application cliente de la mutuelle, une base de données locale devant stocker les différents Tokens d'authentification.
- **Le serveur de service** : il fournit une API à travers laquelle l'on peut effectuer le traitement, l'envoi, la suppression, la modification et la création données.

#### 2. Architecture logicielle

L'architecture d'un logiciel décrit la manière dont seront agencés les différents éléments d'une application et comment ils interagissent entre eux. Elle définit comment le système doit être conçu de manière logicielle pour répondre aux spécifications. Nous allons présenter l'architecture logicielle au niveau de l'application cliente et au niveau du serveur.

Application cliente elle a été faite selon l'architecture Flux. Flux évite MVC au profit d'un flux de données unidirectionnel. Lorsqu'un utilisateur interagit avec une vue, la vue propage une action via un répartiteur central, vers les différents magasins contenant les données et la logique métier de l'application, qui met à jour toutes les vues concernées. L'architecture logicielle au niveau de l'application cliente présente les couches suivantes :



Figure 13: architecture logicielle

- **Dispatcher** : Le répartiteur est le hub central qui gère la distribution des données dans l'application, il détermine quel composant aura accès aux données.

- **Store** : Leur rôle est quelque peu similaire à un modèle dans un MVC traditionnel, Il permet à travers un ensemble de méthodes d'effectuer des opérations simples d'ajout, suppression, mise à jour et sélection d'éléments dans la base de données.
- **View** : Elle est en contact direct avec l'utilisateur. Elle est constituée de composants collaborant ensemble pour former des interfaces utilisateurs dynamiques.

### 3. Architecture de sécurité

L'architecture de sécurité permet d'assurer la protection des données manipulées par le système. Elle permet de répondre efficacement aux incidents de sécurité qui pourraient se produire. La sécurité est mise en œuvre au niveau des deux modes d'échanges de données entre les différents nœuds de l'architecture physique et aussi au niveau de l'accès à l'application.

#### a. Au niveau de l'application cliente

La sécurité à ce niveau est assurée par l'authentification de l'utilisateur qui souhaite utiliser l'application. L'authentification est un processus visant à assurer la légitimité de la demande d'accès faite par une entité. L'authentification consiste généralement à vérifier que l'utilisateur possède une preuve de son identité ou de son statut sous une des formes suivantes :

- Ce qu'il sait (mot de passe, numéro d'identification personnel, etc...)
- Ce qu'il possède (acte de naissance, carte d'identité, carte à puce, Token OTP, diplôme, passeport etc...)
- Ce qu'il est (photo, caractéristique physique, biométrie)

Dans le cas de la mutuelle nous optons pour une authentification simple reposant sur l'email et le mot de passe comme identifiants de l'utilisateur. Une fois fournie il sera envoyé à l'application clientèle un Token, pour son accès à l'application. **La suppression de ce Token entraîne la déconnexion de l'utilisateur.**

#### b. Au niveau des échanges de données entre l'application de Mutuelle et le serveur

L'échange de messages entre l'application cliente et le serveur est sécurisé et assurée par le framework Knox. Les jetons Knox ne sont stockés que sous une forme cryptée. Même si la base de données était volée d'une manière ou d'une autre, un attaquant ne pourrait pas se connecter avec les informations d'identification volées.

## II. QUELQUES DIAGRAMMES DE CONCEPTION

### 1. Diagramme de classe technique

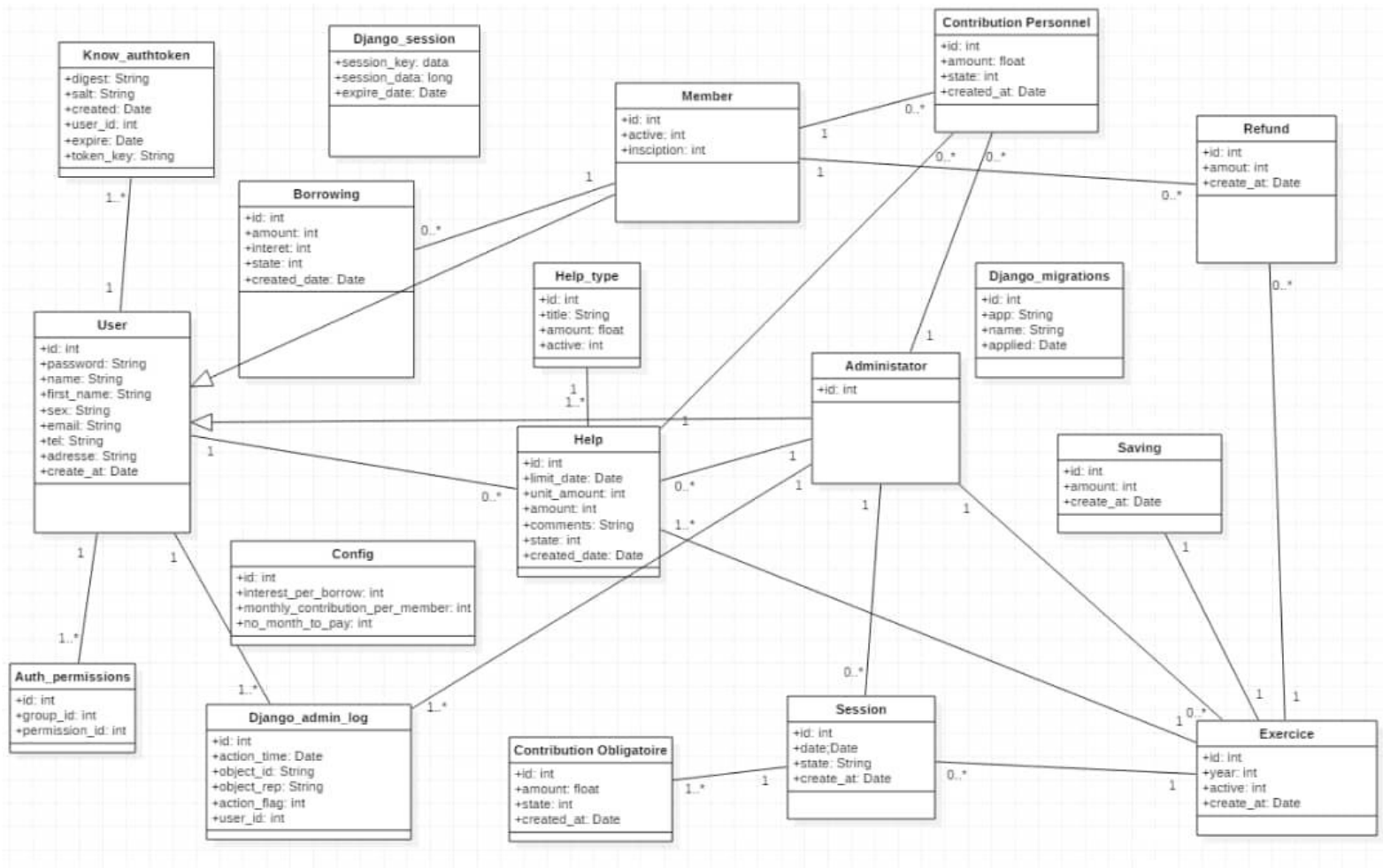


Figure 14: diagramme de séquence technique

## 2. Diagramme de séquence technique

### a. Côté responsable des opérations

#### i. S'authentifier

La description de la procédure d'authentification est décrite par le diagramme suivant :

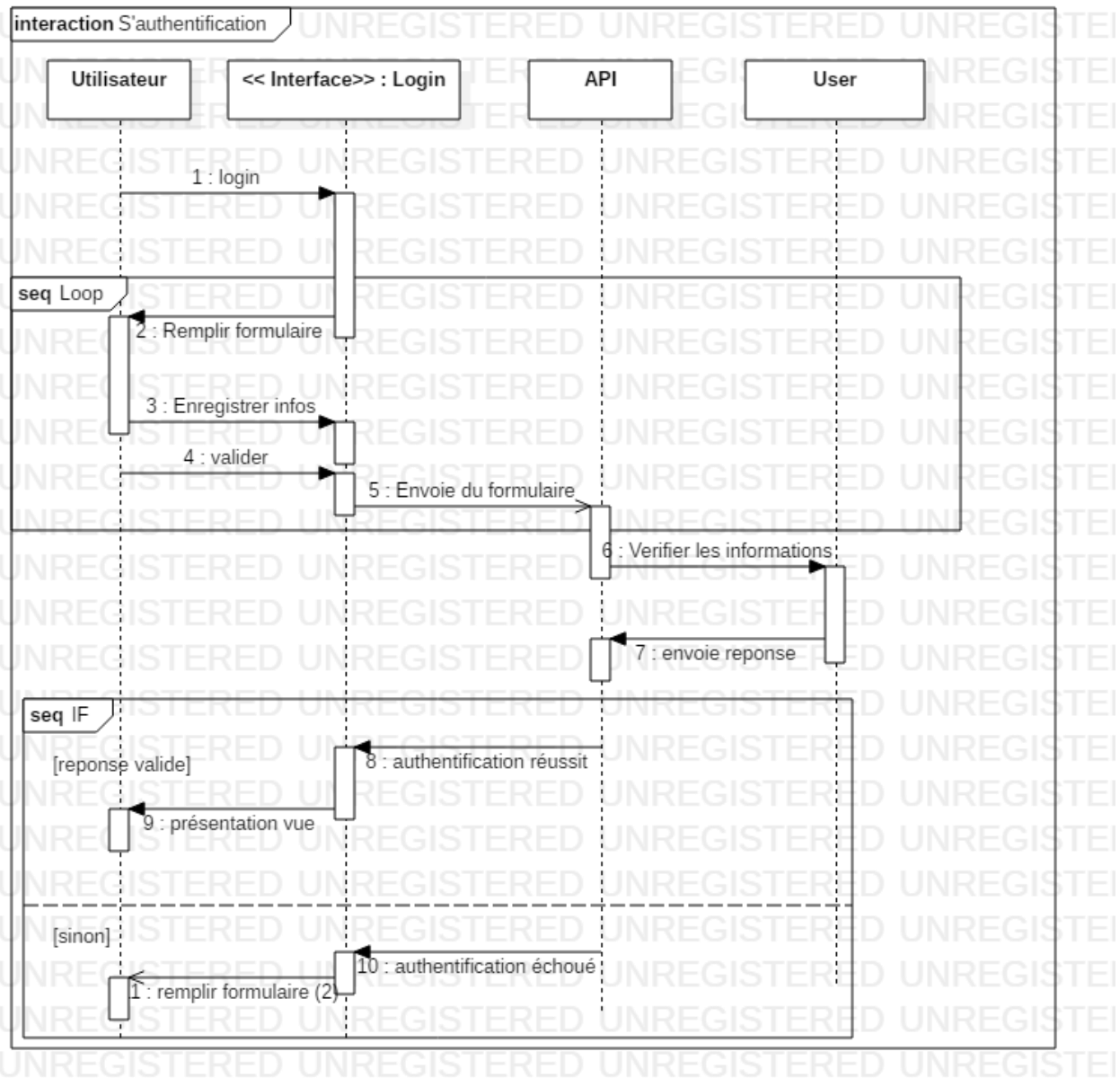


Figure 15: diagramme de séquence technique s'authentifier

ii. [Ajouter un responsable des opérations](#)

La description de la procédure d'ajout d'un responsable des opérations est décrite par le diagramme suivant :

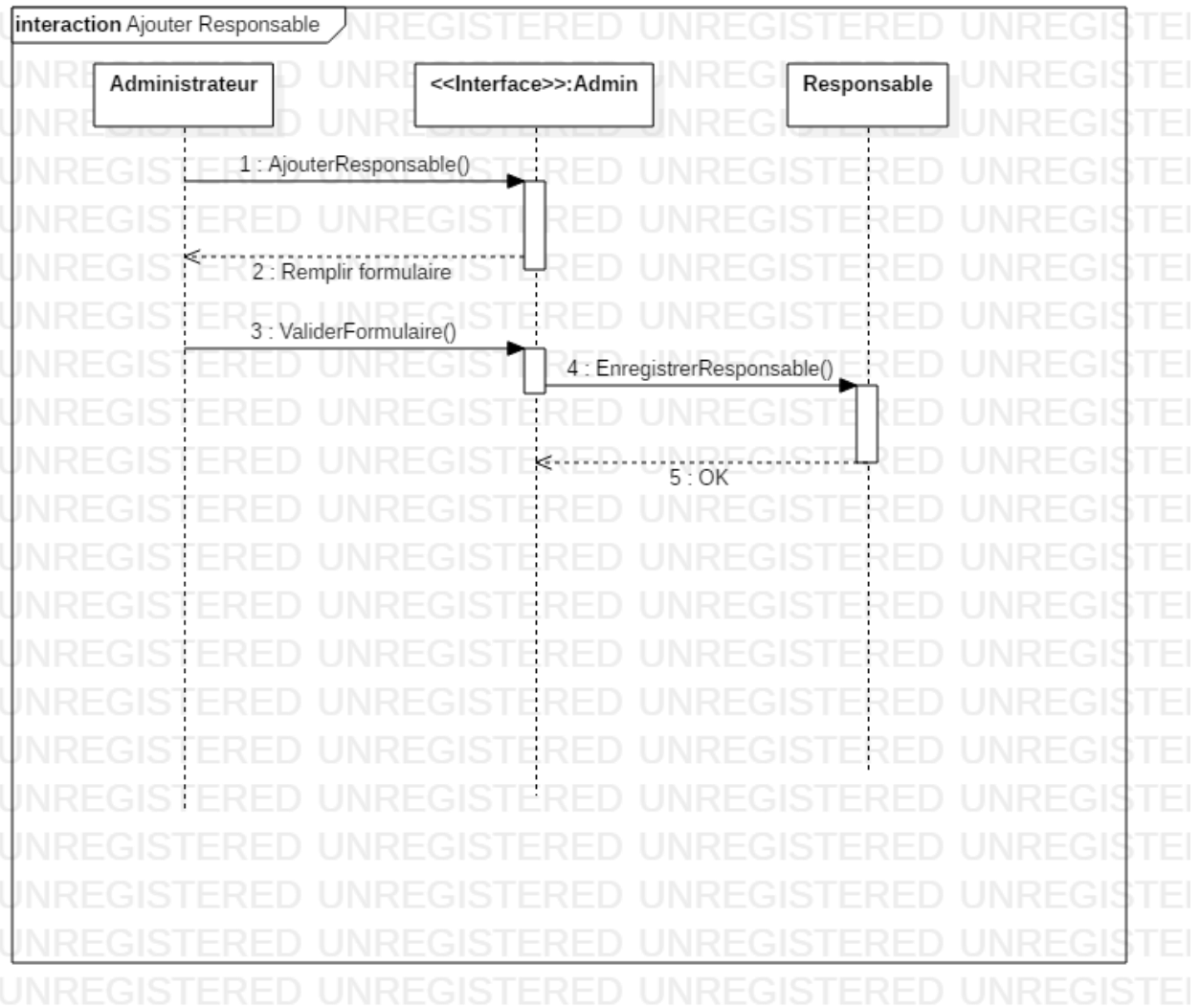


Figure 16: diagramme de séquence technique ajouter responsable des opérations

iii. [Ajouter un membre](#)

La description de la procédure d'ajout d'un membre est résumée par le diagramme suivant :

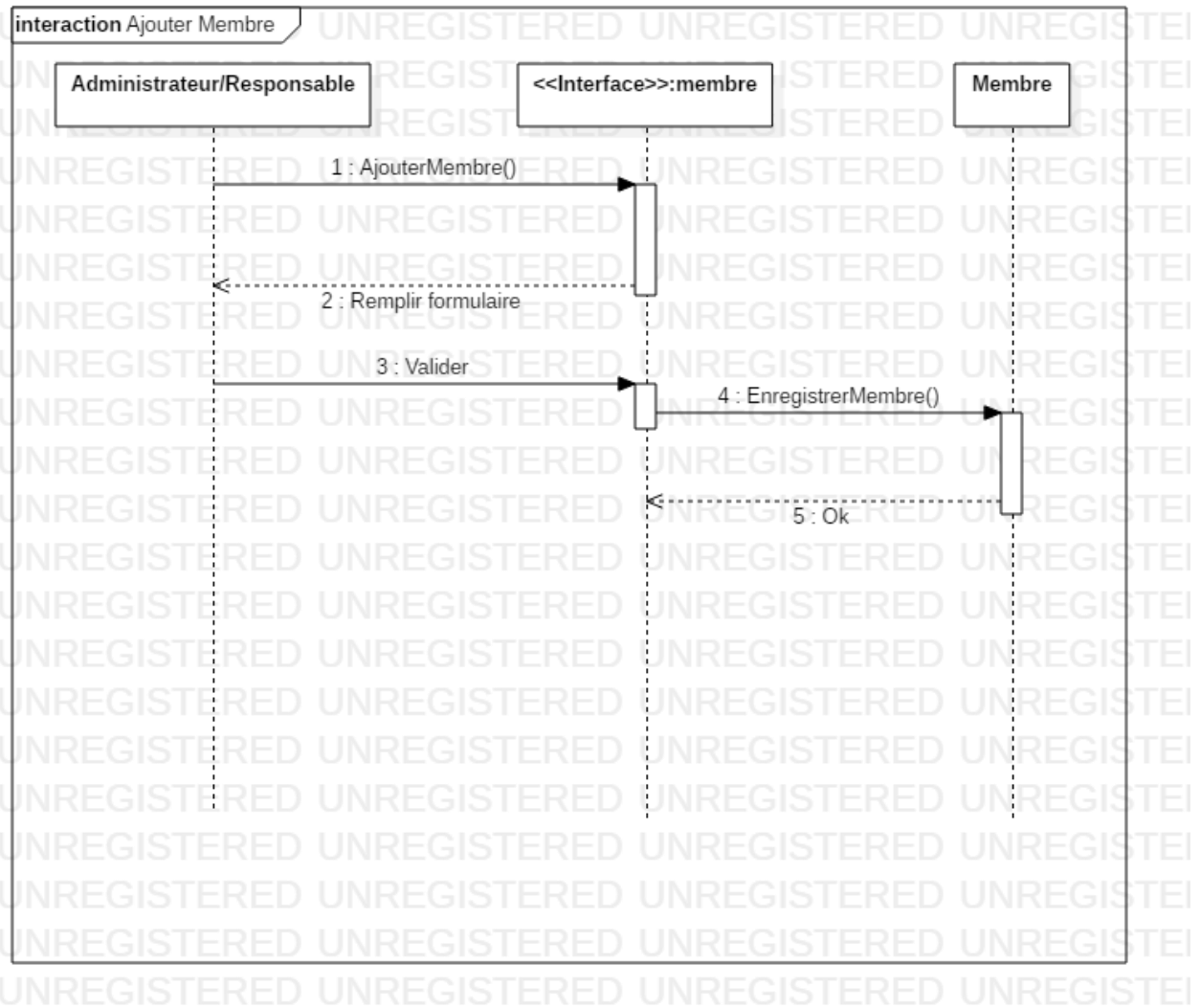


Figure 17: diagramme de séquence technique ajouter membre

iv. [Ajouter un type d'aide](#)

La description de la procédure d'ajout d'un type d'aide est résumée par le diagramme suivant :



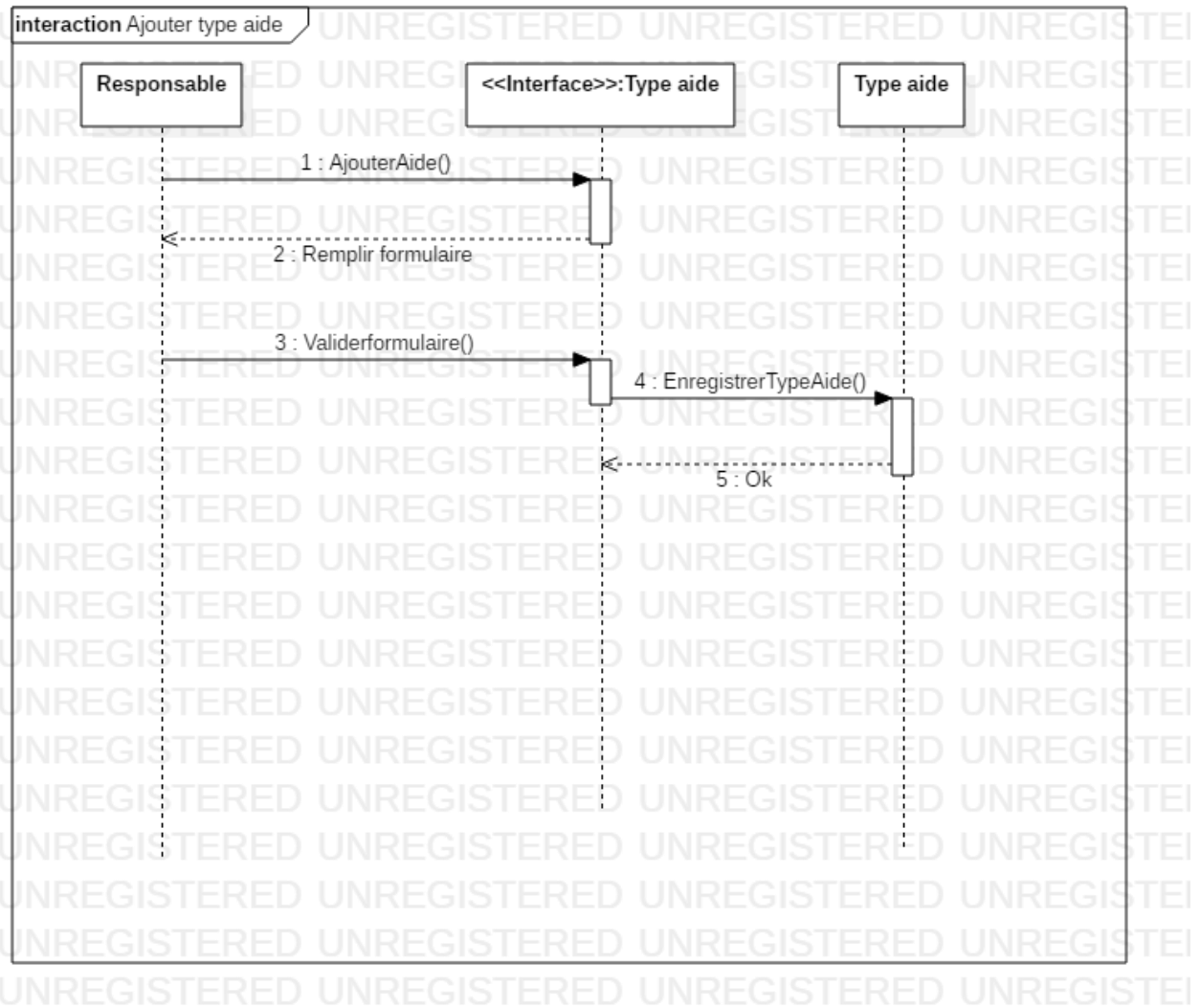


Figure 18: diagramme de séquence technique ajouter type d'aide

v. [Modification de la configuration](#)

La description de la procédure de modification de la configuration est résumée par le diagramme suivant :

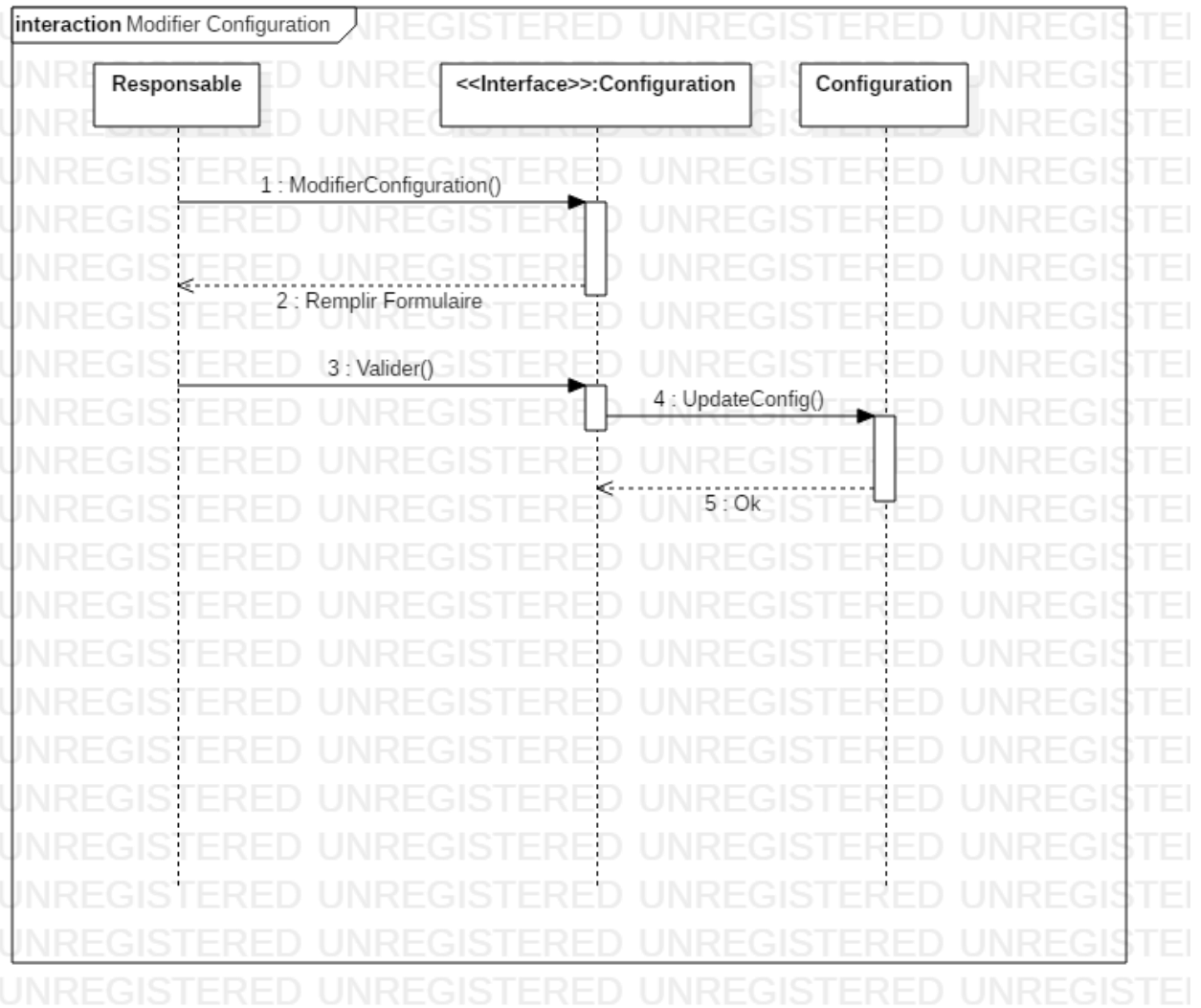


Figure 19: diagramme de séquence technique modifier la configuration

vi. [Modification du profil](#)

La procédure de modification du profil est la suivante :

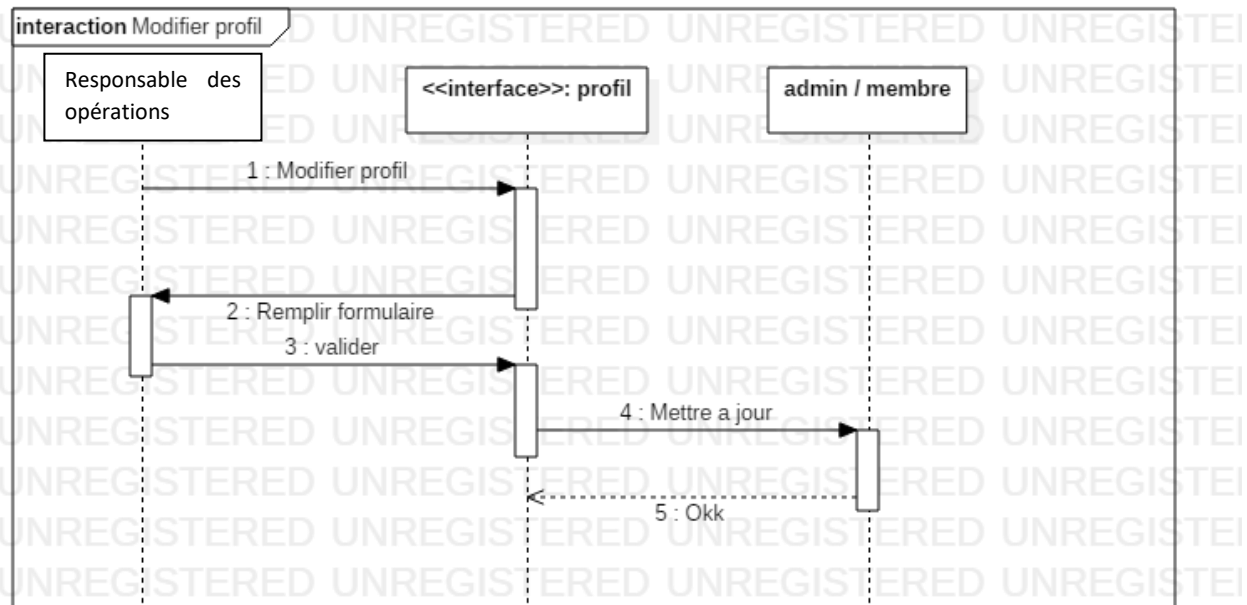


Figure 20: : diagramme de séquence technique modifier profil

vii. [Suppression d'un responsable des opérations](#)

La procédure de suppression d'un responsable des opérations est la suivante :

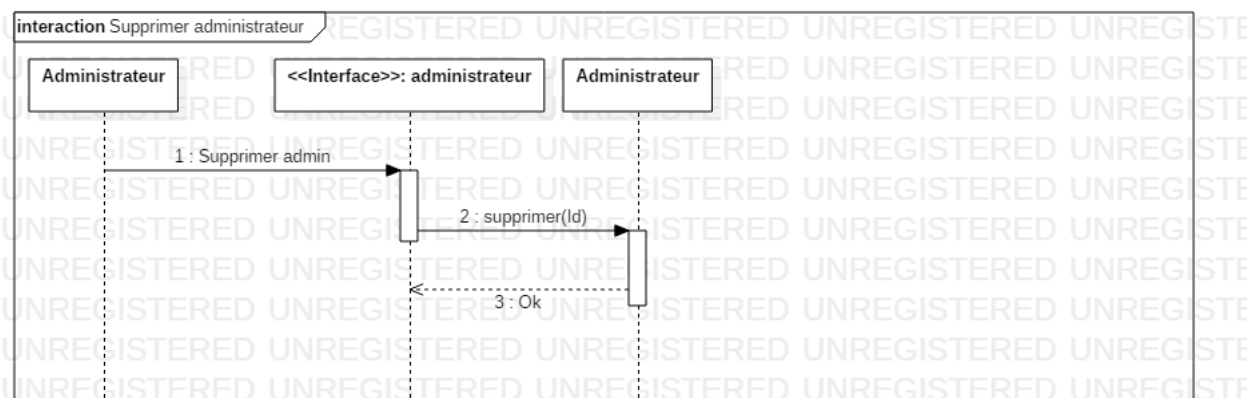


Figure 21: diagramme de séquence technique suppression d'un responsable des opérations

viii. [Suppression d'un membre](#)

La procédure de suppression d'un membre est la suivante :

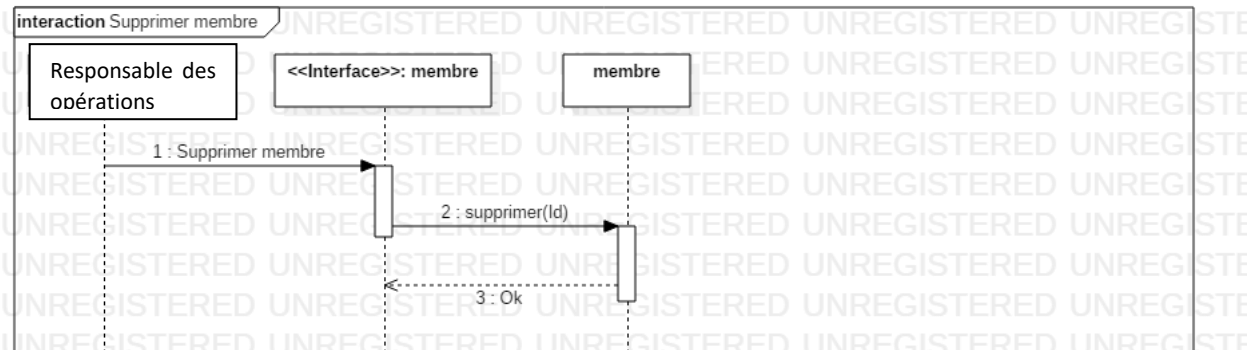


Figure 22: diagramme de séquence technique suppression de membre

#### ix. [Validation d'une aide](#)

La procédure de validation d'une aide est la suivante :

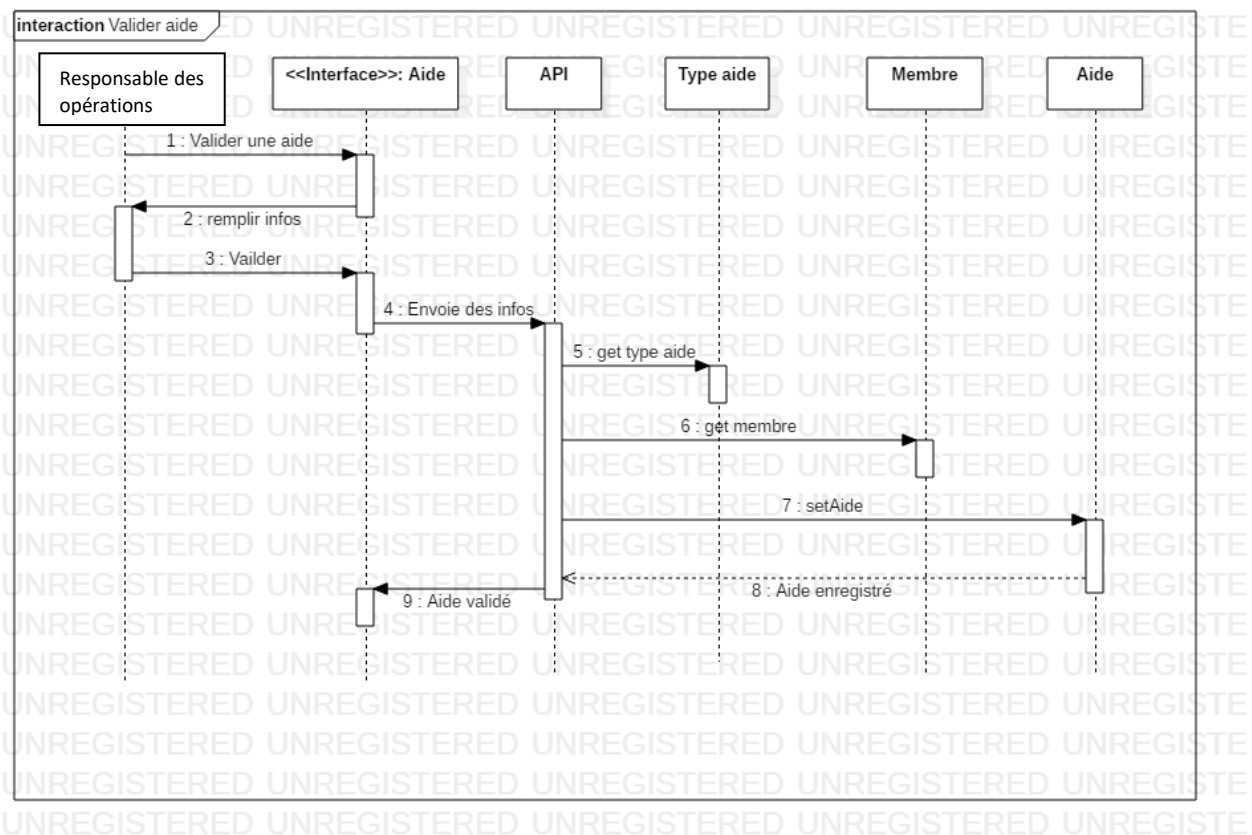


Figure 23: diagramme de séquence technique valider une aide

#### x. [Validation d'un emprunt](#)

La procédure de validation d'un emprunt est la suivante :

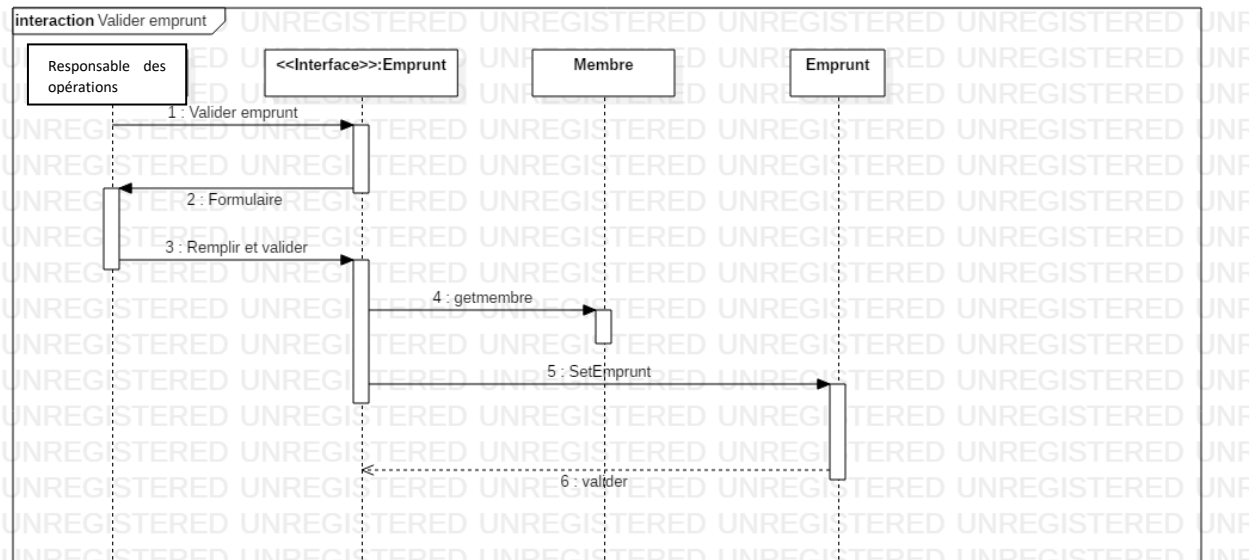


Figure 24: diagramme de séquence technique valider emprunt

#### xi. [Validation d'un remboursement](#)

La procédure de validation d'un remboursement est la suivante :

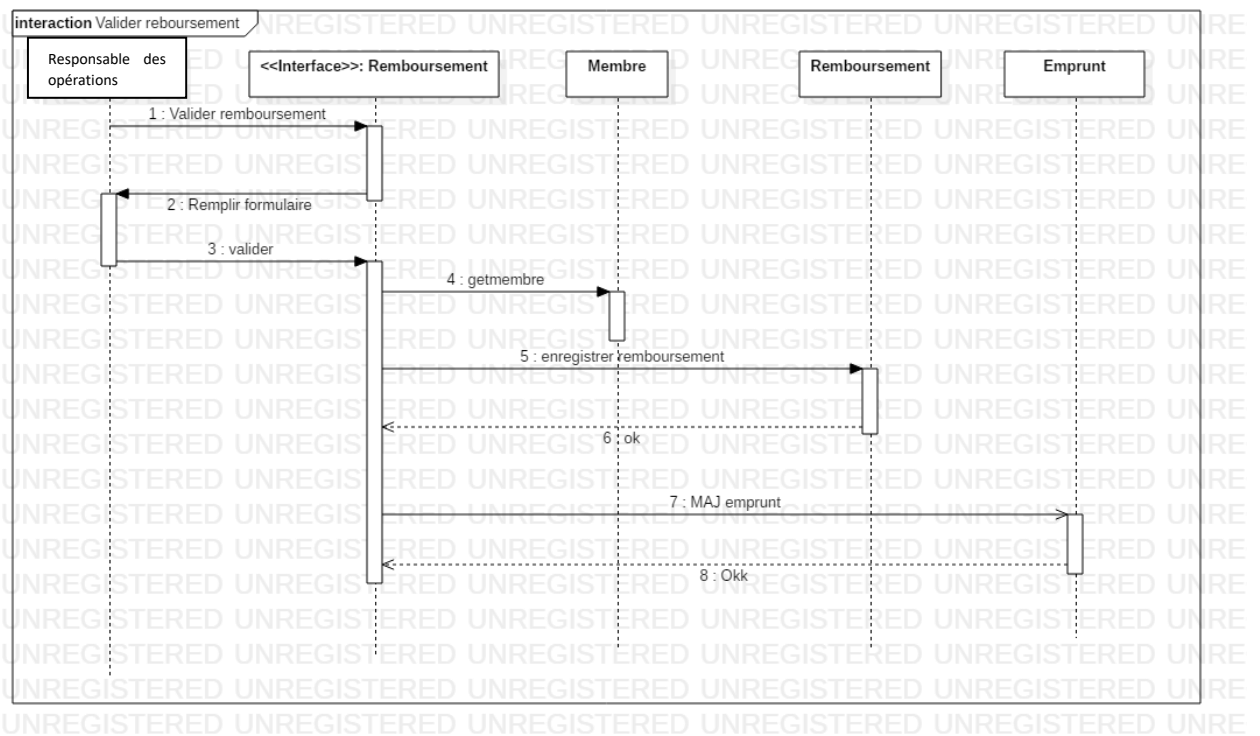


Figure 25: diagramme de séquence technique valider remboursement

#### xii. [Validation d'une contribution](#)

La procédure de validation d'une contribution est la suivante :

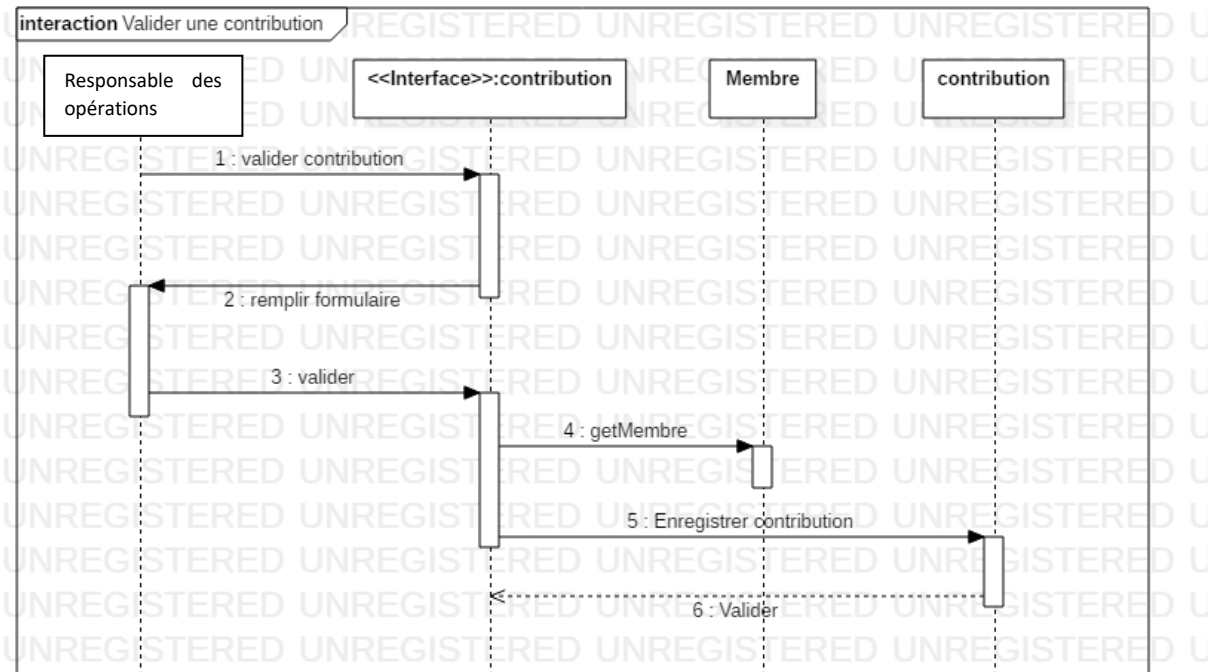


Figure 26: diagramme de séquence technique valider contribution

b. Coté membre

i. S'authentifier

La procédure d'authentification est la suivante :

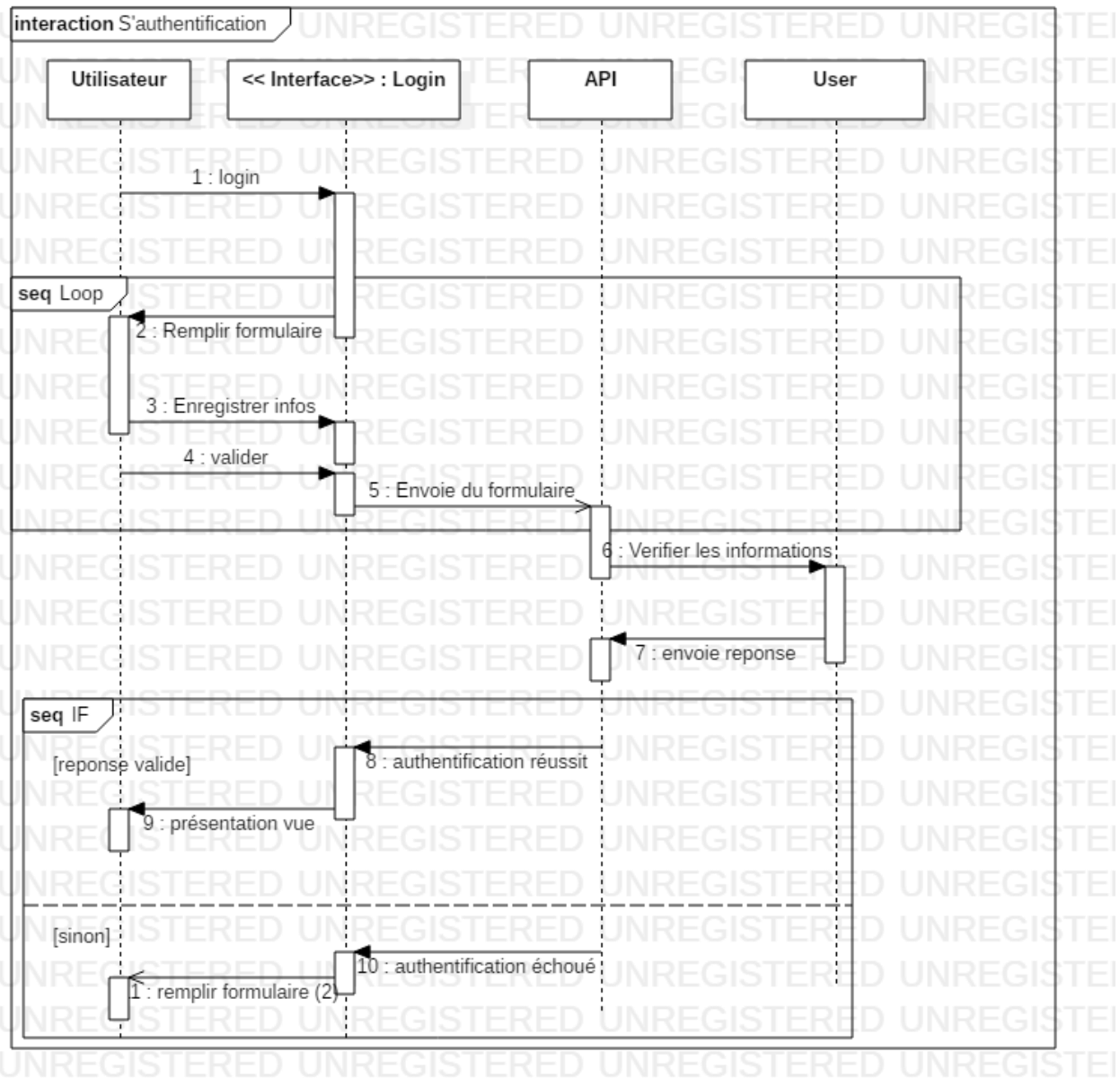


Figure 27: authentication

ii. [Demander une aide](#)

La procédure de demande d'aide est la suivante :

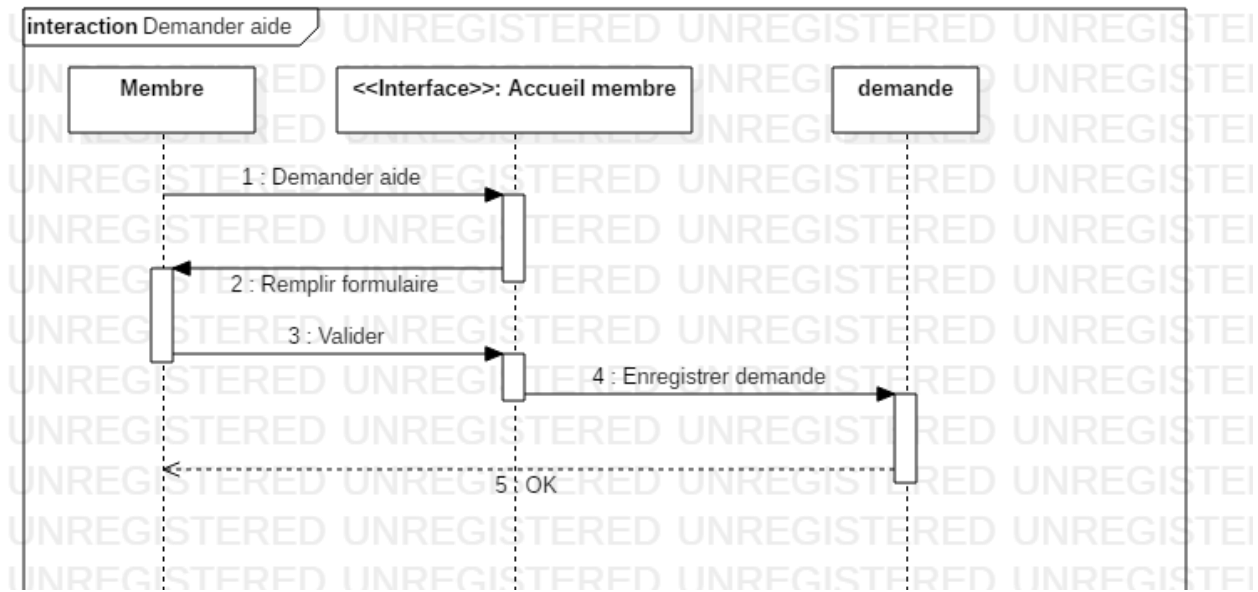


Figure 28 : diagramme de séquence technique demander une aide

iii. [Demander à effectuer une contribution](#)

La procédure de demande à effectuer une contribution est la suivante :

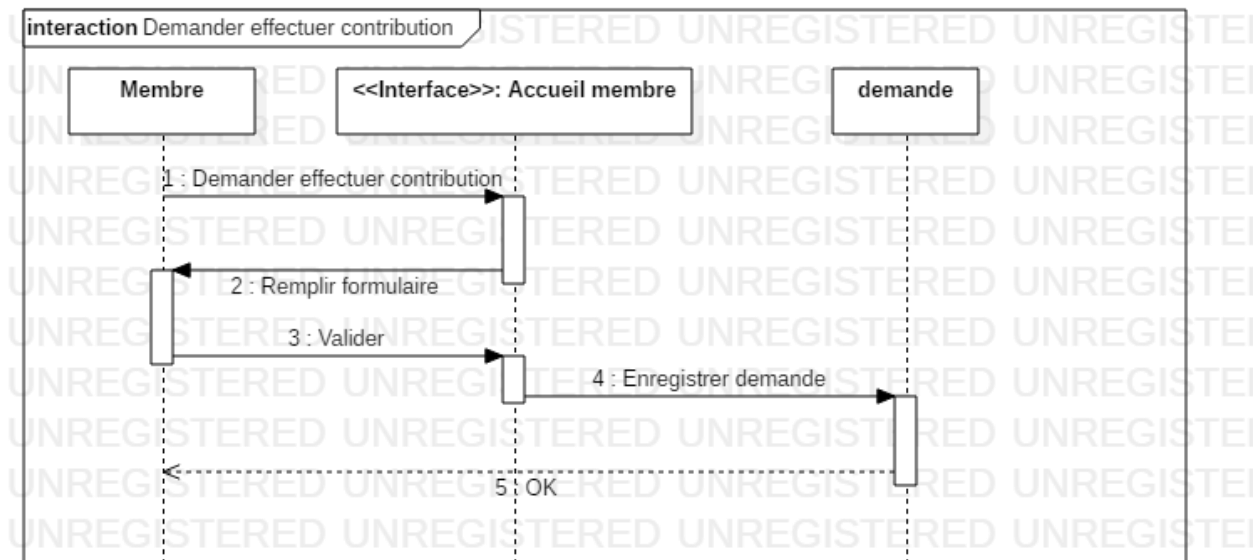


Figure 29 : diagramme de séquence technique demander une contribution

iv. [Demander un emprunt](#)

La procédure de demande d'un emprunt est la suivante :



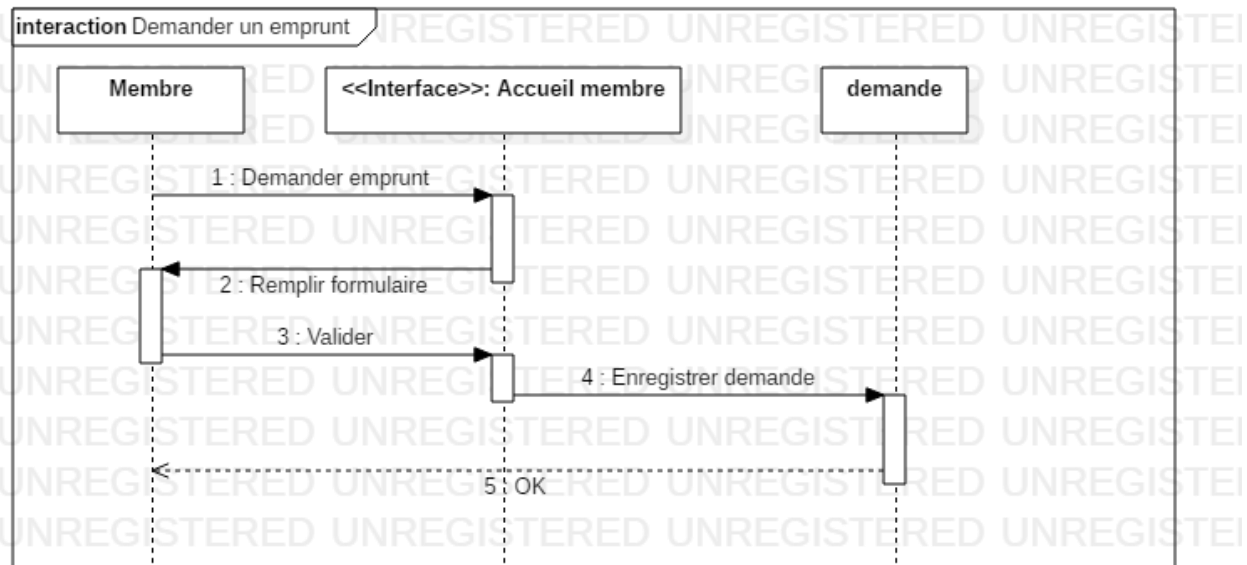


Figure 30 : diagramme de séquence technique demander un emprunt

v. [Modifier son profil](#)

La procédure de modification d'un profil est la suivante :

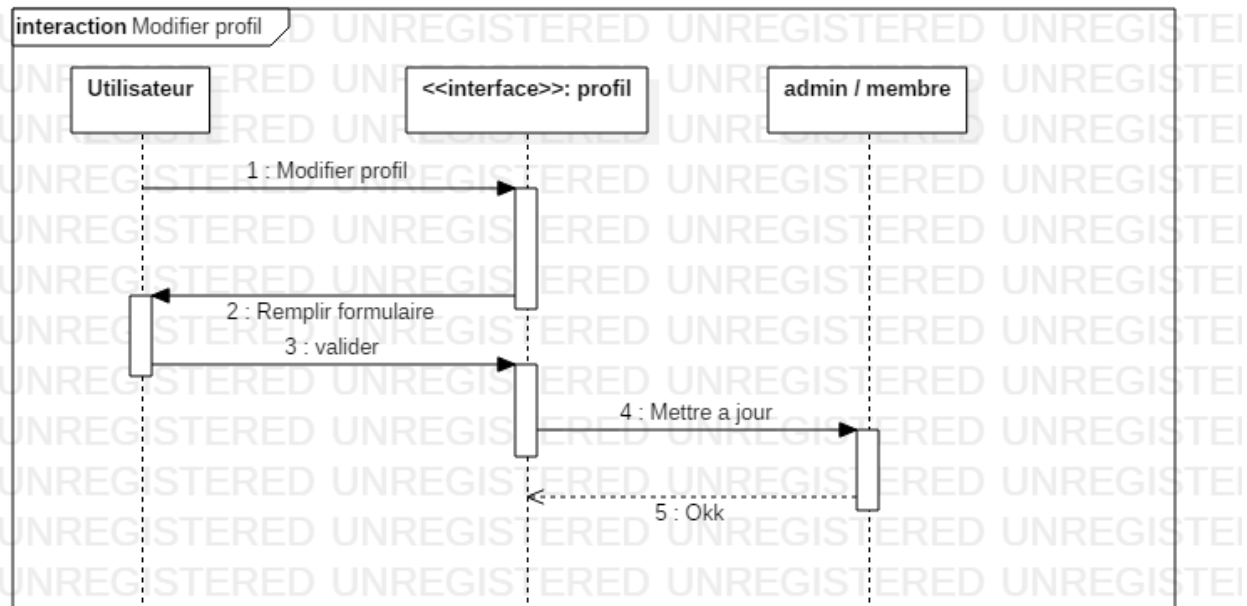


Figure 31 : diagramme de séquence technique modifier profil

## BILAN

Il était question pour nous de présenter la partie conception de notre application à travers une présentation de l'architecture de la solution, puis de quelques diagrammes de conception notamment le diagramme de classe technique et le diagramme de séquence technique.

## CHAPITRE 4 : IMPLÉMENTATION ET TEST

Dans ce chapitre, nous allons présenter les outils que nous avons utilisés pour notre implémentation ainsi que les résultats des tests que nous avons effectués.

### I. OUTILS UTILISÉS ET TECHNOLOGIES

Parmi les outils utilisés nous avons :

- **Le Framework React Native** : un cadre basé sur JavaScript utilisé pour créer des applications mobiles natives sur Android et la plate-forme iOS. Il est basé sur React, la bibliothèque JavaScript de Facebook, pour créer des interfaces utilisateur pour les plateformes mobiles. Les développeurs web peuvent utiliser React Native JavaScript pour écrire des applications Android et iOS qui agissent et apparaissent de manière similaire aux applications natives. Le code écrit avec React Native est également partageable entre les plateformes, ce qui permet un développement efficace et simultané sur iOS et Android. Les applications React Native sont créées en combinant JavaScript avec un balisage de type XML, c'est-à-dire JSX. La passerelle React Native est chargée d'effectuer le rendu natif des API en Java (pour Android) et Swift (pour iOS).
- **Django REST Framework** : est un outil puissant et flexible utilisé pour créer des API Web. Django REST Framework est une librairie permettant la mise en place d'une API pour Django (vous pouvez regarder sa documentation sur le site officiel). Basée sur le framework Django, elle propose la mise en place des endpoints d'une façon similaire à la mise en place des URL, Views et Forms de Django.
- **Visual Studio Code** : est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.
- **Android Studio** : est un environnement de développement d'applications mobiles. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle.
- **Postman** : Il s'agit d'un outil permettant de construire et tester rapidement des requêtes HTTP d'API généralement utilisé lors des tests d'intégration. Il est utilisé pour tester l'envoi et la réception de fichiers.

### II. RÉSULTATS OBTENUS

#### 1. Les interfaces de bienvenu, connexion et accueil

Lorsqu'on ouvre l'application, l'on tombe sur une page de bienvenue. Une interface simple qui souhaite bienvenu à l'utilisateur. Lorsqu'il clique sur login, il arrive sur l'interface de connexion dans laquelle il devra remplir les champs email et mot de passe. Ceci fait, il arrive instantanément sur la page d'accueil.



Figure 32: interface de bienvenue

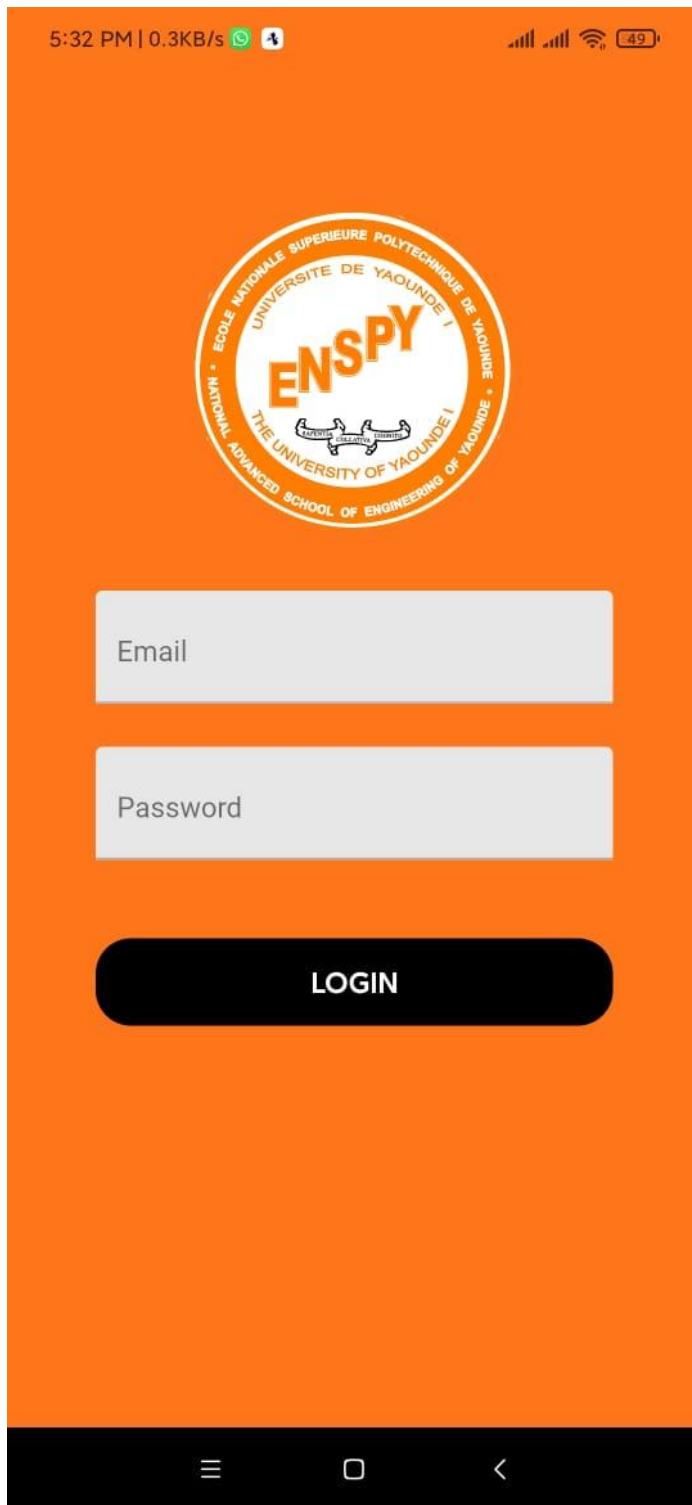


Figure 33: interface de login

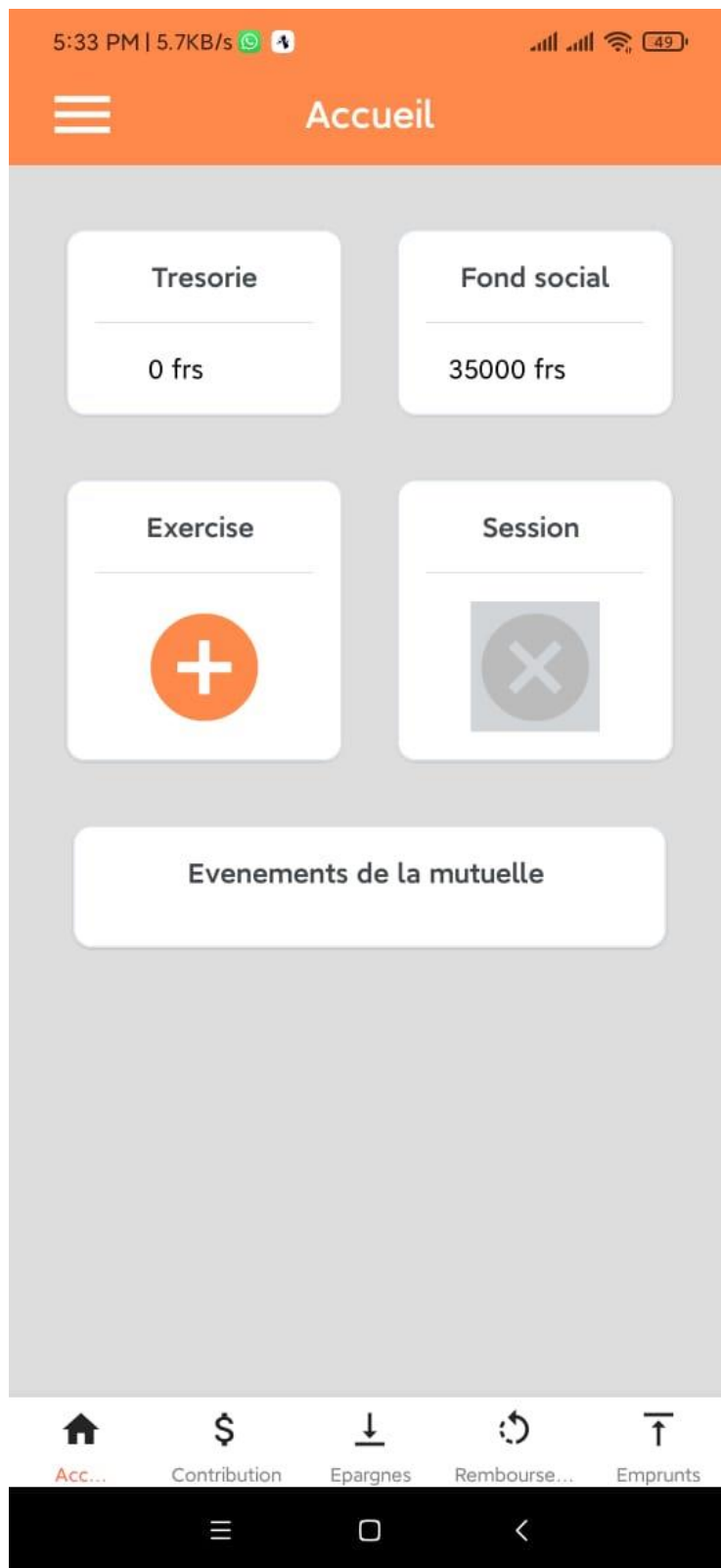


Figure 34: interface d'accueil

## 2. La sidebar

La sidebar regroupe l'ensemble des opérations possibles pour l'utilisateur des opérations.

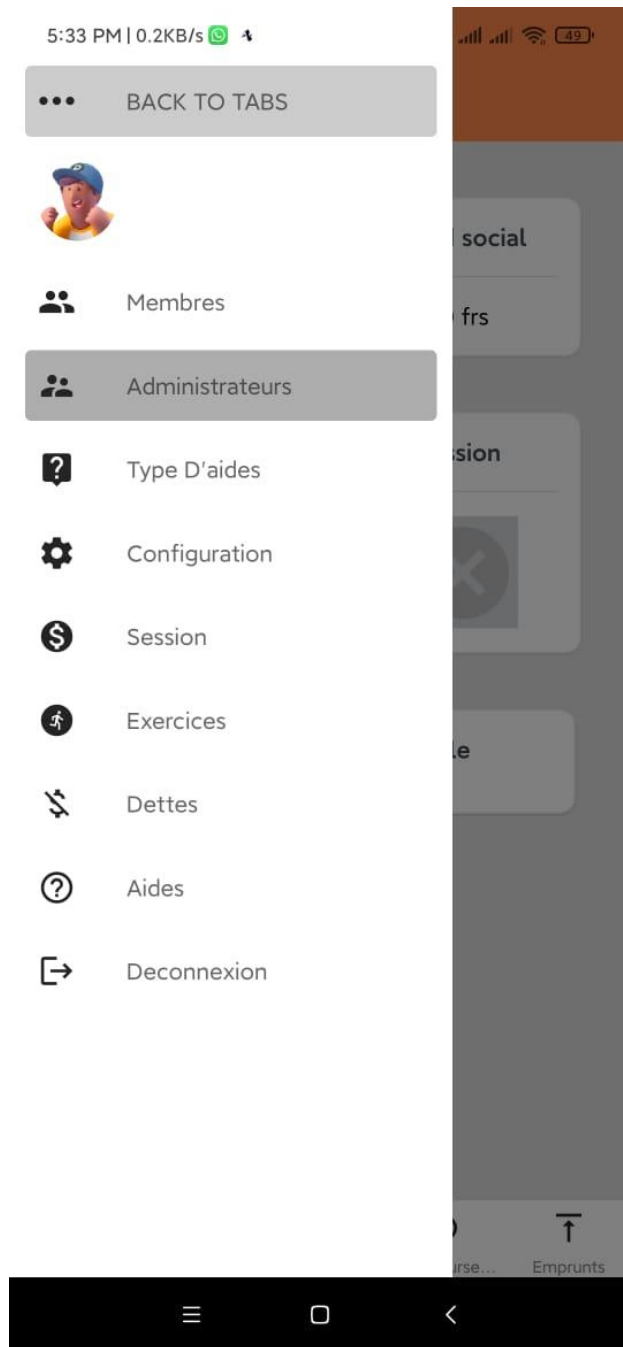


Figure 35: interface sidebar

3. Les membres, les responsable des opérations, les types d'aides, la configuration, la session, les exercices, les dettes, les aides et la deconnexion

Lorsque la sidebar est ouverte, elle présente les différentes opérations que les responsable des opérations peuvent effectuer :

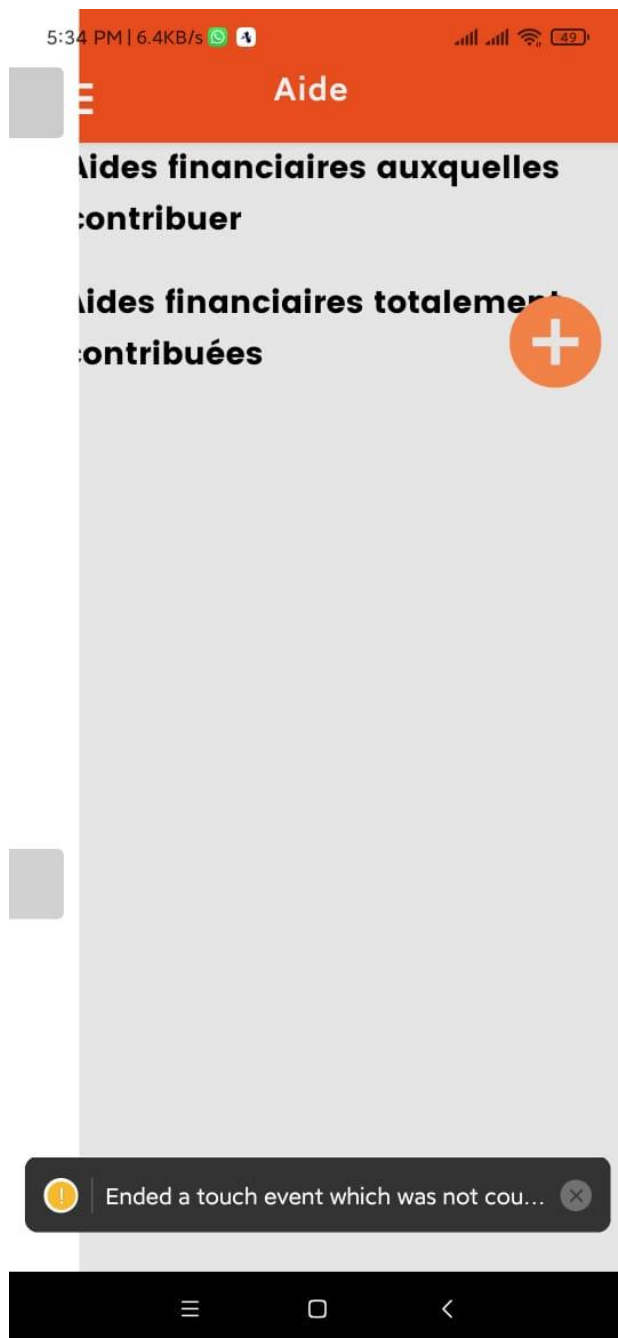
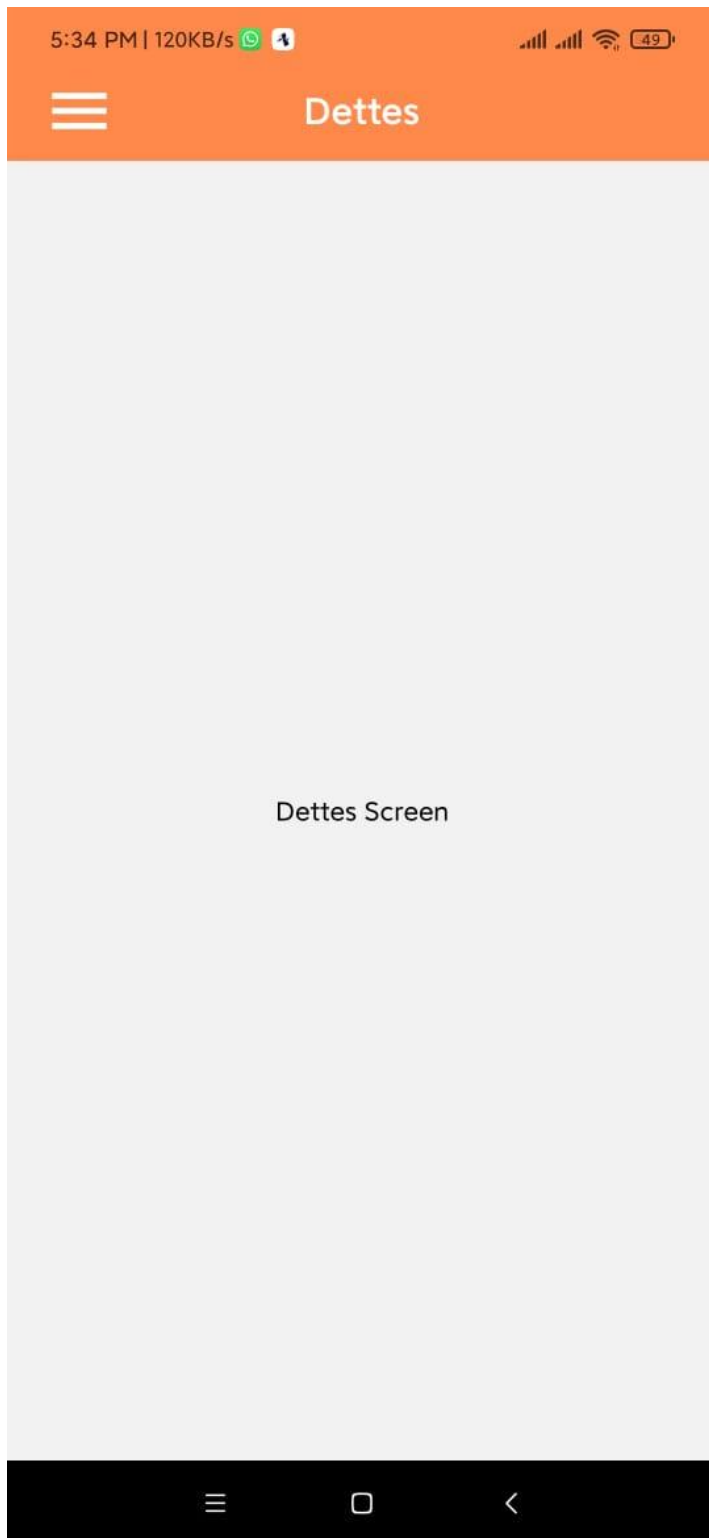


Figure 36: interface d'aide





*Figure 37: interface de dettes*

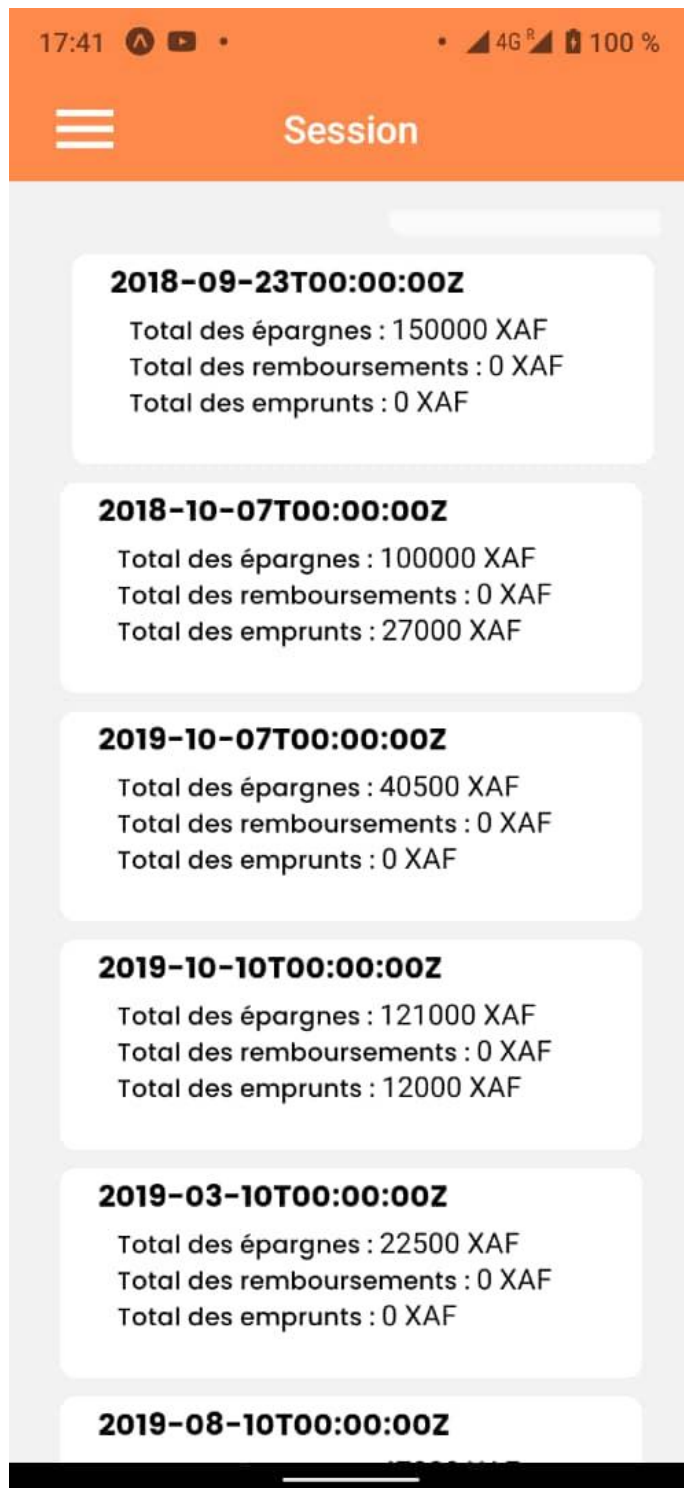


Figure 38: interface session

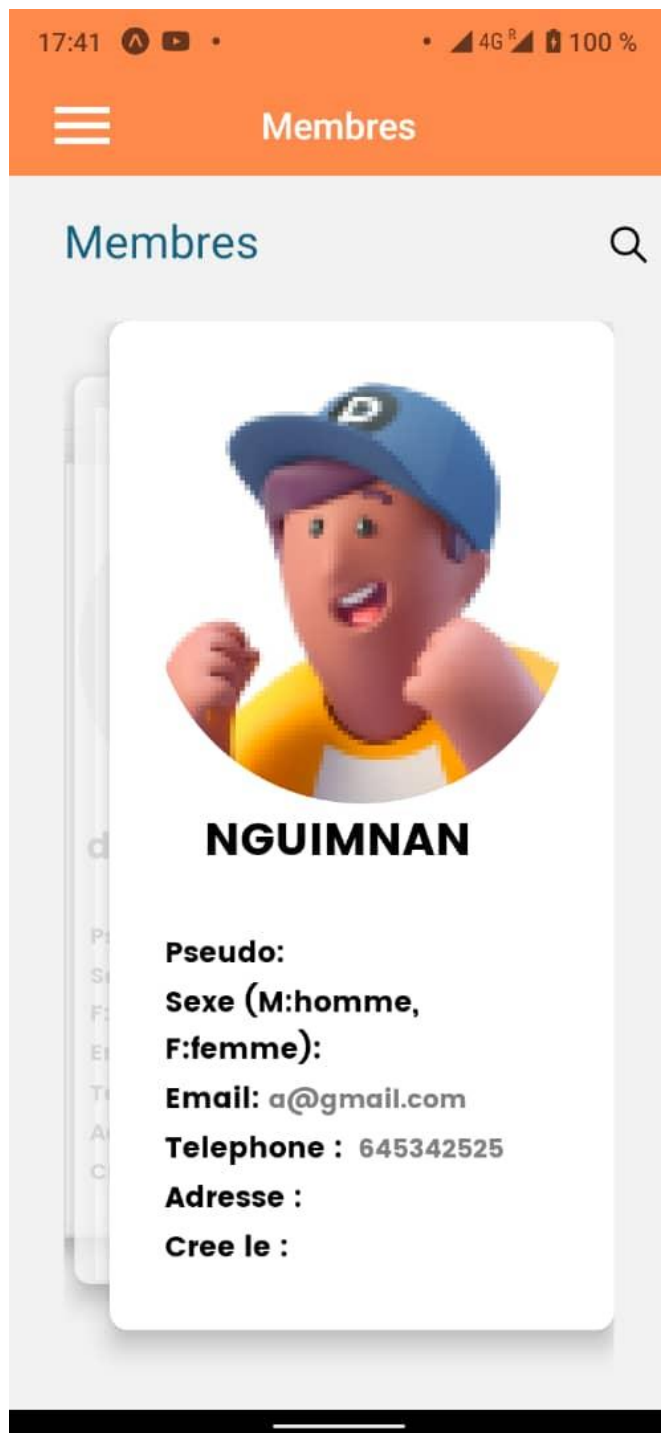


Figure 39: interface liste des membres

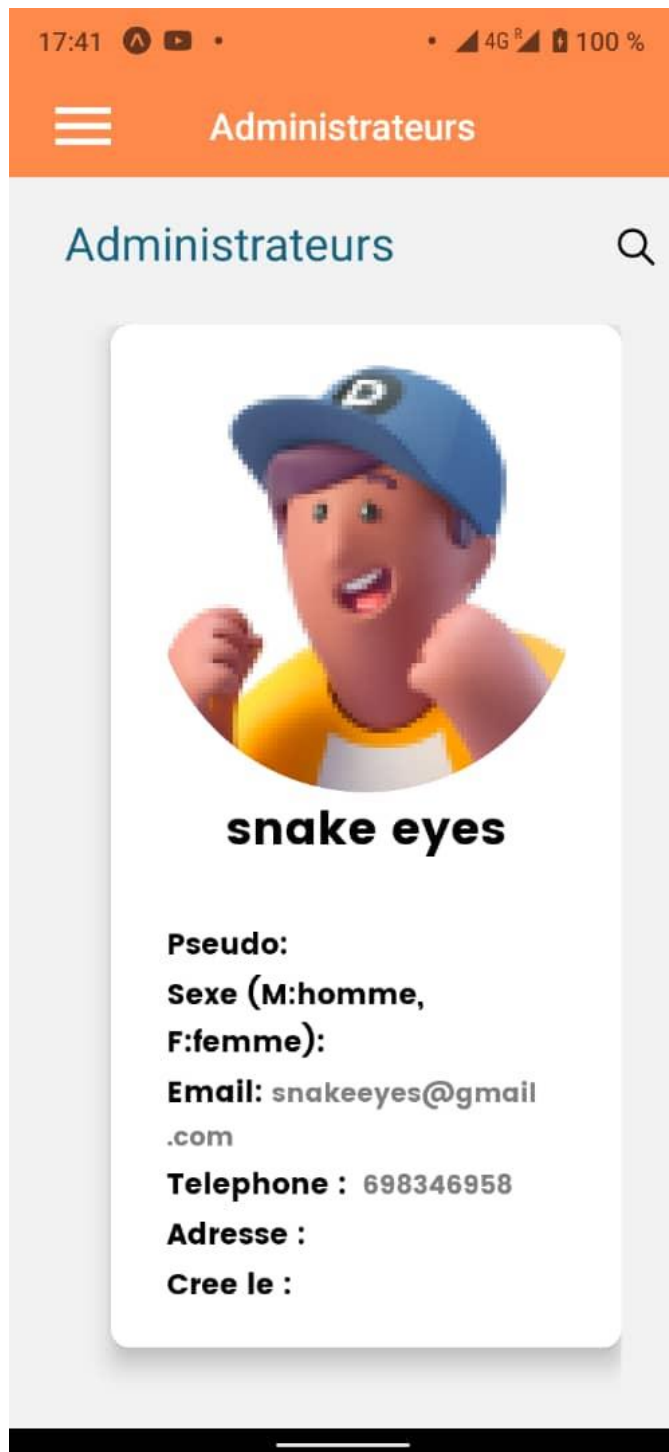


Figure 40: interface responsable des opérations

#### 4. La navbar

Elle permet d'avoir accès à des opérations supplémentaires

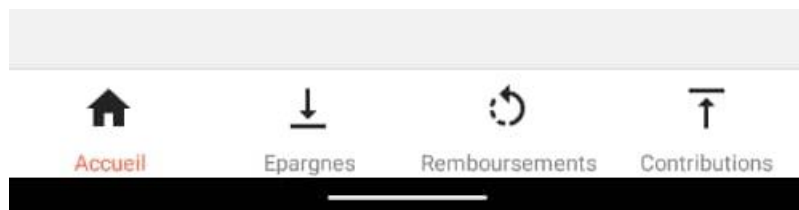


Figure 41: La navbar

## CONCLUSION GÉNÉRALE

Ce projet nous a permis de concevoir, développer et intégrer une application mobile sous la plateforme Android qui a fortement occupé le marché des terminaux mobiles. La réalisation de ce travail est organisée en six phases. Nous avons dans le premier présenté les différentes notions utilisées dans le développement des applications mobiles à savoir, le système d'exploitation Android, la mutuelle et les bases de données et nous présentons notre méthodologie c'est-à-dire comment nous avons modélisé le problème à résoudre et la méthode de gestion de projet dont on fait usage. Le deuxième chapitre portait sur l'analyse du projet à travers une analyse des besoins, une détermination des acteurs du système, une modélisation de quelques diagrammes d'analyse (contexte, package, cas d'utilisation, classe et activité, état transition). Le chapitre trois décrivait la conception du système à travers une présentation de l'architecture de la solution, quelques diagrammes de conception (diagramme de classe technique, diagramme de séquence technique). Nous sommes passés ensuite au chapitre quatre qui est une présentation de l'implémentation et les tests à travers les outils et les technologies utilisés ainsi que des résultats obtenus. Néanmoins, plusieurs perspectives restent envisageables, comme l'ajout des notifications, le mode hors ligne, des discussions, ...

## RÉFÉRENCES

- [En ligne]. Available:<https://www.taktilcommunication.com/blog/applicationsmobile/definition-typologie-applications-mobiles.html>
- [En ligne]. Available:<https://www.taktilcommunication.com/blog/applicationsmobile/definition-typologie-applications-mobiles.html>)
- [Mémoire] : Une application Smartphone pour la gestion et la géo-localisation des pharmacies sous Android
- [Mémoire] : Conception-et-realisation-dune-application-mobile-sous-Android-derecherche-de-corps-medicaux
- [En ligne]. Available : <http://www.phonandroid.com/toute-l-histoire-et-la-chronologie-dandroid-dossier.html>
- Livre de Bases de données GEORGES GARDARIN EYROLLES (www.prism.uvsq.fr/~gardarin/home.html)
- [En ligne]. Available : [https://jmbruel.github.io/MobileModeling2016-2/plateform\\_dev.html](https://jmbruel.github.io/MobileModeling2016-2/plateform_dev.html)
- <https://www.docstring.fr/blog/deployer-une-application-django-sur-pythonanywhere/>
- Google
- Wikipedia