

Redon BRAHIMI
Gabriel SCHOENFELDER
Abdessalam KHATTRI
Kilian NAULEAU
Anas KIOUAZ
Yanis DEBBAL
Rayan NYABI

Cadrage et organisation du projet Weave



Tables des matières :

1. Cadrage	3
1.1 Contexte	3
1.2 Objectifs du projet	3
1.3 Présentation du produit	4
1.4 Contraintes	4
1.5 Parties prenantes	4
1.6 Personas	4
1.6.1 Jeanne	4
1.6.2 Thomas	5
1.6.3 Laurie	5
1.7 EPIC et UserStory	5
1.8 Risques	5
1.8.1 Risque(s) rencontré(s)	6
2. Recueil des Besoins	6
2.1 Besoins fonctionnels	6
2.2 Besoins non fonctionnels	7
2.2.1 Contraintes	7
2.2.2 Maquette	7
2.2.3 Modélisation des cas d'utilisation	8
2.3 Besoin de service	9
3. Réalisation	11
3.1 Choix Techniques	11
3.1.1 Frontend (Interface Utilisateur)	11
3.1.2 Backend (API & Logique)	11
3.1.3 Base de Données	11
3.2 Organisation du travail	11
3.2.1 Répartition du temps	12
3.2.2 Configuration de l'Environnement de Développement (Docker)	12
3.2.3 Sécurité & RGPD (Implémentation Technique)	12
3.2.4 Pipeline CI/CD & Workflow	13
3.2.5 Répartition des rôles	13
3.3 Processus au fil du temps	13
3.3.1 Diagramme de classe	13
3.3.2 Diagramme de séquence	15
3.4 Préparation des Tests Post Réalisation	15
4. Conclusion	16
5. Tables des Annexes	17

1. Cadrage

1.1 Contexte

Face au vieillissement de la population et à la perte d'autonomie associée, l'organisation du soutien quotidien (courses, lien social, etc...) est complexe, les outils de coordination efficaces manquent.

Nous proposons une application web collaborative d'emploi du temps solidaire inspirée des *Voisins Solidaires*, mais avec une dimension organisationnelle structurante. Cette plateforme permettra aux proches, voisins et bénévoles de s'inscrire sur des créneaux pour les tâches d'aide, favorisant l'entraide locale et le maintien à domicile de qualité.

L'objectif est de fluidifier la coordination de l'aide et de renforcer le lien social, via une application simple, accessible et sécurisée.

Ce rapport détaille la phase de cadrage du projet, incluant la définition des besoins, l'analyse des contraintes, des risques et l'étude de faisabilité technique.

1.2 Objectifs du projet

- Notre objectif est de concevoir une application mobile intuitive destinée à un public peu familiarisé avec les outils numériques (personnes âgées ou jeunes). Cette application vise à soutenir les personnes âgées en situation de perte d'autonomie, nécessitant une assistance régulière pour les activités de la vie quotidienne, ainsi que leurs aidants proches (famille, amis, voisins).
- L'application permettra aux personnes concernées d'améliorer l'organisation de leurs activités, de fluidifier la communication et de prévenir les malentendus ou les problèmes de coordination. Nous aspirons à rendre la vie des bénéficiaires et des aidants plus paisible, sereine et prévisible.

1.3 Présentation du produit

Weave est une plateforme web d'entraide locale pensée pour faciliter l'organisation du soutien autour des personnes en perte d'autonomie.

Aujourd'hui, les familles, voisins et bénévoles veulent aider, mais manquent d'outils simples pour se coordonner, répartir les tâches et assurer un suivi fiable du quotidien : courses, visites, appels, accompagnements, vérification du moral, etc.

Weave apporte une solution concrète :

une application collaborative qui permet à l'entourage d'une personne dépendante de s'inscrire sur des créneaux d'aide, de visualiser le planning, de communiquer efficacement et de suivre l'état moral et les besoins du bénéficiaire.

1.4 Contraintes

- - Temporelle : Retard dans le développement, charge de travail dans des délais courts.
- - Qualité : Rendre un travail soigné et de qualité et correspondant aux critères demandés.
- - Économique : Hébergement potentiellement coûteux, ressources limitées techniquement.
- - Humaine : Manque de communication dans l'équipe, fatigue car projets longs.
- - Technique : Difficulté à synchroniser les tâches en temps réel / Fuite de données personnelles / Compatibilité mobile-web

1.5 Parties prenantes

- **Personnes âgées dépendantes** : Il s'agit des individus ayant des difficultés à s'autogérer complètement et qui ont ponctuellement besoin d'aide pour des tâches essentielles comme la prise de médicaments, les déplacements (rendez-vous), ou les tâches ménagères (lessive, etc.), nécessitant ainsi un soutien extérieur.
- **Aidants (familiaux, proches)** : Ils englobent les membres de la famille, les voisins ou les amis de la personne aidée. Leur motivation est soit de fournir une assistance concrète à un proche, soit simplement de passer du temps de qualité avec celui-ci.
- **Bénévoles** : Bien que plus rares, ces individus offrent gracieusement leur temps pour aider ou passer du temps avec la personne âgée, sans contrepartie financière.

1.6 Personas

Voici différents "personas" pour rajouter du contexte à notre projet, ce sont des personnages fictifs (avatars) pour représenter les clients cibles que nous pouvons potentiellement atteindre avec notre application.

** Les photos des personas sont fictives, venant du site <https://thispersondoesnotexist.com> **

1.6.1 Jeanne

Ce persona est une représentation de la personne aidée, cela nous permet de mieux représenter cette personne bien qu'elle ne soit pas un utilisateur de l'application (application destinée aux aidants). Mais ce persona est utile afin de mettre en contexte l'application et les autres personas.

[Se rendre à l'annexe de Jeanne](#)

1.6.2 Thomas

Le persona Thomas est l'utilisateur type d'un aidant principal, il est le fils direct de la personne dépendante (Jeanne) donc il sait qui accepter, il connaît ses restrictions et problèmes médicaux. Ce persona est très important car l'application utilise un aidant principal (aidant initiateur) pour la sécurité et pour simplifier certaines fonctionnalités.

[Se rendre à l'annexe de Thomas](#)

1.6.3 Laurie

Laurie est un persona représentant un aidant "secondaire", elle viendra aider si besoin et si possible, ce genre de profil sera le plus commun car il y aura plusieurs aidants secondaires et un seul aidant principal par personne aidée. Ce persona est donc tout aussi important pour nous car il représente une grande partie des utilisateurs et fonctionnalités de notre application.

[Se rendre à l'annexe de Laurie](#)

1.7 EPIC et UserStory

En plus des personas, afin d'encore mieux représenter les fonctionnalités nécessaires nous avons réalisé des User Story, ce sont de simples descriptions d'un besoin ou d'une attente exprimée par un utilisateur. Dû au grand nombre de User Story, nous les avons regroupés en entités nommées EPIC, ce sont des regroupements de User Story dans différentes catégories distinctes afin de rendre plus clair la lecture des User Story.

[Se rendre à l'annexe des User Story et EPIC](#)

1.8 Risques

Voir annexes pour les tableaux de criticités et de mesures pour évaluations ⇒ [\(figure 5.1 et 5.12\)](#)

Tableau d'évaluation et de gestion :

Risque	Type	Criticité	Mesures et Gestion
Retard dans le respect des deadlines IUT	Organisationnel / Temps	Forte	Mesure corrective : Planification précise (Trello / GitLab), chef de projet responsable du suivi. Priorisation d'un MVP fonctionnel.
Mauvaise communication	Humain	Faible	Acceptation : Réunions régulières, utilisation de Discord / GitLab issues, rôles bien définis.
Absence d'un membre clé (maladie)	Humain	Moyenne	Polyvalence des membres, partage des connaissances, binômes.
Données sensibles	Sécurité / Légal	Moyenne	Anticipation Respect du RGPD, pas de stockage de données réelles, chiffrement des données.

Risque	Type	Criticité	Mesures et Gestion
(médicales) mal protégées			
Difficultés tech(hébergement, db, framework)	Technique	Moyenne	Recherche précoce de solutions techniques, tests dès la phase 2.
Application trop complexe pour les aidants âgés	Fonctionnel / UX	Forte	Tests utilisateurs simulés, maquettes simples, ergonomie prioritaire.

1.8.1 Risque(s) rencontré(s)

Lors du choix de l'hébergement nous avons initialement choisis un serveur privé appartenant à un proche (non-membre de l'équipe). Cependant, des raisons externes nous ont obligés à faire un changement, en effet nous avons dû déplacer toute l'architecture sur un autre serveur, tout est expliqué dans la partie [Réalisation/Choix Techniques](#).

2. Recueil des Besoins

2.1 Besoins fonctionnels

L'application devra permettre à chaque utilisateur de créer un compte personnel, de rejoindre un groupe d'aidants et de collaborer autour d'une personne âgée.

Un calendrier partagé constitue le cœur du système, il permet de planifier les tâches, répartir les rôles et recevoir des rappels automatiques.

Une messagerie intégrée facilitera la communication entre les membres du groupe, tandis qu'un journal de bord permettra de suivre l'état moral de la personne aidée et d'assurer un suivi global.

La sécurité des données sera primordiale, avec un accès limité aux membres autorisés et une conformité stricte au RGPD. En effet, bien que notre application tourne autour des personnes dépendantes (à cause d'une maladie ou non), nous ne stockons pas vraiment l'état de santé de la personne. Cependant, les aidants peuvent utiliser l'application afin de rappeler de prendre des médicaments par exemple, cela peut être une donnée sensible qui identifie une maladie particulière.

Les besoins fonctionnels sont la traduction technique des tâches identifiées dans les diagrammes BPMN. Ils sont formulés ci-dessous sous forme de *User Stories* (Récits Utilisateurs) pour centrer le développement sur la valeur apportée. ([Voir annexe User Story](#))

2.2 Besoins non fonctionnels

2.2.1 Contraintes

Conformément aux principes de gestion de projet, nous distinguons ici les **contraintes** (éléments imposés et invariables) des besoins fonctionnels. Ces exigences conditionnent la viabilité du projet.

A. Contraintes Légales et de Sécurité (Critique)

L'application manipulant des données à caractère personnel et sensibles (niveau de dépendance GIR, incidents de santé, présence au domicile), le respect du cadre légal est une contrainte absolue.

- **Conformité RGPD** : Mise en œuvre du principe de *Privacy by Design*. L'accès aux données doit être cloisonné : un bénévole ponctuel ne doit accéder qu'aux informations strictement nécessaires à sa mission (principe de minimisation), sans visibilité sur le dossier global du bénéficiaire.
- **Hébergement des données** : Les données doivent être stockées de manière sécurisée (chiffrement en base de données) pour prévenir toute fuite d'information.

B. Contraintes d'Ergonomie et d'Accessibilité

La population cible incluant des aidants âgés (conjointes) et potentiellement les bénéficiaires eux-mêmes, l'interface doit pallier la fracture numérique.

- **Accessibilité (UI/UX)** : Respect des normes de contraste et de lisibilité (taille des polices, zones de clic larges) pour s'adapter aux troubles visuels ou moteurs légers (en lien avec le persona de Laurie qui peut avoir des problèmes avec les technologies).
- **Simplicité cognitive** : Le nombre d'actions pour réaliser une tâche critique (signaler un problème) doit être minimisé (max. 3 clics).

2.2.2 Maquette

Pour le bon déroulement du projet nous avons fait des maquettes, une pour téléphone et une pour ordinateur. Cela nous permet d'avoir une bonne vision d'ensemble de ce que nous allons devoir réaliser dans la phase 3.

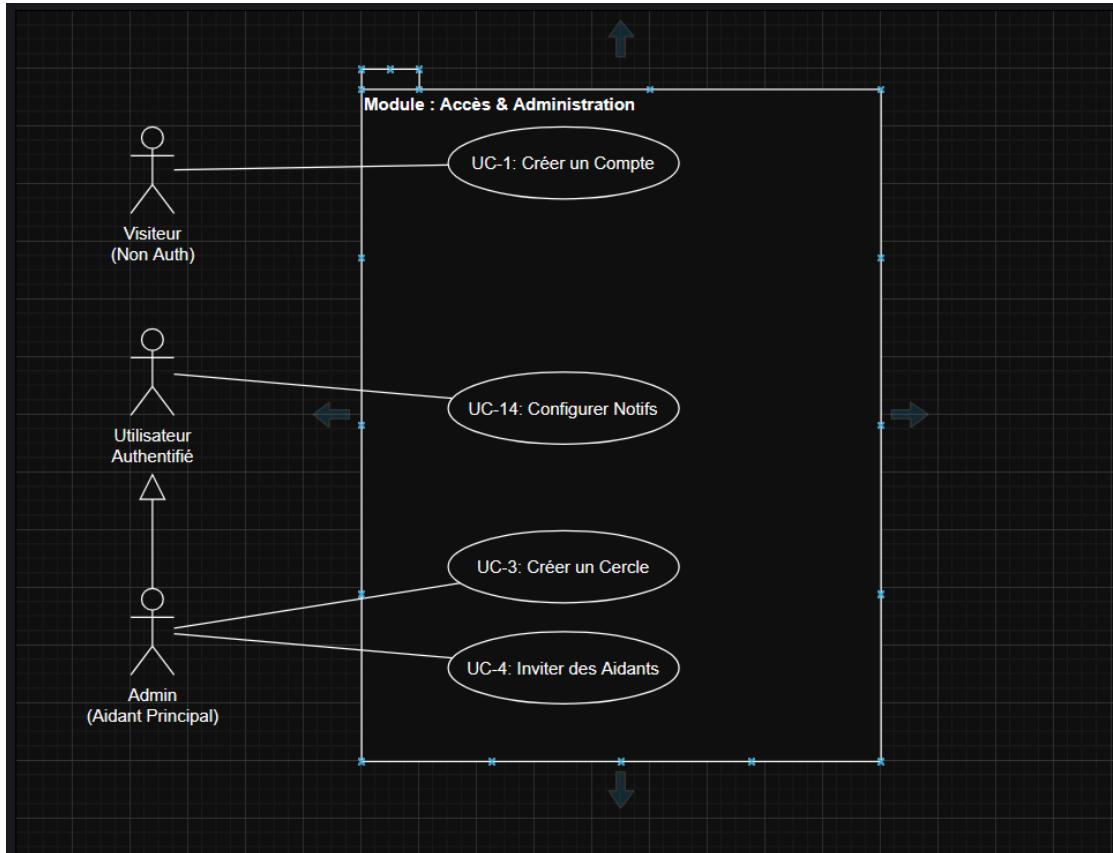
Ci dessous vous trouverez des screens des maquettes ainsi que les liens pour y accéder directement.

[Lien Telephone](#) | [Lien Ordinateur](#)

The image displays two wireframe prototypes of the Weave platform interface. The left prototype is designed for mobile devices (phone and tablet) and features a top header with 'Bonjour, Sophie' and a notification bell icon. Below this is a contact card for 'Marcelle Dubois' with a status of 'Bon moral' and a 'Stable' badge. The main area shows three large buttons: 'Rendez-vous' (5), 'Aidants' (8), and 'Moments' (12). A section for the current date ('Aujourd'hui - Mardi 2 Décembre') shows a scheduled 'Visite médicale' at 14:00 for 'Marie'. At the bottom is a navigation bar with icons for Accueil, Calendrier, Souvenirs, Messages, Profil, and Admin. The right prototype is designed for desktop computers and features a top header with the Weave logo and a 'Tableau de bord' section. This section includes a summary of 'Aidants actifs: 8', 'Tâches cette semaine: 12', 'Souvenirs partagés: 45', and 'Messages non lus: 3'. Below this is a 'État de santé' section showing 'Moral: Bon' and 'Santé générale: Stable'. A 'Dernier contact: Aujourd'hui 10:00' is also indicated. The final section is 'Prochaines interventions' listing 'Visite médicale' at 14:00 for 'Marie Dupont', 'Courses alimentaires' at 16:30 'À pourvoir', and 'Promenade au parc' 'Demain 10:00' for 'Jean Martin'. At the bottom is a large orange 'Urgence' button.

Les maquettes ne sont pas définitive mais donne une première version à ce que l'application ressemblera, les fonctionnalités et/ou éléments pourront être supprimés/ajoutés.

2.2.3 Modélisation des cas d'utilisation



Ce premier module pose les bases de la sécurité de l'application. Nous avons distingué deux types de personnes à l'entrée : le simple Visiteur, qui ne peut que se créer un compte, et l'Utilisateur Authentifié qui a accès au reste.

L'architecture repose sur une hiérarchie simple : l'administrateur (souvent l'Aidant Principal) est un utilisateur comme les autres, mais avec des droits supplémentaires. C'est lui seul qui pilote le Cercle de Soins : il le crée et invite les autres bénévoles. Enfin, pour respecter les standards actuels, nous n'avons pas mis la connexion comme une fonctionnalité à part entière, mais comme une sécurité obligatoire pour accéder à tout le reste.

[Voir les diagrammes de cas d'utilisation](#)

2.3 Besoin de service

Cette section répertorie les engagements de service que l'application Weave doit fournir pour garantir son utilité et sa fiabilité auprès des aidants et des bénéficiaires.

Services Métiers (Cœur de l'application)

Numéro BNS	Besoin de service	Explication

BNS1	Service Coordination Planification de et	Le service doit permettre la centralisation de l'agenda du bénéficiaire et garantir la bonne affectation des bénévoles aux tâches (courses, visites) sans doublons ni oubli (gestion des quotas).
BNS2	Service de Lien Social et Suivi	Le service doit fournir un espace d'échange fluide (Messagerie, Journal de bord) pour lutter contre l'isolement du senior et permettre aux aidants de partager des informations qualitatives (humeur, souvenirs).

Qualité et Performance (Contraintes techniques)

Numéro BNS	Besoin de service	Explication
BNS3	Service de disponibilité (Incidents)	Le service de gestion des incidents doit être opérationnel 24h/24 et 7j/7. Le système doit garantir l'exécution des tâches de fond (Timers d'escalade) sans interruption pour assurer la sécurité des personnes.
BNS4	Service d'Accessibilité et Ergonomie	Le service doit offrir une interface adaptée aux populations cibles (Seniors et Aidants âgés) en respectant les normes d'accessibilité (taille des textes, contrastes, navigation simplifiée au moins 3 clics).
BNS5	Service de Réactivité (Feedback)	Le service doit garantir un temps de réponse rapide, notamment pour l'envoi des notifications Push d'urgence (délai < 5 secondes) afin de ne pas retarder une prise en charge critique.
BNS6	Service de Portabilité (Multi-supports)	Le service doit assurer une expérience cohérente et fonctionnelle quel que soit le terminal utilisé : ordinateur pour l'administration (Web) ou smartphone pour les bénévoles en déplacement (Responsive/Mobile).

Orienté Sécurité et Confidentialité

Numéro BNS	Besoin de service	Explication
BNS6	Service de Protection des Données de Santé	Le service doit garantir la stricte confidentialité des données sensibles (notes médicales, code d'accès domicile) via un chiffrement en base de données et une conformité RGPD (Privacy by Design).
BNS7	Service de Gestion des Habilitations	Le service doit assurer un cloisonnement strict des accès : un bénévole ponctuel ne doit voir que les informations nécessaires à sa mission, tandis que l'Administrateur (Aidant Principal) conserve le contrôle total du Cercle.

3. Réalisation

depot git : <https://github.com/anaskiouaz/WEAVE>

3.1 Choix Techniques

3.1.1 Frontend (Interface Utilisateur)

- **Langage & Framework** : JavaScript / React 19.2.
- **Build Tool** : Vite 7.2.
- **Styling** : TailwindCSS 4.0.
- **Composants & Icônes** : lucide-react pour les icônes.
- **Génération PDF** : jspdf.
- **Routing** : react-router-dom v7.
- **Hébergement Production** : Vercel (Déploiement Edge/Serverless).

3.1.2 Backend (API & Logique)

- **Runtime** : Node.js 22.21.1.
- **Framework** : Express v4 .

- **Hébergement Production** : Azure App Service (Web App for Containers).
 - *Nom* : weave-be-server
 - *OS* : Linux (Conteneur Docker).
 - *Plan* : B1.
- **Sécurité & Auth** :
 - Authentication : JSON Web Tokens (JWT) (`jsonwebtoken`).
 - Hachage mots de passe : `bcryptjs`.
 - CORS : Configuré pour accepter uniquement le frontend.

3.1.3 Base de Données

- **SGBD** : PostgreSQL.
- **Hébergement Production** : Azure Database for PostgreSQL (Serveur Flexible).
 - *Nom* : weave-db-server.
 - *Version* : 17.7.
 - *Configuration* : B2s (Standard), 32 Gio stockage.

3.2 Organisation du travail

Outils utilisés

Pour assurer une coordination efficace de l'équipe et un suivi rigoureux du projet, plusieurs outils ont été utilisés :

- **Discord** : pour la communication instantanée et les échanges informels entre membres.
- **Google Drive** : pour le partage et la centralisation des documents, maquettes, rapports et supports de travail.
- **GitHub** : pour le suivi du code, la gestion des versions et le traitement des issues/problèmes.
- **GitLab Issues** : pour la planification des tâches, le suivi de l'avancement et la gestion des priorités.
- **Linear** : pour la gestion des tâches et du temps

3.2.1 Répartition du temps

Le projet a été organisé sur plusieurs phases principales :

1. **Conception** : jusqu'en décembre
 - Définition des besoins, étude des personas, analyse des contraintes et risques
 - Maquette et prototype initial
2. **Développement** : trois premières semaines de janvier
 - Backend : création de la base de données, API, sécurité
 - Frontend : interface web et mobile, intégration des fonctionnalités
 - Tests unitaires et validation de chaque module
 -
3. **Coordination et suivi** : tout au long de l'année
 - Réunions hebdomadaires pour faire le point sur l'avancement
 - Suivi des tâches et réajustements via Trello/GitLab
 - Suivi de la communication et des interactions utilisateurs

3.2.2 Configuration de l'Environnement de Développement (Docker)

Pour garantir que tous les développeurs travaillent sur la même base, nous utilisons docker-compose. Cela permet de lancer la base de données, l'API et le frontend avec une seule commande.

Fichier `docker-compose.yml` actuel :

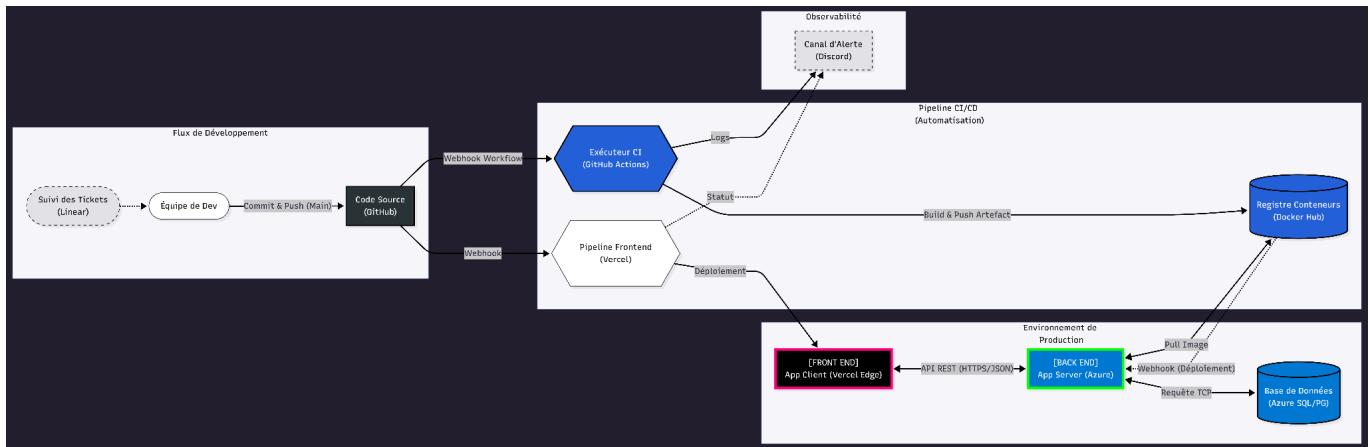
- **Services :**
 - postgres : Instance locale isolée (Port 5432).
 - api : Backend Node.js avec rechargement à chaud (Volume monté sur ./be), tourne sur le port 4000.
 - frontend : Serveur de développement Vite (Volume monté sur ./fe), exposé sur le port 5173.
- **Réseau :** Création d'un bridge network weave-net pour que l'API puisse parler à la BDD via le nom d'hôte postgres.

3.2.3 Sécurité & RGPD (Implémentation Technique)

Les concepts évoqués dans le cadrage sont implémentés ainsi :

1. **Authentification Sécurisée** : Utilisation de bcryptjs pour ne jamais stocker de mots de passe en clair et jsonwebtoken (JWT) pour gérer les sessions sans état (stateless) entre Vercel et Azure.
2. **Gestion des Rôles** : Gérée au niveau applicatif (Middleware Express) pour restreindre l'accès aux routes sensibles.
3. **Confidentialité** :
 - L'API Azure n'est accessible que via HTTPS.
 - Les variables d'environnement (Mots de passe BDD, JWT Secret) ne sont **jamais** committées sur GitHub (utilisation de .env). les variables d'environnement ne sont ni public ni dans le répertoire git
 - CORS restreint : L'API n'accepte les requêtes que venant de localhost (en dev) ou du domaine Vercel (en prod).

3.2.4 Pipeline CI/CD & Workflow



[workflow mermaid](#)

3.2.5 Répartition des rôles

Chaque membre de l'équipe a eu des responsabilités claires pour éviter les chevauchements et assurer l'efficacité du projet :

- **Chef de projet** : Yanis Debbal - suivi global, planification et coordination.
- **Responsable Conception** : Abdessalam Khattri, Kilian Nauleau - conception des maquettes et de l'interface.
- **Responsable Réalisation** : Gabriel Schoenfelder, Anas Kiouaz - développement technique backend et frontend.
- **Responsable Communication** : Rayan Nyabi - suivi de la communication, documentation et présentation.
- **Responsable Anglais** : Redon Brahimi - relecture et adaptation en anglais.

Le tableau complet de répartition des rôles (RACI) avec les détails sur qui est Responsable,

Approuve, Consulté et Informé est présenté en [Annexe 4.3](#).

3.3 Processus au fil du temps

3.3.1 Diagramme de classe

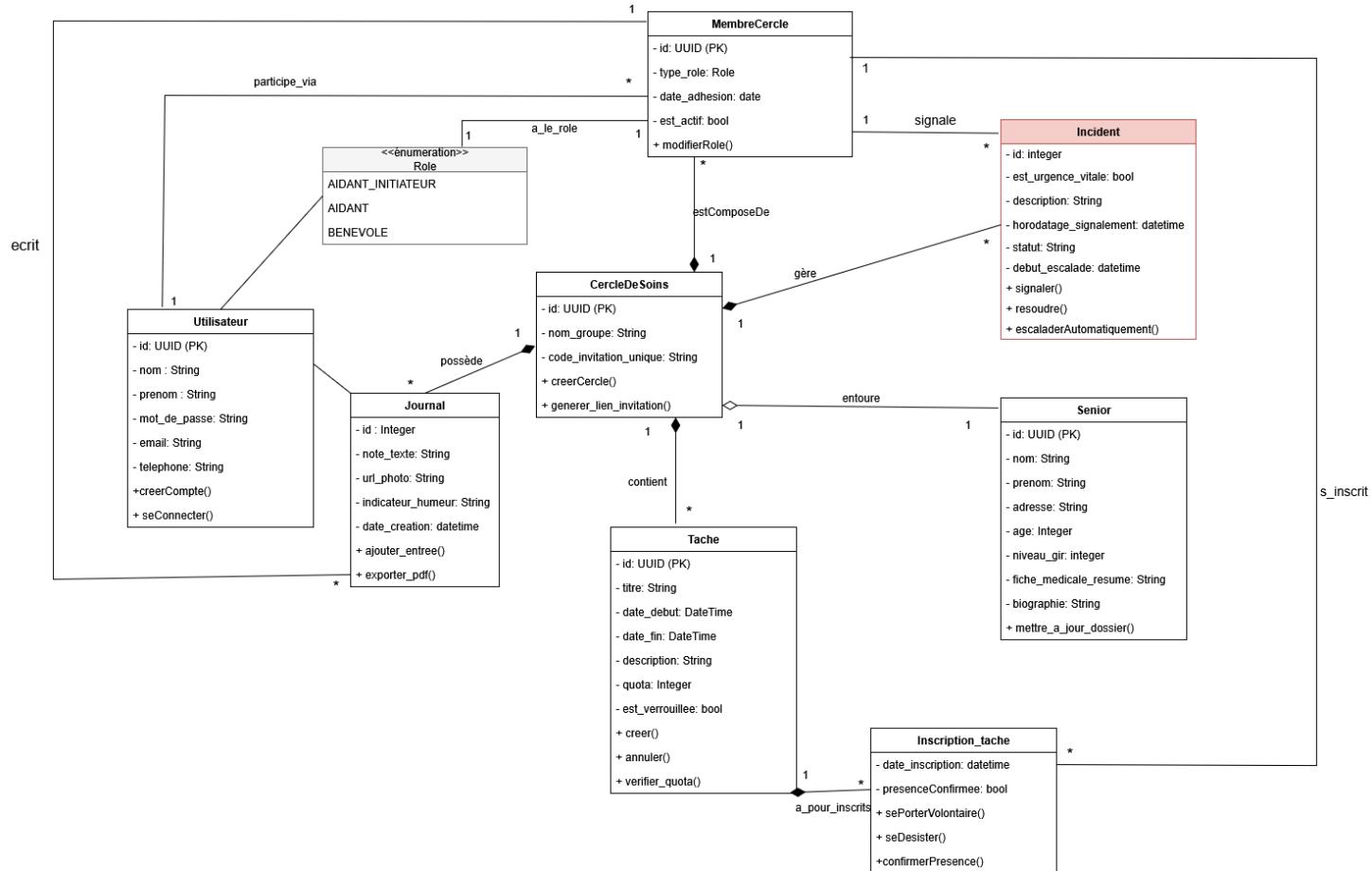


Diagramme de classe.

Notre modèle de données est centré sur l'entité **CercleDeSoins**. Il utilise une structure de composition forte pour regrouper de manière étanche les tâches, les incidents et le journal autour du bénéficiaire (**Senior**). La gestion des accès passe par une classe intermédiaire **MembreCercle**, ce qui nous permet de gérer finement les rôles (Admin/Bénévole) au sein de chaque groupe.

3.3.2 Diagramme de séquence

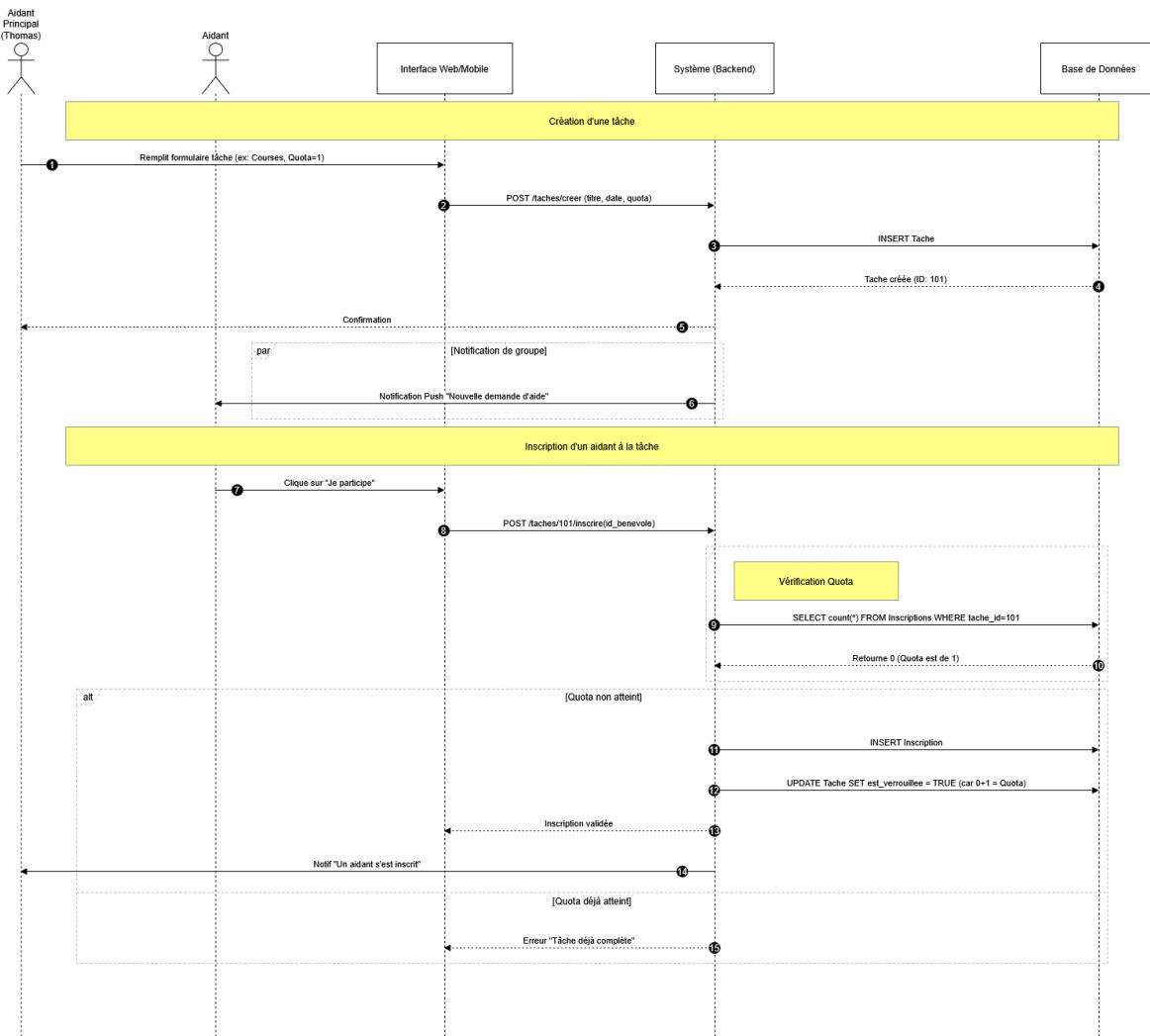


Diagramme de séquence : Gestion de création de tâche et inscription.

Ce diagramme de séquence détaille la mécanique technique derrière l'entraide. Il met en évidence notre architecture 3-tiers (App, API, BDD) et surtout notre gestion de la concurrence. Grâce à la vérification des quotas en temps réel avant chaque inscription, nous empêchons les doublons et garantissons que chaque tâche reçoit exactement le nombre de bénévoles requis, ni plus, ni moins.

[Voir le reste des diagrammes de séquence](#)

3.4 Préparation des Tests Post Réalisation

Afin de maximiser nos chances de réussite, nous avons décidé de faire des tests pour vérifier que notre application soit crédible aux yeux des utilisateurs ciblés. C'est pourquoi nous allons mettre en place des tests, nous avons estimé que réaliser des tests avec la méthode qualitative "Think Aloud" était plus

approprié que la méthode quantitative “SUS” car malgré que la méthode “SUS” permette un test sur un plus grand nombre de sujet, nous pensons que nous aurons des retour plus précis et constructif de manière à faire évoluer notre application dans le bon sens.

Malgré tout nous avons étudié la méthode “SUS” de manière à confirmer notre choix de type de test. Cette méthode a pour principe de faire effectuer un test du logiciel puis noter l'utilisabilité à l'aide d'un questionnaire standard et issue d'études scientifiques. Après la phase de questionnaire il faut calculer des scores à l'aide d'une méthode bien précise pour pouvoir l'interpréter avec une échelle de degré d'acceptabilité du système évaluer pouvant aller de Non Acceptable (Worst imaginable) a Acceptable (Best imaginable) en passant par une Acceptabilité Probable (OK).

Maintenant venant au test choisis par notre équipe, la méthode “Think Aloud” qui vise à demander aux utilisateurs de verbaliser leurs pensées et leurs actions lors du test du logiciel de manière à les noter et faire des tableau de résultat avec des point négatif, positif ou encore des suggestion d'amélioration et blocage. Il faut faire attention à être bien neutre, il faut aussi chronométrier et finir le test (à la toute fin) par une petite série de questions globales de ressentiment. Nous avons prévu de faire passer le test à des membres de notre famille et à des inconnues susceptible de se servir de l'application et n'étant pas familier avec les technologies.

4. Conclusion

Le projet **Weave** répond à un besoin croissant d'organisation et d'entraide autour des personnes âgées en perte d'autonomie. Ce cadrage a permis de définir clairement les objectifs, les utilisateurs, les fonctionnalités clés et les contraintes du projet. L'analyse des besoins et des personas confirme l'importance d'une plateforme simple, sécurisée et centrée sur la coordination entre aidants.

L'étude de faisabilité montre que le projet est techniquement réalisable grâce à une architecture backend–frontend solide et des technologies adaptées. Les principaux défis concernent surtout la sécurité des données, la conformité réglementaire et la gestion de la communauté d'utilisateurs.

Ce travail fournit ainsi une base structurée pour la suite du développement. Les prochaines étapes consisteront à concevoir puis développer un prototype afin de concrétiser le produit.

5. Tables des Annexes

5.1 Contraintes et risques – tableaux	20
a) Tableau de la Criticité (fig 5.1) :	20
b) Tableau des Mesures (fig 5.12):	20
5.2 Diagramme BPMN	21
Processus de planification d'une demande :	21
Processus de gestion d'incident :	22
5.3 Tableau des rôles (RACI)	23
1. Identification des rôles (RACI)	23
5.3 Diagramme de Gantt (Jira)	24
5.4 Personas clairs	25
5.4.1 Jeanne	25
5.4.2 Thomas	25
5.4.3 Laurie	26
5.4 Faisabilité et Veille	26
5.4.1 Faisabilité Technique	26
Base de données (SQL)	26
Backend (API REST / API sécurisée)	26
Frontend (Web et Mobile)	27
Notifications (Push / Email)	27
Sécurité des données	28
5.4.2 Faisabilité opérationnelle	28
5.4.3 Faisabilité légale et réglementaire	28
5.4.4 Veille concurrentielle	29
5.5 UserStory et EPIC	29
5.5.1 Gérer les utilisateurs, l'authentification et la création de cercles de soutien pour les personnes âgées	29
5.5.1.1 Autoriser les nouveaux utilisateurs à s'inscrire via le site web.	29
5.5.1.2 Autoriser les utilisateurs à se connecter via un navigateur.	30
5.5.1.3 Créez un groupe de soutien autour d'une personne âgée via l'application web.	30
5.5.1.4 Inviter des assistants via un lien Web	30
5.5.1.5 Gérez les rôles au sein du groupe via le web.	30
5.5.2 Fournir un calendrier hebdomadaire collaboratif pour la planification des tâches et la coordination des bénévoles.	31
5.5.2.1 Ajouter des tâches via le calendrier web.	31
5.5.2.2 Mettre à jour les tâches via le site web.	31
5.5.2.3 Les personnes qui proposent leurs services s'inscrivent en ligne pour effectuer les tâches.	32
5.5.2.4 Verrouillage automatique des tâches une fois le quota d'assistant atteint.	32
	16

5.5.2.5 Rappel dans le navigateur ou par courriel pour confirmation.	32
5.5.3 Permettre une communication privée et sécurisée entre les assistants.	33
5.5.3.1 Messages de groupe dans le navigateur.	33
5.5.3.2 Conversation liée aux tâches.	33
5.5.3.3 Notifications de messages sur le Web.	33
5.5.4 Fournir un journal pour que la personne âgée puisse suivre son humeur, ses activités et partager ses souvenirs.	34
5.5.4.1 Ajouter des notes/photos en ligne.	34
5.5.4.2 Sélectionnez votre niveau d'humeur en ligne.	34
5.5.4.3 Afficher les entrées dans le navigateur.	34
5.5.4.4 Télécharger le journal au format PDF/HTML.	34
5.5.5 Gérer les incidents, urgents et non urgents, lors des visites.	35
5.5.5.1 Rapport d'incident en ligne.	35
5.5.5.2 Alerte automatique sur PC.	35
5.5.5.3 Faites remonter l'information si pas de réponse en ligne.	35
5.5.5.4 Résoudre l'incident via le navigateur.	36
5.5.6 Fournir des notifications push et par e-mail pour les tâches, les messages et les incidents.	36
5.5.6.1 Configurer les préférences de notification en ligne.	36
5.5.6.2 Alertes concernant les tâches nouvelles ou mises à jour.	36
5.5.6.3 Les notifications d'incidents ne peuvent pas être désactivées.	37
5.5.7 Protéger les données médicales et personnelles sensibles grâce à des règles de sécurité strictes	37
5.5.7.1 Limiter les données en fonction du rôle sur le web.	37
5.5.7.2 Sécuriser les données sensibles en transit et au repos.	37
5.5.7.3 Recueillir le consentement pour les données sensibles via le web.	38
5.5.7.4 Suivre l'accès aux données.	38
5.5.8 Créer une interface web claire et accessible	38
5.5.8.1 Créer l'interface utilisateur principale pour le web.	38
5.5.8.2 Page d'accueil simple et accessible.	38
5.5.8.3 Gros boutons, polices lisibles, contraste.	39
5.5.9 Modélisation	40
5.5.9.1 Diagramme de cas d'utilisation	40
5.5.9.2 Diagramme de séquence	42

5.1 Contraintes et risques – tableaux

a) Tableau de la Criticité (fig 5.1) :

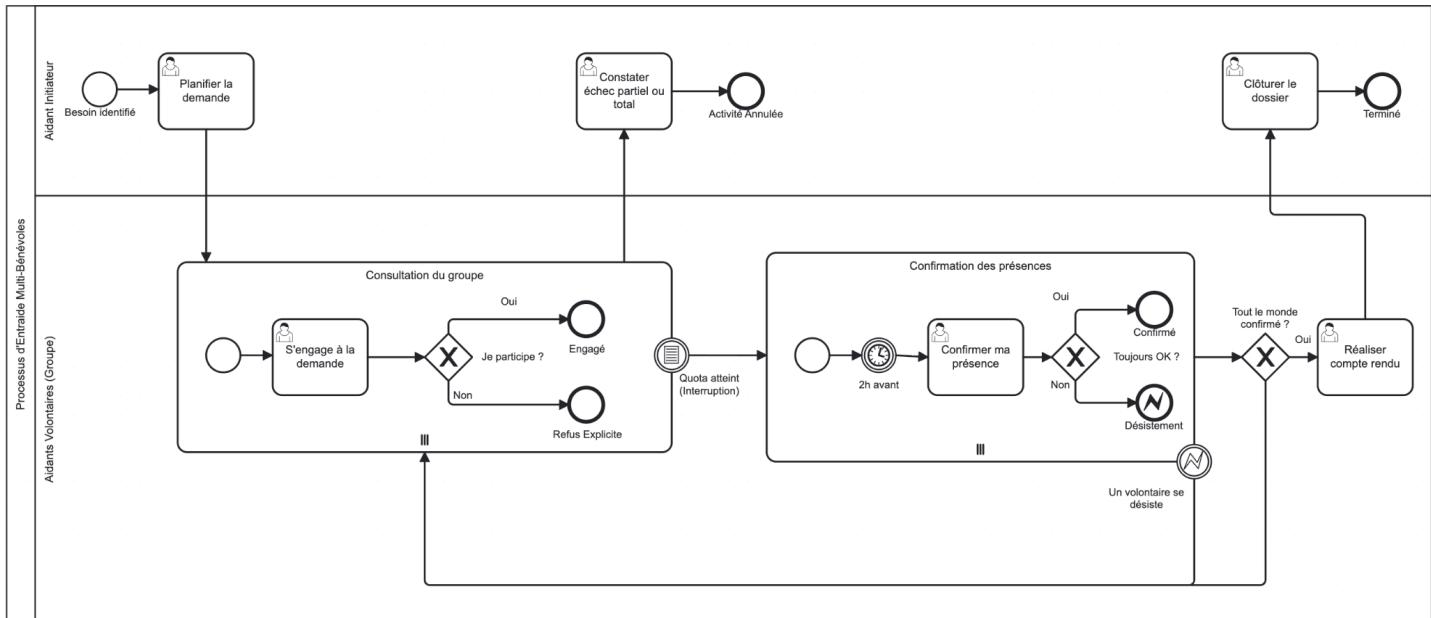
	Probabilité d'apparition (x)				
Impact (y)		Faible 1	Moyenne 2		Forte 3
	Faible 1	Faible 1	Moyenne 2		Moyenne 3
	Moyenne 2	Moyenne 2	Forte 4		Forte 6
	Forte 3	Moyenne 3	Forte 6		Très Forte 9

b) Tableau des Mesures (fig 5.12):

Criticité	Mesure
Faible 1	Acceptation
Moyenne 2-3	Measure corrective
Forte 4-6	Measure corrective / préventive
Très Forte 9	Measure préventive

5.2 Diagramme BPMN

Processus de planification d'une demande :



Le cœur de l'application Weave réside dans sa capacité à transformer un besoin exprimé par la famille en une action concrète réalisée par un bénévole/aidant, sans générer de charge mentale supplémentaire pour les aidants familiaux. Le diagramme BPMN modélisant ce flux a été conçu pour gérer l'incertitude liée au bénévolat (disponibilité, oubli) tout en garantissant que chaque besoin trouve une réponse.

Lancement et Recherche Collective

Tout commence par l'Aidant Initiateur (généralement un proche) qui identifie un besoin précis, comme des courses ou un accompagnement médical. Une fois la demande qualifiée dans l'application, le processus déclenche une phase de recherche active. Contrairement à une approche séquentielle où l'on contacterait les gens un par un, la demande est diffusée simultanément à l'ensemble du groupe de volontaires. Cette phase de "consultation" est gérée de manière autonome : chaque volontaire reçoit la notification et dispose d'un temps imparti pour répondre. Pour éviter les blocages, le système intègre une gestion tacite du refus : si un volontaire ne répond pas dans le délai imparti, il est considéré comme indisponible sans avoir besoin de se justifier, ce qui préserve l'engagement sur le long terme.

Interruption Intelligente et Quota

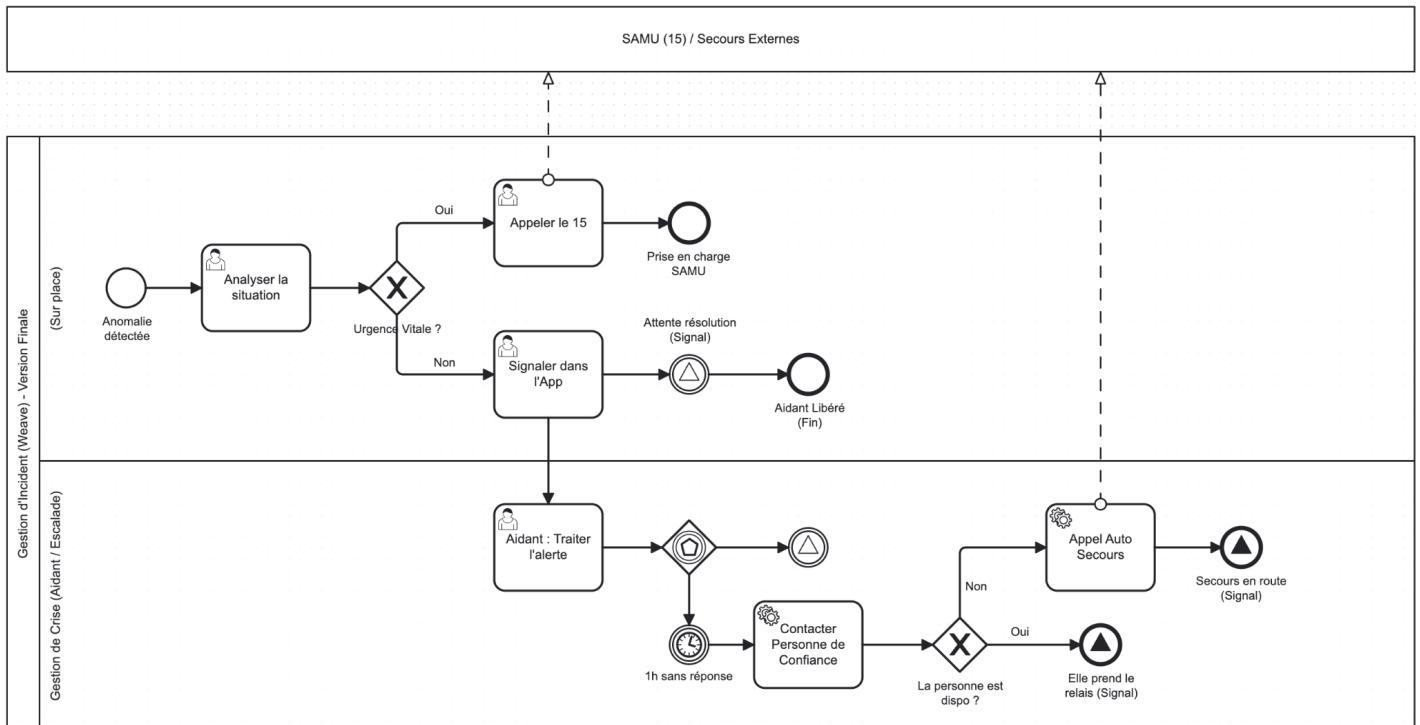
Un point crucial de notre logique est la gestion des ressources humaines. Si une activité nécessite deux personnes, il est inutile que cinq bénévoles s'engagent. Nous avons donc modélisé un mécanisme d'interruption automatique basé sur un quota : dès que le nombre de volontaires requis (par exemple, un seul chauffeur) s'est engagé, la recherche est immédiatement stoppée pour les autres membres du groupe. Cela évite les frustrations et optimise la sollicitation de la communauté.

Sécurisation de l'Engagement (La double confirmation)

L'expérience montre qu'un engagement pris plusieurs jours à l'avance peut être oublié. Pour pallier ce risque opérationnel, le processus intègre une étape de sécurité indispensable : la confirmation de présence. Un délai avant l'activité (paramétrable, par exemple 2 heures avant), le système sollicite à nouveau le volontaire engagé. Ce dernier doit confirmer qu'il est toujours disponible. Si le volontaire confirme, l'activité est validée. En revanche, si un imprévu survient ou si le volontaire ne répond pas, le système ne laisse pas la personne âgée dans l'attente : il détecte l'absence de confirmation comme un désistement et peut, selon la configuration, relancer une recherche

d'urgence ou alerter la famille. Une fois l'activité réalisée, le processus se clôture par une mise à jour de l'historique, assurant la traçabilité des interventions.

Processus de gestion d'incident :



Le second processus, sans doute le plus critique, concerne la sécurité de la personne aidée. Il définit comment l'application réagit lorsqu'un bénévole, présent sur place, détecte une anomalie (chute, confusion, problème domestique). Notre modélisation a été pensée pour couvrir tous les scénarios, du plus bénin au plus grave, en garantissant qu'aucune alerte ne reste sans réponse.

Le Tri Initial et la Priorité Vitale

Dès la détection d'une anomalie, le processus impose un choix binaire strict au bénévole. Si la situation relève d'une urgence vitale, l'application s'efface au profit des procédures de secours classiques : le bénévole doit appeler le 15 (SAMU) immédiatement. L'application ne sert alors qu'à notifier la prise en charge. Si l'urgence n'est pas vitale, le bénévole effectue un signalement via l'application. À cet instant, il entre dans un état d'attente active : il reste auprès de la personne aidée tant qu'il n'a pas reçu le "feu vert" (un signal de libération) indiquant que la situation est prise en main par un tiers.

La Chaîne de Solidarité (Escalade Automatisée)

Une fois le signalement envoyé, la responsabilité bascule vers l'Aidant Principal. C'est ici que notre système de sécurité prend tout son sens. Nous ne pouvons pas supposer que l'aidant principal est disponible à tout moment (il peut travailler, dormir, etc.). C'est pourquoi nous avons intégré un mécanisme de temporisation (Timer). Si l'Aidant Principal traite l'alerte rapidement, il envoie un signal de résolution qui libère le bénévole sur place. Cependant, si l'Aidant Principal ne réagit pas sous un délai critique (par exemple 1 heure), le processus déclenche automatiquement une escalade. L'alerte est transférée à une personne de confiance préalablement identifiée. Le système vérifie la disponibilité de cette personne. Si elle accepte d'intervenir, elle prend le relais et libère le bénévole initial.

L'Ultime Recours

Enfin, notre modélisation couvre le cas du "pire scénario" : si l'Aidant Principal est absent et que la personne de confiance n'est pas disponible. Dans cette impasse, pour ne jamais laisser une personne vulnérable en danger potentiel, le système déclenche une procédure d'appel ultime aux services de secours ou de police pour une levée de doute. Peu importe qui résout finalement l'incident (l'Aidant, la Personne de confiance ou les Secours), l'action finale génère systématiquement un signal technique unique : "Incident Clôturé". Ce signal est capté instantanément par l'interface du bénévole sur place, l'informant qu'il a rempli sa mission et qu'il peut quitter les lieux en toute sérénité. Cette boucle fermée garantit une sécurité maximale pour le bénéficiaire et une clarté totale pour les intervenants.

5.3 Tableau des rôles (RACI)

1. Identification des rôles (RACI)

Pour assurer une organisation claire et efficace du projet, nous avons défini les rôles de chaque membre de l'équipe. Chaque rôle est associé à des responsabilités spécifiques. Nous avons utilisé la matrice RACI pour clarifier qui est Responsable, qui Approuve, qui est Consulté et qui est Informé pour chaque tâche. Cette répartition des rôles permet d'éviter les chevauchements et les confusions, assurant ainsi une meilleure gestion du projet.

Rôle	Nom	Responsable (R)	Approuve (A)	Consulté (C)	Informé (I)
Chef de Projet	Yanis Debbal	R	A	Tous les autres rôles	-
Responsable Communication	Rayan Nyabi	R	A	Chef de projet	Tous les membres de l'équipe
Responsable Réalisation	Gabriel Schoenfelder	R	A	Responsable Conception	Chef de projet
Responsable Réalisation	Anas Kiouaz	R	A	Responsable Conception	Chef de projet
Responsable Anglais	Brahimi Redon	R	A	Chef de projet	Tous les membres de l'équipe
Responsable Conception	Nauleau Kilian	R	A	Responsable conception	Tous les membres de l'équipe
Responsable Conception	Abdessalam Khattri	R	A	Chef de projet	Tous les membres de l'équipe

- **R (Responsable)** : La personne qui effectue le travail.
- **A (Approuve)** : La personne qui valide ou approuve le travail.
- **C (Consulté)** : Les personnes dont l'avis est sollicité.
- **I (Informé)** : Les personnes qui sont tenues informées de l'avancement.

5.3 Diagramme de Gantt (Jira)



5.4 Personas clairs

5.4.1 Jeanne

JEANNE

PROFIL	
Genre	Femme
Age	78 ans
Profession	Retraite, ancienne fleuriste
Situation	Veuve, vit seule dans une maison dans un quartier calme.



“
"Je me débrouille très bien toute seule. Thomas s'inquiète pour rien, mais c'est vrai que j'oublie des choses et que je me sens souvent seule..."

BIOGRAPHIE

Jeanne est une femme atteinte de la maladie de l'Alzheimer (maladie neuro-dévolutive qui fait perdre la dépendance d'une personne au fil du temps). Elle vit seule depuis le décès de son mari et son état d'autonomie est entre GIR2 et GIR3, donc elle a besoin d'aide pour une majorité des soins corporels et une partie des tâches quotidiennes. Cependant, elle néglige son bien-être pensant être un fardeau pour son fils Thomas. En effet, son fils vient l'accompagner tous les jours pour les soins principaux mais Jeanne n'évoque pas son besoin social.

OBJECTIFS

- Ne pas inquiéter Thomas
- Se sentir un peu moins seule
- Avoir une assistance pour ses nécessités, se sentir moins vulnérable

FRUSTRATIONS

- La solitude : Jeanne se sent souvent seule car elle pense être un poids pour ses proches
- La confusion : Jeanne oublie souvent ses médicaments/activités, cela crée de l'anxiété
- La peur : peur de tomber et de ne pas se relever car elle est seule
- L'Inquiétude : ne pas déranger son fils Thomas

AISANCE TECHNOLOGIQUE

Téléphone

Very low

Low

Medium

High

Very high

5.4.2 Thomas

THOMAS

PROFIL	
Genre	Thomas
Age	26 ans
Profession	Etudiant en école d'ingénieur
Situation	Vit seul dans un village à proximité pour aider sa mère.



“
"Je veux juste savoir qu'elle va bien sans avoir à l'appeler toutes les deux heures. Je m'inquiète constamment car je dois gérer ma vie à coté."

BIOGRAPHIE

Thomas est le fils unique de Jeanne. Il est très proche d'elle, il gère seul la situation depuis le diagnostic de l'Alzheimer. Il est pris entre ses études exigeantes, sa vie sociale, et il s'inquiète beaucoup pour sa mère. Il lui rend visite tous les jours après les cours et gère ses courses, ses rendez-vous médicaux et ses papiers. La charge mentale est énorme et il se sent souvent dépassé et coupable de ne pas être là plus de temps. Bien sûr, il aimerait être aidé afin d'avoir du temps pour lui aussi.

OBJECTIFS

- Réunir les informations avec les autres aidants
- Avoir un peu plus d'aide pour l'aider
- Ne pas s'inquiéter pour sa mère, être serein afin d'étudier tranquillement

FRUSTRATIONS

- L'inquiétude : Ne pas savoir ce qui se passe entre ses visites. (Ex : "A-t-elle pris ses pilules ?")
- Les fausses alertes : S'inquiéter parce qu'elle ne répond pas au téléphone, alors qu'elle sieste
- La coordination : Communiquer les mêmes informations plusieurs fois à différentes personnes (infirmière, voisine...)

AISANCE TECHNOLOGIQUE

Téléphone

Very high

5.4.3 Laurie

LAURIE	
PROFIL Genre : Femme Age : 35 ans Profession : Professeur des écoles Situation : Vit avec son mari et sa fille de 7 ans, elle est la voisine de Jeanne	BIOGRAPHIE <p>Laurie connaît Jeanne depuis son arrivée dans le quartier il y a 5 ans. Elle l'apprécie beaucoup de part sa bienveillance et voit les effets de la maladie progresser. Elle est de bonne volonté, mais sa vie de famille et son travail lui prennent déjà beaucoup de temps. Elle veut aider sans devenir une aidante à temps plein.</p>
 “ "Je suis souvent à la maison et ça me fait plaisir d'aider, mais je ne sais pas toujours quand il faut s'inquiéter ou si je dérange."	OBJECTIFS Consignes claires de Thomas sur ce qu'il a besoin Un planning simplifié (si quelqu'un doit passer) Un moyen rapide et simple de signaler un problème ou une observation sans devoir déranger Thomas
	AISANCE TECHNOLOGIQUE Téléphone 
	FRUSTRATIONS La responsabilité : Avoir peur de rater un signe important alors qu'elle était juste à côté Le manque d'info : Ne pas savoir si Jeanne est là ou si Thomas l'a emmenée en rendez-vous L'incertitude : Hésiter à prévenir Thomas par peur de le déranger au travail pour une fausse alerte

5.4 Faisabilité et Veille

5.4.1 Faisabilité Technique

Base de données (SQL)

SQLite n'est pas adapté ici car il ne gère pas efficacement la concurrence et les accès simultanés par plusieurs utilisateurs. Une base de données relationnelle de type **PostgreSQL** (avec **Supabase**) est envisagée pour sa robustesse, sa scalabilité et sa capacité à gérer des données fortement structurées.

Elle permettra notamment de stocker de manière fiable :

- les **profils utilisateurs** (aidants, seniors, familles),
- les **besoins médico-sociaux**, dont la **grille AGGIR simplifiée**,
- les **disponibilités, compétences, antécédents**,
- l'historique des **interventions, tâches et interactions**.

Critères pour la base de données :

- **Sécurité** : support natif du chiffrement et des permissions fines
- **Compatibilité** : facile à intégrer avec Node.js, FastAPI ou Symfony.
- **Apprentissage facilité** : PostgreSQL a déjà été étudié en cours → temps d'apprentissage null

Backend (API REST / API sécurisée)

Le backend expose une **API REST sécurisée** qui centralise l'ensemble de la logique métier de la plateforme. Il prend en charge :

- l'**authentification** et la gestion des permissions,

- le **gestion des profils** et des besoins (GIR, dépendances),
- l'**appariement** aidant ⇔ senior selon compétences et disponibilités,
- la gestion des **notifications**,
- la **communication sécurisée** avec la base PostgreSQL

Critères pour le backend :

- **Communauté et documentation** : large documentation et tutoriels.
- **Sécurité** : support JWT, OAuth2, contrôle fin des rôles et permissions.
- **Phase d'apprentissage facilitée** :
 - **Node.js + Express** : déjà étudié partiellement → **temps d'apprentissage modéré**.
 - **FastAPI** : syntaxe simple → **temps d'apprentissage modéré**.
 - **Symfony (PHP)** : complet donc long à prendre en main → **temps d'apprentissage important** (même si vue en partie en cours)

Fonctions clés gérées : authentification, gestion des profils et besoins, appariement aidant ⇔ senior, notifications, communication sécurisée avec PostgreSQL.

Frontend (Web et Mobile)

Critères pour le frontend :

- **Ergonomie et accessibilité** : interface intuitive pour seniors et aidants.
- **Réactivité et performance** : temps de chargement faible (React, Next.js, Vue.js).
- **Modularité et réutilisabilité** : composants modulaires pour faciliter la maintenance.
- **Compatibilité multi-plateforme** : responsive design, adaptabilité mobile et web.
- **Phase d'apprentissage** :
 - **JavaScript** : déjà étudié. (en partie) → temps d'apprentissage **négligeable**.
 - **React / Next.js / Vue.js** : syntaxe plus avancée → temps d'apprentissage **modéré**.
 - **PHP** (stack classique) : déjà étudié . → temps d'apprentissage **négligeable**.

Le frontend constitue le principal point de contact pour les aidants, les seniors et les familles.

- l'inscription et la gestion de profil,
- la consultation des disponibilités,
- un filtrage avancé (compétences, distance, niveau AGGIR),
- le suivi des interventions,
- un module de messagerie ou de chat sécurisé.

Deux approches technologiques sont possibles :

- **Stack classique** : HTML / CSS / PHP.
- **Stack moderne** : Next.js, React, Angular ou Vue.js.

Notifications (Push / Email)

Critères de choix :

- **Fiabilité** : les messages doivent toujours être envoyés (Firebase, SendGrid, Mailjet).
- **Rapidité** : notification instantanée pour les urgences ou messages.
- **Simplicité d'intégration** : compatible avec les frameworks choisis.
- **Phase d'apprentissage** : Firebase / SendGrid / Mailjet → temps d'apprentissage **négligeable**.

La plateforme doit être capable d'informer rapidement les utilisateurs :

- nouveaux messages,
- demandes d'aide,
- rappels d'intervention,
- suivi des tâches.

Des services spécialisés comme **Firebase Cloud Messaging**, **SendGrid** ou **Mailjet** assurent une diffusion fiable et scalable.

Sécurité des données

La plateforme traitant des **données sensibles**, incluant :

- les niveaux de dépendance (GIR),
- des besoins médicaux,
- des informations sociales confidentielles,

Les données traitées étant sensibles, la sécurité constitue un impératif majeur.

Elle repose sur :

- **Confidentialité** : chiffrement HTTPS et stockage sécurisé.
- **Anonymisation et minimisation** : limiter les données personnelles stockées.
- **Authentification sécurisée** (JWT, OAuth2),
- **Gestion stricte des rôles** (aidant, senior, famille, administration),
- **Conformité RGPD** : consentement, droit à l'oubli, consentement explicite, traçabilité.
- protection contre les attaques courantes (XSS, CSRF, injections SQL). (en fonction du temps restant)

La faisabilité est bonne grâce à des technologies solides, mais la sécurité dépend fortement de la coordination autour du membre spécialisé réseau : **Yanis Debbal** (parcours B).

5.4.2 Faisabilité opérationnelle

La faisabilité opérationnelle dépend principalement de la capacité à gérer et structurer la communauté d'utilisateurs.

- **Points Forts :**
 - **Centralisation** : agenda partagé et mutualisation des besoins médicaux facilitant la coordination.
 - **Accessibilité** : la grille AGGIR simplifiée réduit la complexité et facilite l'entrée dans le service.
- **Risques et Défis Majeurs :**
 - **Effet de réseau** : nécessité d'attirer simultanément aidants et familles (zone géographique pilote recommandée).
 - **Confiance et sécurité** : nécessité de procédures de vérification (identité, antécédents, charte).
 - **Modération** : besoin futur d'une équipe pour gérer litiges et comportements abusifs.

5.4.3 Faisabilité légale et réglementaire

Ce volet présente des obstacles significatifs qui doivent être traités avant tout développement.

- **Responsabilité :**
 - Que se passe-t-il en cas d'accident, de négligence, de vol ou d'abus de la part d'un bénévole ?
 - En cas d'accident ou de négligence, les limites de responsabilité doivent être clairement définies.
 - La plateforme doit se positionner comme **intermédiaire de mise en relation**, via des **CGU précises**.
- **Protection des Données (RGPD) :**
 - Le traitement de données de santé (GIR, traitements) est strictement encadré. Un DPO (Délégué à la Protection des Données) sera nécessaire.

- Le consentement explicite pour la collecte et le partage (même partiel) de ces informations aux bénévoles est obligatoire.

- **Droit du Travail :**
 - La relation doit rester dans un cadre de bénévolat, sans rémunération ni lien de subordination, afin d'éviter toute requalification en travail dissimulé.
 - Le projet est faisable sous réserve de respecter strictement le cadre RGPD et d'établir des CGU solides définissant clairement les responsabilités.

5.4.4 Veille concurrentielle

Afin de concevoir une application pertinente, simple et adaptée aux besoins réels des utilisateurs, nous avons appuyé notre démarche sur plusieurs sources de données et références.

Ces supports ont permis d'ancrer notre projet dans un contexte social concret et de mieux comprendre les pratiques existantes en matière d'entraide et de coordination des aidants.

Nous avons analysé plusieurs plateformes déjà existantes afin d'identifier leurs points forts et leurs limites :

- **Voisins Solidaires**, qui favorise l'entraide de proximité entre habitants, mais sans réelle gestion d'emploi du temps ni suivi personnalisé.
- **Doctolib**, pour son interface fluide et sa gestion efficace des créneaux et notifications.
- **Mon Espace Santé**, qui met l'accent sur la sécurité et la confidentialité des données personnelles, aspects essentiels pour notre projet.
Ces comparaisons nous ont permis de définir une application combinant **simplicité d'usage, coordination des tâches et sécurité des échanges**.
- **Gitlab (Gricad)**, l'envoi d'un problème ("issue") déclenche un courriel à l'ensemble des professeurs concernés. L'un d'eux prend en charge le problème et contacte l'équipe appropriée.

5.5 UserStory et EPIC

5.5.1 Gérer les utilisateurs, l'authentification et la création de cercles de soutien pour les personnes âgées

5.5.1.1 Autoriser les nouveaux utilisateurs à s'inscrire via le site web.

En tant que nouvel utilisateur,
Je souhaite créer un compte sur le site web,
Afin de pouvoir accéder aux services en ligne de Weave.

Critères d'acceptation :

1. **Étant donné que je suis sur la page d'inscription**
Quand je saisiss tous les champs obligatoires avec des informations valides
Alors mon compte est créé et je suis redirigé vers l'écran de confirmation.
2. **Étant donné que le formulaire est incomplet ou invalide**
Quand je clique sur « Créer un compte »
Alors je vois des erreurs de validation expliquant ce qui doit être corrigé.
3. **Étant donné que j'ai déjà un compte**
Quand je tente de m'enregistrer avec la même adresse e-mail
Alors le système affiche un message d'erreur.

5.5.1.2 Autoriser les utilisateurs à se connecter via un navigateur.

En tant qu'utilisateur,
Je souhaite me connecter via l'interface web,
Afin de pouvoir accéder à mon tableau de bord en toute sécurité.

Critères d'acceptation :

1. **Étant donné que j'ai un compte**
Quand j'entre des identifiants valides
Alors je suis connecté et redirigé vers mon tableau de bord.
2. **Étant donné que j'entre des identifiants invalides**
Quand je clique sur « Se connecter »
Alors je reçois une erreur « Adresse e-mail ou mot de passe incorrect ».
3. **Étant donné que je suis connecté**
Quand j'actualise la page
Alors ma session reste active.

5.5.1.3 Créez un groupe de soutien autour d'une personne âgée via l'application web.

En tant que proche aidant principal,
Je souhaite créer un cercle de soins depuis le site web,
Afin que les aidants puissent collaborer en ligne.

Critères d'acceptation :

1. **Étant donné que je suis connecté**
Quand j'ouvre « Créer un cercle de soins » et que je remplis les champs obligatoires
Alors un nouveau cercle de soins est créé et je suis défini comme Aidant Principal.
2. **Étant donné que des champs obligatoires sont manquants**
Quand je tente de soumettre
Alors des erreurs de validation apparaissent.

5.5.1.4 Inviter des assistants via un lien Web

En tant que proche aidant principal,
Je souhaite envoyer des liens d'invitation directement depuis l'application web,
Afin que d'autres puissent rejoindre facilement.

Critères d'acceptation :

1. **Étant donné que je suis l'Aidant Principal**
Quand je clique sur « Générer un lien d'invitation »
Alors un lien/code unique est créé.
2. **Étant donné qu'un aidant ouvre le lien d'invitation**
Quand il accepte l'invitation
Alors il est ajouté à la liste des membres du cercle de soins.

5.5.1.5 Gérez les rôles au sein du groupe via le web.

En tant que proche aidant principal,
Je souhaite attribuer des rôles via l'interface web,
Afin que les droits d'accès soient correctement gérés.

Critères d'acceptation :

1. **Étant donné que je suis l'Aidant Principal**
Quand j'attribue un rôle à un membre
Alors ses permissions changent immédiatement.
2. **Étant donné que je modifie le rôle de quelqu'un**
Quand j'enregistre la mise à jour
Alors le système confirme la modification.

5.5.2 Fournir un calendrier hebdomadaire collaboratif pour la planification des tâches et la coordination des bénévoles.

5.5.2.1 Ajouter des tâches via le calendrier web.

En tant qu'aidant,
Je souhaite créer une tâche via le calendrier web,
Afin que les aidants puissent voir les besoins à venir.

Critères d'acceptation :

1. **Étant donné que je suis dans un cercle de soins**
Quand je soumets une tâche avec tous les champs obligatoires
Alors elle apparaît dans le calendrier partagé.
2. **Étant donné que des champs obligatoires sont manquants**
Quand je tente d'enregistrer
Alors un message d'erreur apparaît.

5.5.2.2 Mettre à jour les tâches via le site web.

En tant qu'aidant,
Je souhaite modifier ou supprimer des tâches directement depuis l'interface web,
Afin que le planning reste exact.

Critères d'acceptation :

1. **Étant donné que je suis autorisé**
Quand je modifie une tâche
Alors les changements apparaissent instantanément pour tous les membres.
2. **Étant donné que je supprime une tâche**
Quand je confirme la suppression
Alors la tâche est retirée du calendrier.
3. **Étant donné qu'un aidant est assigné**
Quand je modifie ou supprime une tâche
Alors il reçoit une notification.

5.5.2.3 Les personnes qui proposent leurs services s'inscrivent en ligne pour effectuer les tâches.

En tant qu'aidant,
Je souhaite me porter volontaire pour des tâches via le site web,
Afin de pouvoir offrir mon soutien facilement.

Critères d'acceptation :

1. **Étant donné qu'une tâche a des créneaux disponibles**
Quand je clique sur « Me porter volontaire »
Alors je suis ajouté comme aidant.
2. **Étant donné que la tâche est complète**
Quand j'ouvre la tâche
Alors l'option « Me porter volontaire » est masquée ou désactivée.

5.5.2.4 Verrouillage automatique des tâches une fois le quota d'assistant atteint.

En tant que système,
Je souhaite verrouiller automatiquement les inscriptions en ligne lorsque le quota est atteint,
Afin qu'aucun volontaire supplémentaire ne puisse rejoindre.

Critères d'acceptation :

1. **Étant donné que le nombre requis d'aidants est atteint**
Quand un autre utilisateur consulte la tâche
Alors l'inscription est bloquée.
2. **Étant donné qu'un aidant se désiste**
Quand le nombre repasse en dessous du quota
Alors l'inscription est automatiquement réactivée.

5.5.2.5 Rappel dans le navigateur ou par courriel pour confirmation.

En tant qu'aidant,
Je souhaite confirmer ma présence via le système de rappel web,
Afin que le groupe sache que je suis toujours disponible.

Critères d'acceptation :

1. **Étant donné qu'une tâche approche**
Quand le rappel se déclenche
Alors les aidants assignés reçoivent une demande de confirmation.
2. **Étant donné que je confirme**
Quand je clique sur le bouton de confirmation
Alors mon statut devient « Confirmé ».
3. **Étant donné que je ne confirme pas**
Quand le délai expire
Alors je suis retiré et le créneau se libère.

5.5.3 Permettre une communication privée et sécurisée entre les assistants.

5.5.3.1 Messages de groupe dans le navigateur.

Critères d'acceptation :

1. **Étant donné que je fais partie d'un cercle de soins**
Quand j'ouvre le chat
Alors je vois tous les messages précédents.
2. **Étant donné que j'envoie un message**
Quand je clique sur « Envoyer »
Alors il apparaît instantanément pour tous les membres.
3. **Étant donné que je ne fais pas partie du groupe**
Quand j'accède au chat
Alors l'accès est refusé.

5.5.3.2 Conversation liée aux tâches.

En tant qu'aidant,
Je souhaite discuter de tâches spécifiques dans un chat web,
Afin que la communication reste pertinente et organisée.

Critères d'acceptation :

1. **Étant donné que je suis assigné à une tâche**
Quand j'ouvre le chat de la tâche
Alors je peux consulter et publier des messages.
2. **Étant donné que je ne suis pas assigné**
Quand j'essaie d'ouvrir le chat de la tâche
Alors je ne peux pas y accéder.

5.5.3.3 Notifications de messages sur le Web.

En tant qu'utilisateur,
Je souhaite recevoir des notifications par navigateur ou par e-mail pour les nouveaux messages du chat,
Afin de rester informé même en dehors de l'écran de discussion.

Critères d'acceptation :

1. **Étant donné qu'un nouveau message est publié**
Quand je ne suis pas sur la page du chat
Alors je reçois une notification visible.
2. **Étant donné que j'ai désactivé les notifications de messages**
Quand de nouveaux messages arrivent
Alors je ne reçois PAS d'alertes.

5.5.4 Fournir un journal pour que la personne âgée puisse suivre son humeur, ses activités et partager ses souvenirs.

5.5.4.1 Ajouter des notes/photos en ligne.

En tant qu'aidant,

Je souhaite télécharger des notes et des photos depuis le site web,
Afin que le cercle de soins puisse conserver des souvenirs.

Critères d'acceptation :

1. **Étant donné que j'ai terminé une tâche**
Quand j'ouvre « Ajouter une entrée de journal »
Alors je peux écrire du texte et télécharger des photos.
2. **Étant donné que j'enregistre l'entrée**
Quand je la soumets
Alors elle apparaît dans la chronologie du journal.

5.5.4.2 Sélectionnez votre niveau d'humeur en ligne.

En tant qu'aidant,

Je souhaite sélectionner un indicateur d'humeur via le journal web,
Afin que les membres de la famille puissent suivre les tendances émotionnelles.

Critères d'acceptation :

1. **Étant donné que je crée une entrée de journal**
Quand je sélectionne une humeur
Alors elle est enregistrée avec l'entrée.
2. **Étant donné que je consulte l'entrée plus tard**
Quand je l'ouvre
Alors l'humeur sélectionnée est affichée.

5.5.4.3 Afficher les entrées dans le navigateur.

En tant que membre de la famille,

Je souhaite consulter les entrées du journal dans une chronologie web,
Afin de suivre facilement l'évolution de la personne âgée.

Critères d'acceptation :

1. **Étant donné que des entrées de journal existent**
Quand j'ouvre le journal
Alors les entrées apparaissent dans l'ordre chronologique.
2. **Étant donné que certaines entrées contiennent des photos**
Quand je consulte la chronologie
Alors des vignettes des photos apparaissent.

5.5.4.4 Télécharger le journal au format PDF/HTML.

En tant que proche aidant principal,

Je souhaite exporter le journal depuis le site web,
Afin de pouvoir le partager ou l'archiver.

Critères d'acceptation :

1. **Étant donné que des entrées existent**
Quand je clique sur « Exporter le journal »
Alors un fichier téléchargeable est généré.
2. **Étant donné que l'exportation est terminée**
Quand j'ouvre le fichier
Alors toutes les entrées apparaissent avec les dates et les photos.

5.5.5 Gérer les incidents, urgents et non urgents, lors des visites.

5.5.5.1 Rapport d'incident en ligne.

En tant que **bénévole**,
Je veux **signaler un incident via un formulaire web**,
Afin que les aidants reçoivent des alertes immédiates.

Critères d'acceptation :

1. **Étant donné que je clique sur « Signaler un incident »**
Quand le système demande « **S'agit-il d'une urgence vitale ?** »
Alors je dois choisir **oui/non**.
2. **Étant donné que je choisis « Oui »**
Quand je continue
Alors le système m'indique d'**appeler les services d'urgence**.
3. **Étant donné que je choisis « Non »**
Quand je remplis le formulaire
Alors un **incident non vital** est soumis.

5.5.5.2 Alerte automatique sur PC.

En tant que **système**,
Je veux **envoyer des alertes au principal aidant via des notifications web ou par email**,
Afin qu'il puisse agir rapidement.

Critères d'acceptation :

1. **Étant donné qu'un incident est soumis**
Quand il est enregistré
Alors le principal aidant reçoit une **notification**.
2. **Étant donné que l'aidant ouvre le tableau de bord**
Quand un incident est ouvert
Alors il apparaît dans sa **liste des incidents ouverts**.

5.5.5.3 Faites remonter l'information si pas de réponse en ligne.

En tant que **système**,
Je veux **escalader les incidents à un contact de confiance si l'aidant principal ne répond pas en ligne**,
Afin qu'aucune alerte ne soit ignorée.

Critères d'acceptation :

1. **Étant donné qu'un incident est ouvert**
Quand l'aide principal ne répond pas dans les **X minutes**
Alors le contact de confiance reçoit une **notification**.
2. **Étant donné que** le contact de confiance répond
Quand il accuse réception
Alors l'incident lui est **assigné**.

5.5.5.4 Résoudre l'incident via le navigateur.

En tant que **aidant**,
Je veux marquer les incidents comme résolus dans le tableau de bord web,
Afin que les bénévoles sachent que le problème est pris en charge.

Critères d'acceptation :

1. **Étant donné que** je suis aidant
Quand je clique sur « **Résoudre l'incident** »
Alors l'incident est **marqué comme résolu**.
2. **Étant donné qu'un** incident est **résolu**
Quand le bénévole actualise
Alors le statut mis à jour apparaît.

5.5.6 Fournir des notifications push et par e-mail pour les tâches, les messages et les incidents.

5.5.6.1 Configurer les préférences de notification en ligne.

En tant qu'**utilisateur**,
Je veux configurer les notifications dans les paramètres web,
Afin que je ne reçoive que ce dont j'ai besoin.

Critères d'acceptation :

1. **Étant donné que** je suis sur la **page des paramètres**
Quand je modifie les **types de notification**
Alors le système **enregistre mes préférences**.
2. **Étant donné que** je désactive une **catégorie**
Quand de **nouveaux événements** surviennent
Alors je ne reçois pas d'**alertes pour cette catégorie** (sauf pour les incidents).

5.5.6.2 Alertes concernant les tâches nouvelles ou mises à jour.

En tant que **bénévole**,
Je veux recevoir des notifications via le site web ou par email,
Afin que je reste informé des tâches.

Critères d'acceptation :

Étant donné qu'une nouvelle tâche est créée
Quand je fais partie du **cercle de soins**
Alors je reçois une **notification**.
Étant donné qu'une tâche change
Quand je suis **assigné à celle-ci**
Alors je reçois une **notification de mise à jour**.

5.5.6.3 Les notifications d'incidents ne peuvent pas être désactivées.

En tant que **système**,

Je veux **envoyer des alertes d'incident obligatoires via le tableau de bord web et par email**,
Afin que les alertes de sécurité soient toujours communiquées.

Critères d'acceptation :

Étant donné qu'un incident est signalé

Quand je suis aidant

Alors je reçois toujours la notification.

Étant donné que j'ai désactivé d'autres catégories

Quand un incident survient

Alors je reçois toujours les alertes d'incident.

5.5.7 Protéger les données médicales et personnelles sensibles grâce à des règles de sécurité strictes

5.5.7.1 Limiter les données en fonction du rôle sur le web.

En tant que **système**,

Je veux **appliquer un contrôle d'accès basé sur les rôles dans l'application web**,
Afin que les données sensibles soient protégées.

Critères d'acceptation :

1. **Étant donné que j'ai un certain rôle**
Quand j'accède à une page protégée
Alors l'accès est accordé ou refusé en conséquence.

2. **Étant donné que je ne suis pas autorisé**
Quand j'essaie de consulter des données sensibles
Alors je reçois un message d'accès refusé.

5.5.7.2 Sécuriser les données sensibles en transit et au repos.

En tant que **système**,

Je veux **chiffrer toutes les données sensibles transmises ou stockées via le web**,
Afin que la confidentialité soit maintenue.

Critères d'acceptation :

1. **Étant donné que j'utilise le site**
Quand je charge une page
Alors la connexion utilise HTTPS.
2. **Étant donné que des données sensibles sont stockées**
Quand elles sont écrites dans la base de données
Alors elles sont chiffrées.

5.5.7.3 Recueillir le consentement pour les données sensibles via le web.

En tant que **personne âgée**,
Je veux donner mon **consentement via une interface web**,
Afin que je contrôle qui voit mes informations sensibles.

Critères d'acceptation :

1. **Étant donné que je soumets des informations sensibles**
Quand le consentement est requis
Alors le système affiche une demande de consentement.
2. **Étant donné que je retire mon consentement**
Quand je mets à jour mes paramètres de confidentialité
Alors le système masque ou supprime les données sensibles.

5.5.7.4 Suivre l'accès aux données.

En tant que **système**,
Je veux **journaliser tous les accès aux données sensibles sur le web**,
Afin que les accès non autorisés puissent être détectés.

Critères d'acceptation :

Étant donné qu'un utilisateur accède à des données sensibles
Quand l'accès se produit
Alors une entrée de journal d'audit est générée.
Étant donné qu'un administrateur consulte les journaux
Quand il accède au tableau de bord d'audit
Alors il peut filtrer les journaux par utilisateur, date ou événement.

5.5.8 Créer une interface web claire et accessible

5.5.8.1 Créer l'interface utilisateur principale pour le web.

En tant qu'**utilisateur**,
Je veux utiliser une **interface web claire et intuitive**,
Afin que je puisse naviguer facilement sur Weave.

Critères d'acceptation :

1. **Étant donné que je charge l'application**
Quand la page s'affiche
Alors la navigation est cohérente sur toutes les pages.
2. **Étant donné que je clique sur un élément de navigation**
Quand la route se charge
Alors la page correcte apparaît.

5.5.8.2 Page d'accueil simple et accessible.

En tant que **personne âgée**,
Je veux une **page d'accueil claire et minimaliste sur le web**,
Afin que je puisse comprendre rapidement les informations clés.

Critères d'acceptation :

1. **Étant donné que je suis une personne âgée**
Quand j'ouvre la **page d'accueil**
Alors je vois des **grands boutons et des informations simplifiées.**
2. **Étant donné que je consulte les visites à venir**
Quand je regarde le **panneau d'accueil**
Alors je peux **voir clairement le prochain événement planifié.**

5.5.8.3 Gros boutons, polices lisibles, contraste.

En tant qu'utilisateur avec une expérience numérique limitée,
Je veux des contrôles accessibles sur l'interface web,
Afin que je puisse utiliser la plateforme confortablement.

Critères d'acceptation :

1. **Étant donné que j'utilise le site web**
Quand je navigue
Alors le **texte respecte les standards de contraste minimum WCAG.**
2. **Étant donné que j'utilise des outils d'assistance**
Quand je navigue
Alors tous les **éléments cliquables sont accessibles au clavier.**

5.5.9 Modélisation

5.5.9.1 Diagramme de cas d'utilisation

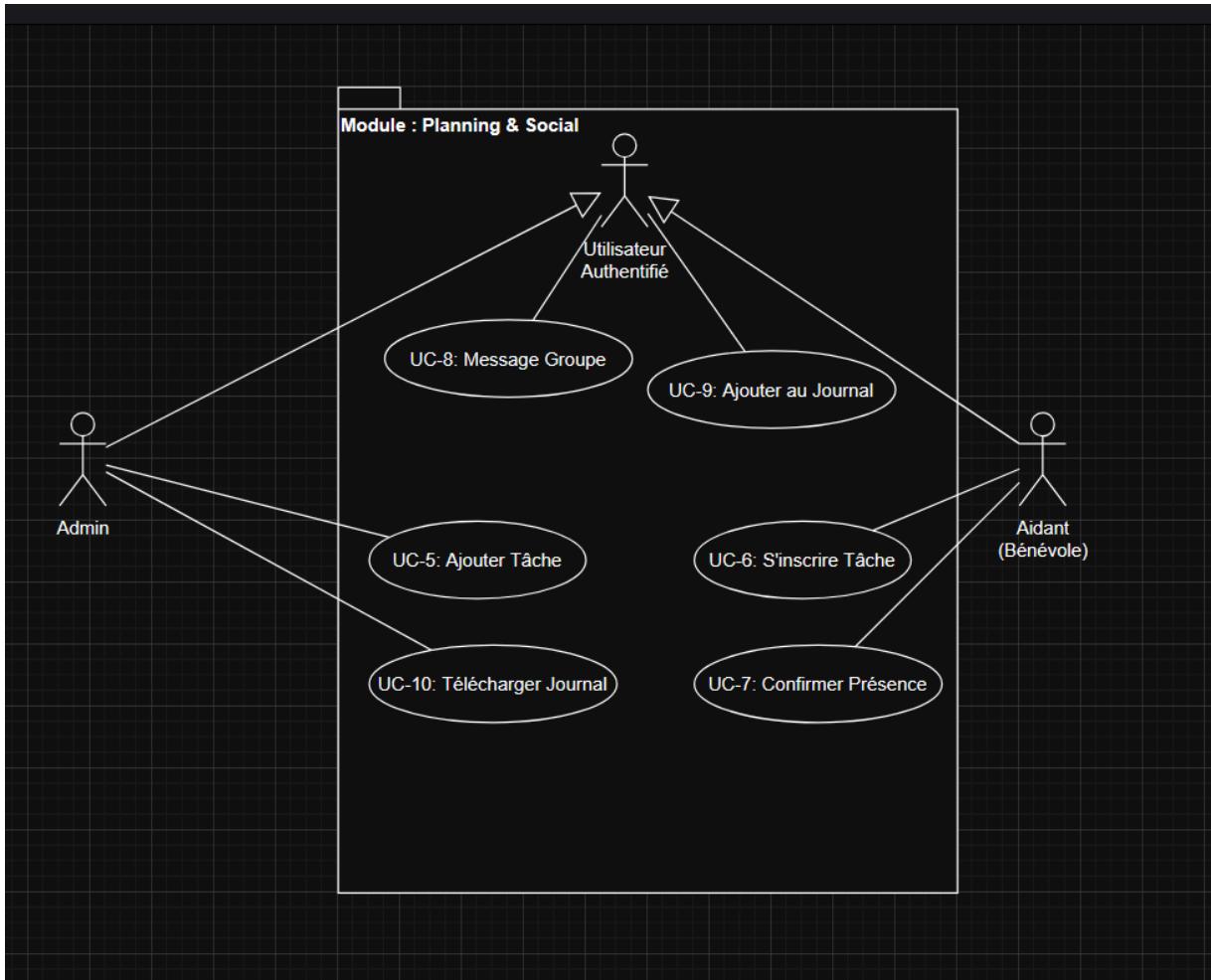
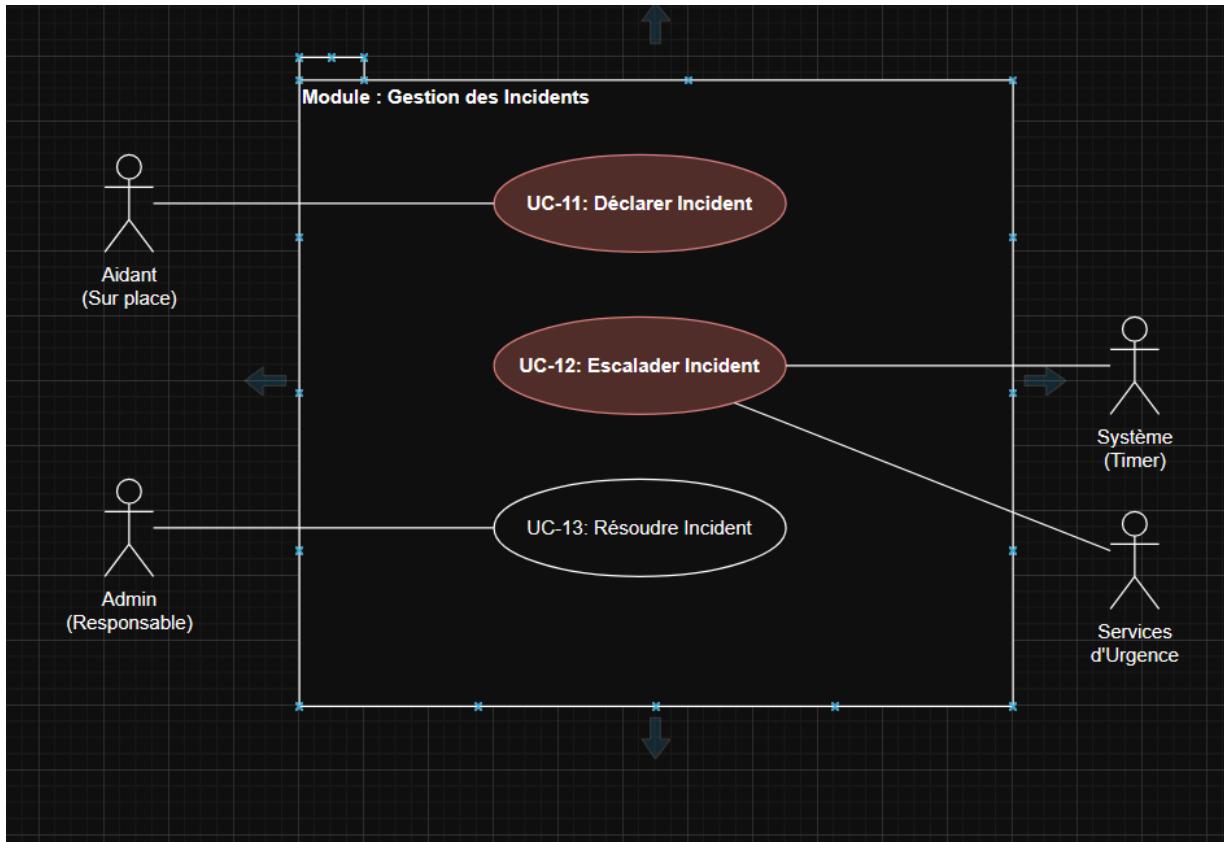


Diagramme : Planning & Social

Ce diagramme représente la vie quotidienne de l'application Weave. Pour simplifier la lecture et éviter les répétitions, nous avons regroupé les fonctionnalités d'échange (Messagerie et Journal) sous un acteur générique : l'Utilisateur Authentifié. Cela signifie que dès qu'une personne est connectée, elle peut participer à la vie sociale du groupe.

Les responsabilités opérationnelles sont ensuite réparties selon les rôles. D'un côté, l'Administrateur agit comme le planificateur du groupe : c'est lui qui alimente le calendrier en créant les besoins. De l'autre, les Bénévoles (Aidants) consultent ce planning pour s'inscrire aux missions et confirmer leur présence, assurant ainsi une coordination fluide sans nécessiter d'appels téléphoniques incessants.



Use Case : Gestion des incidents

Ce dernier module est isolé car il traite de la fonction la plus critique : la sécurité de la personne aidée. Sa particularité est d'introduire un acteur numérique autonome, le Système (Timer), qui agit comme une sécurité passive.

Le processus est conçu comme une chaîne de responsabilité ininterrompue. Tout commence par une détection humaine : l'Aidant sur place signale une anomalie via l'application. L'Administrateur est alors notifié pour résoudre le problème. Cependant, si aucune réaction humaine n'est détectée dans le délai imparti, le Système prend automatiquement le relais pour escalader l'alerte vers les Services d'Urgence. Ce mécanisme garantit qu'une situation de danger ne restera jamais sans réponse, même en cas d'indisponibilité de l'entourage.

5.5.9.2 Diagramme de séquence

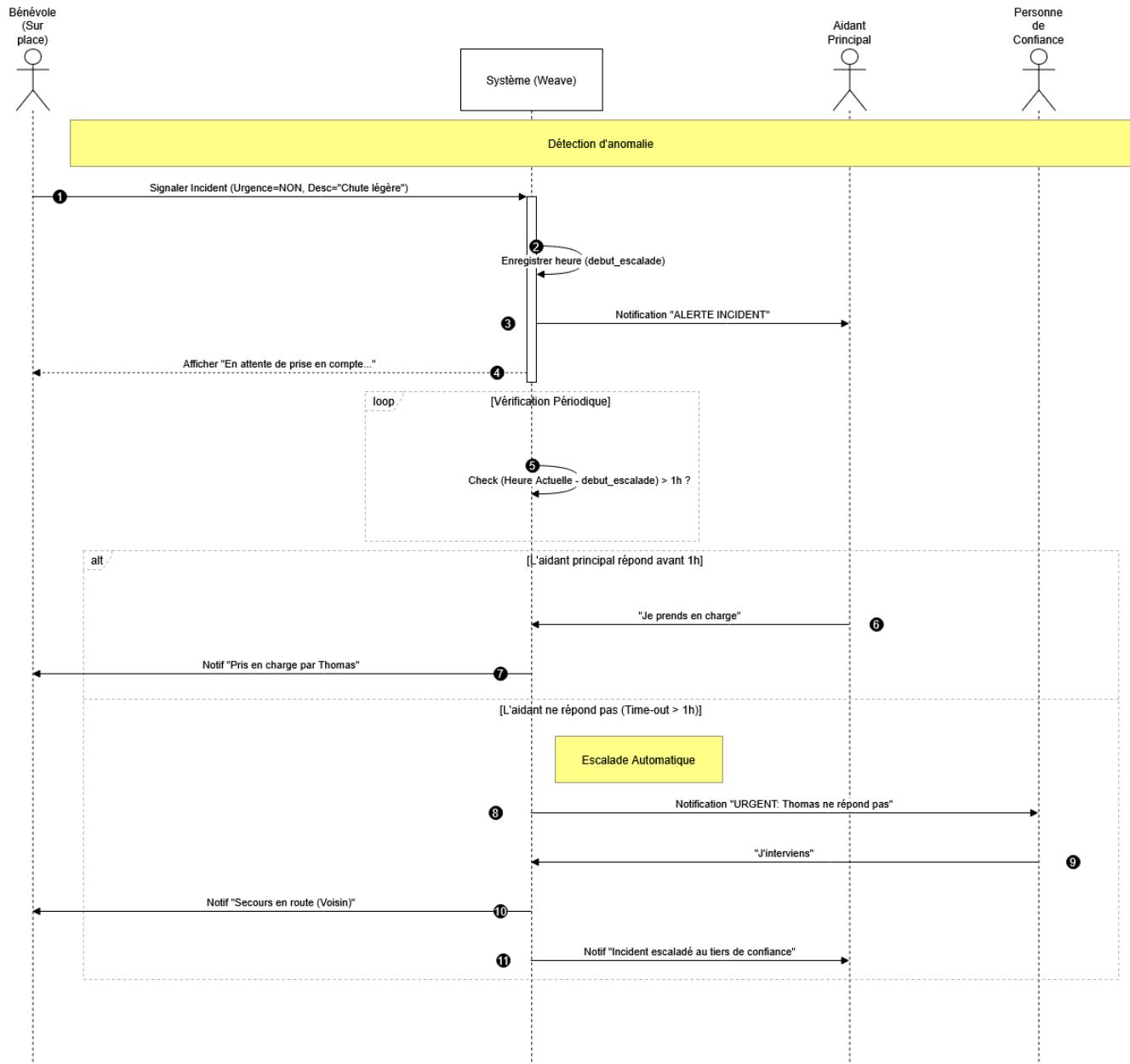


Diagramme de séquence : Gestion d'incident

Ce diagramme illustre la résilience de notre système. Nous avons modélisé une **surveillance active** : le serveur possède un automatisme (Cron Job) qui mesure le temps de réponse. Si l'humain responsable fait défaut (pas de réponse sous 1h), l'application prend la décision autonome d'escalader l'alerte vers un tiers de confiance. C'est ce mécanisme qui garantit la sécurité continue du bénéficiaire

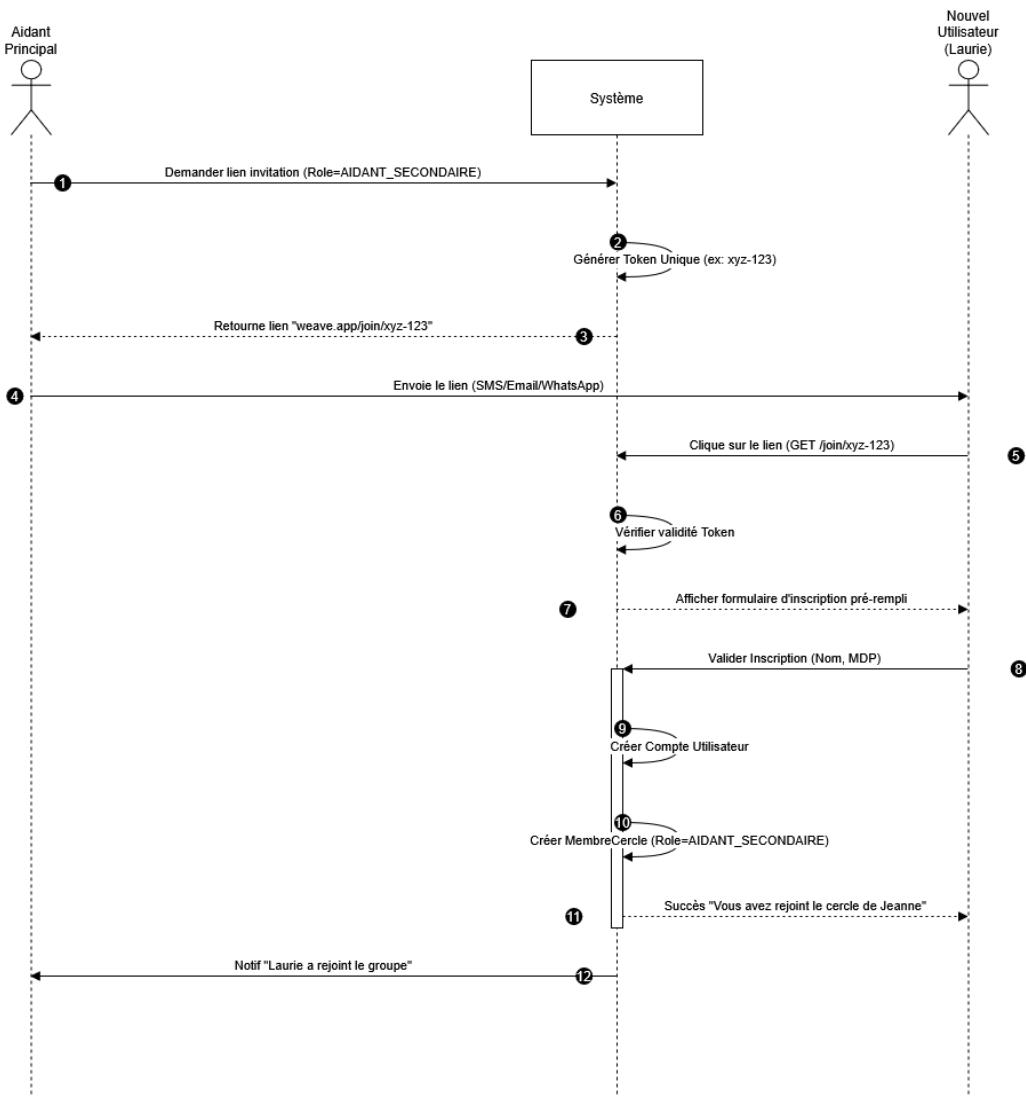


Diagramme de séquence : Invitation dans un cercle d'aideant.

Ce diagramme montre notre procédure d'enrôlement ("Onboarding"). Nous avons banni l'inscription libre pour des raisons de confidentialité. L'accès repose sur un système d'**invitation par token**. Cela permet à l'Admin de pré-définir le rôle du futur membre (Bénévole, Famille). Ainsi, quand le nouvel utilisateur clique sur le lien, il est automatiquement affecté au bon cercle avec les bons droits d'accès, sans aucune configuration complexe de sa part.