



RÉVÉLATEUR D'INGÉNIEURS  
DEPUIS 1961



# CoHub

---

*Une solution de partage et d'entraide à l'ENIB*

---

<b>CHEVALIER Anouk</b>	a5cheval@enib.fr
<b>DELABRE Jarod</b>	j5delabr@enib.fr
<b>EVEN Elouan</b>	e5even@enib.fr
<b>LEVEQUE Dorian</b>	d5levequ@enib.fr

26 Octobre 2018

# Sommaire

I- Quel est le problème ? .....	5
II- Notre solution .....	5
III-Notre environnement de travail .....	10
IV-Nos diagrammes UML.....	10
IV-1 Diagrammes de cas d'utilisation .....	11
IV-2 Scénario.....	12
IV-3 Diagrammes de séquences .....	16
IV-4 Diagrammes de classe .....	23
IV-5 Diagrammes d'états-transitions.....	24
IV-6 Diagrammes d'activité .....	25
V- Conclusion .....	28
VI- Annexes.....	29
VI.1- Diagramme de Gantt.....	29
VI.2- Présentation de l'équipe .....	29
VI.3- Pourquoi CoHub ? .....	30

---

Enibiennes, Enibiens,

Pas de tente pour le weekend d'intégration ? Qui a une visseuse pour installer une étagère ? Qui a une imprimante ? Besoin d'aide en maths la veille d'un DS ? Qui rentre à Quimper ce weekend ?

Toutes ces situations racontent un quotidien des é nibien-ne-s.

Parfois les réponses peuvent être si nombreuses que refuser devient un casse-tête. Mais d'autres fois des questions restent sans solution.

C'est la répétition de ces tracas tout au long de notre formation qui nous a donné l'idée de [CoHub](#) : une application internet qui permet facilement et efficacement à tout enibien-ne de partager des biens et des services.

Disponible sur n'importe quel appareil disposant d'un navigateur internet, cette application vous permettra de demander de l'aide ou de la proposer.

Avec [CoHub](#), fini les lacunes et les problèmes. [CoHub](#) est là pour vous !

[L'équipe CoHub](#)

---

Enibiennes, Enibiens,

No tent before the WAF? Who has a screwdriver to install this shelf? Who has a printer? Need help in math the day before an interrogation? Who goes back to Quimper this weekend?

All of those situations are a common occurrence for an Enibien.

Sometimes those answers can be so numerous that refusing one can be a baffling problem.

But sometimes the questions have no answer.

It's the repetition of this hassle all along our formation that gave us the idea of [CoHub](#): a website that allows any Enibien to easily and efficiently share goods or services.

Available on any device connected to the internet, this application will allow you to ask for help or to offer your help.

With [CoHub](#), no more shortcomings and issues. [CoHub](#) is here for you!

[Team CoHub](#)

---

Collaboration

Partage

Entraide



# CoHub

Echange

Réseaux

Coopération

# I- Quel est le problème ?

Alors que les réseaux sociaux sont l'objet de critiques, des réseaux d'échanges et de partage se diversifient et se spécialisent dans de nombreux domaines, du simple partage de fichiers à des applications ou des sites spécifiques de prêts et d'échanges entre particuliers.

C'est une pratique qui progresse et qui s'accompagne d'une image positive.

Des applications mobiles ou des sites internet existent déjà mais sont organisés autour de besoins spécifiques. Comment partager localement, à l'échelle de l'ENIB, à la fois des biens, un coup de main, des compétences ?

Il s'agit par exemple de pouvoir éviter des achats individuels difficiles à rentabiliser ou utilisables seulement de manière occasionnelle ce qui peut permettre une maîtrise de la consommation, des économies et rend service aussi à la planète !

## II- Notre solution

Notre solution est le site internet **CoHub**. C'est un site de partage et d'échange fait par des étudiants de l'ENIB pour répondre aux demandes spécifiques des étudiants de l'ENIB. Le site a pour ambition de faciliter le partage, la collaboration et l'entraide.

**CoHub** recense donc toutes les demandes d'aide car nous pensons qu'elles ont plus de chances d'aboutir ainsi. Pour stimuler les interactions, il doit être accessible à tous les étudiants de l'ENIB. **CoHub** doit aussi être disponible à partir de n'importe quelle interface utilisateur (téléphone, tablette ou ordinateur) à tout moment et en tout lieu où une connexion internet est possible. Pour plus d'efficacité, toutes les demandes sont classées par catégorie afin de faciliter des réponses expertes.

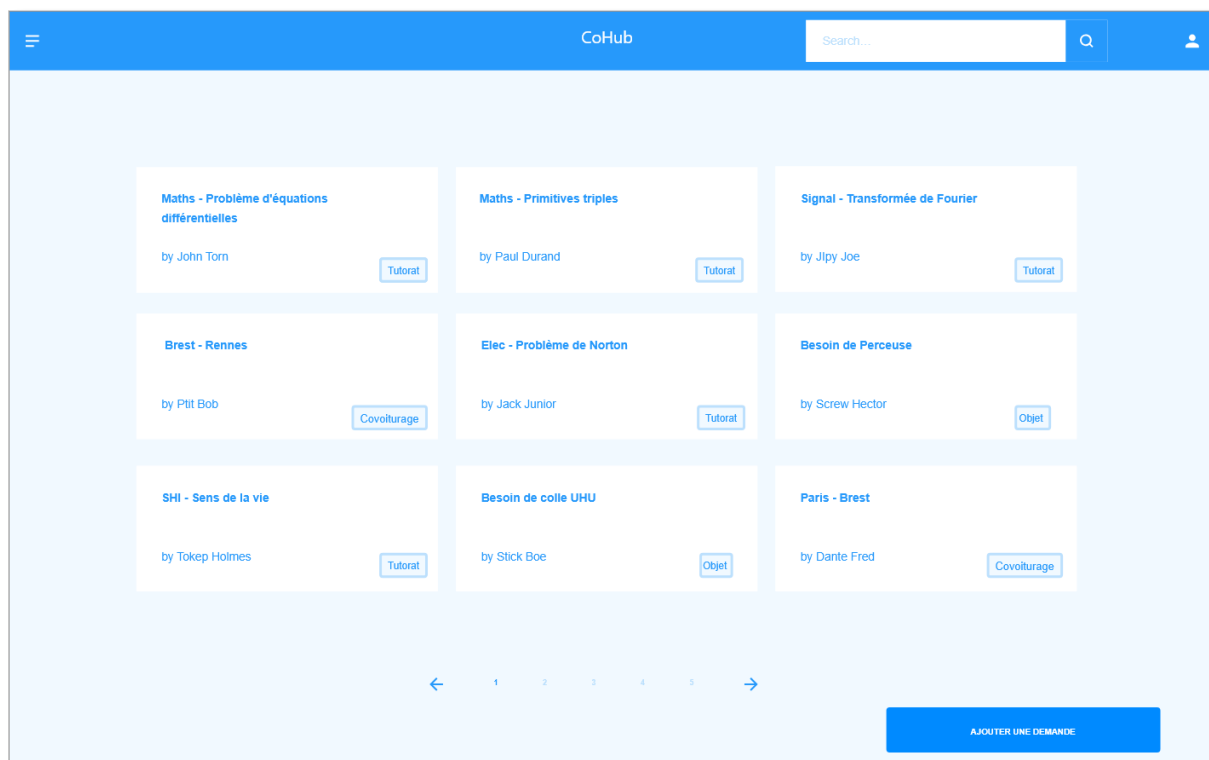
La consultation des demandes émises n'est pas restreinte, par contre dans un souci de sécurité, l'émission ou la réponse à une demande est contrôlée par un identifiant et un mot de passe.

Afin de répondre à ces différents besoins, le site **CoHub** collecte les demandes d'aide, les range par catégorie et les rend disponibles à la consultation.

Les trois catégories prédéfinies sont tutorat, covoiturage et prêt d'objets. Il serait très facile d'implémenter des catégories supplémentaires.

Le site **CoHub** doit permettre sans authentification de consulter toutes les demandes en cours et aussi de consulter une demande particulière afin d'avoir plus de détails sur celle-ci. La consultation de demandes peut être filtrée par catégorie ou par mot-clé.

### Page d'accueil de **CoHub** où l'on peut consulter l'ensemble des demandes :



La liste affiche les informations de base de la demande, c'est à dire, le titre, la catégorie et le demandeur. Il suffit de sélectionner une demande afin d'afficher toutes les informations relatives à celles-ci.

## Exemple de demande :

**DEMANDE**

TITRE  
Maths - Problème d'équation différentiel

Demandeur  
John Tron / Jippy

Catégorie  
TUTORAT

Sujet  
SHI

Professeur  
Gaubert

Theme  
Equations différentielles

Semestre  
S2

**DESCRIPTION**

Bonjour,  
J'essaye de comprendre les équations différentielles.  
Je me suis un peu renseigné et il me semble qu'il y ait plusieurs types d'équations différentielles. Donc dans un premier temps il me semble important de comprendre si la forme d'une équation différentielle est propre à un système (physique, chimique, quantique) ou si sa forme est générale.  
Merci

**CONVERSATION**

Type something...

→

ANNULER CETTE PERSONNE

Exemple de demande

Une demande se caractérise par une liste de champs. Chaque catégorie a des champs spécifiques. Les champs importants sont la description, le titre et la conversation.

## Sélection de catégorie :



Un simple bandeau sur la page d'accueil permet de filtrer les résultats en fonction d'une catégorie de ticket.

## Fenêtre d'authentification :

Connexion

Name  
John Doe

Password  
\*\*\*\*\*

CONNEXION

Sorry, wrong password

Fenêtre d'authentification

L'utilisateur rentre ses identifiants de connexion.

Après l'authentification, il est possible de créer une nouvelle demande ou de répondre à une demande via un système de messages entre le demandeur et l'aidant.

Un demandeur peut aussi d'éditer sa demande déjà existante après authentification.

## Page de création ou d'édition d'une demande :

DEMANDE

TITRE  
Entrez un titre

Catégorie  
TUTORAT

Sujet  
SHI

Professeur  
Entrez le nom d'un professeur

Theme  
Entrez un theme

Semestre  
s2

DESCRIPTION  
Type something...

VALIDER

Fenêtre de création ou d'édition de demande



Ici l'utilisateur renseigne les champs afin que sa demande puisse être prise en compte et comprise par les autres utilisateurs. Lorsque tous les champs sont renseignés l'appuis sur valider enregistre la demande.

### Page permettant à l'aidant potentiel de dialoguer avec le demandeur :

The screenshot displays the 'TICKETO' interface for creating a request. The form is titled 'DEMANDE' and includes fields for 'TITRE', 'Demandeur', 'Catégorie', 'Sujet', 'Professeur', 'Theme', and 'Semestre'. A 'DESCRIPTION' section contains a text area with a sample message about differential equations. Below this is a 'CONVERSATION' section showing a list of messages between 'John Tron / Jippy' and 'Joe Rodriguez / Papy'. At the bottom, there is a text input field for the user to type a message, a 'Valider Aide' button, and a 'Type something...' placeholder.

**Fenêtre consultation de la demande.**

**L'aidant peut dialoguer avec le demandeur**

Cette interface permet à l'utilisateur souhaitant aider de communiquer avec le demandeur.

# III-Notre environnement de travail

## GitLab :

Afin de développer notre application dans les meilleures conditions, nous avons opté pour l'utilisation d'un dépôt . L'école nous propose via nos comptes Enibien la possibilité de pouvoir créer un dépôt (un emplacement sur un serveur) afin d'y stocker et de partager entre nous les différents fichiers de notre projet.

**GIT**



## PlantUML :



PlantUML, un modelleur UML adapté à nos besoins. Il s'agit d'un moyen alternatif à Modelio afin de générer nos différents diagrammes UML.

Ce choix n'est pas dû au hasard. En effet, pour avancer dans ce projet, comme nous utilisons Gitlab, il est ainsi plus facile pour toute l'équipe de consulter et de mettre à jour nos différents diagrammes lors de notre phase de conception sans empiéter sur le travail des autres.

De plus, il repose sur un langage simple open source constamment mis-à-jour.

# IV-Nos diagrammes UML

L'acronyme UML signifie Unified Modeling Language.

C'est un langage normé de modélisation graphique d'un système utilisé pour concevoir et documenter un système indépendamment d'un langage de programmation.

L'UML permet de décrire le système d'un point de vue statique et dynamique.



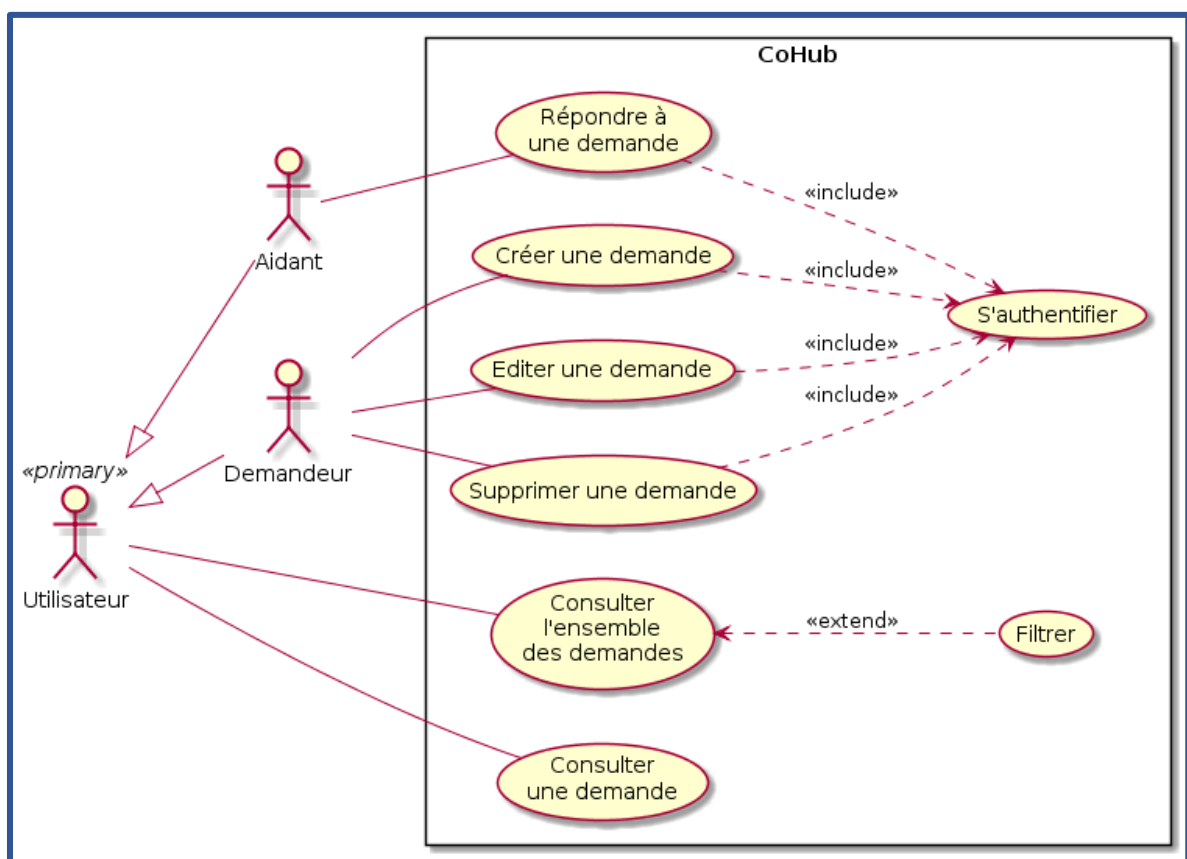
# IV-1 Diagrammes de cas d'utilisation

Après une analyse fonctionnelle, l'élaboration du diagramme de cas d'utilisation est la première étape UML d'analyse d'un système.

Ce diagramme recense et synthétise les grandes fonctionnalités du système du point de vue de l'utilisateur.

Il permet aussi d'identifier les éléments extérieurs qui interagissent avec le système.

Voici le diagramme de cas d'utilisation de [CoHub](#) :



Lorsque l'utilisateur se connecte au site [CoHub](#), il peut consulter l'ensemble des demandes d'aide émises affichées sur la page d'accueil.

Il peut aussi filtrer les demandes c'est-à-dire effectuer une recherche par mot-clé ou par catégorie. Il lui est aussi possible de consulter une demande en particulier : afficher plus de détails la concernant.

L'utilisateur s'authentifie s'il souhaite répondre à une demande ou bien poster une demande.

Après l'authentification, il a la possibilité de créer une nouvelle demande ou d'éditer ses demandes pour les modifier ou les supprimer.

Lorsque l'aidant souhaite répondre à une demande il peut discuter par message de la solution avec le demandeur.

## IV-2 Scénario

La deuxième étape lors de la création d'un système consiste à décrire textuellement chaque cas d'utilisation. C'est ce qu'on appelle un scénario. Dans chaque scénario on décrit chronologiquement l'enchaînement des étapes qui constituent le cas d'utilisation grâce à un scénario nominal, des scénarios alternatifs et des scénarios d'échec.

En ce qui concerne [CoHub](#), nous avons réalisé 8 scénarios. Nous avons regroupé certaines extensions optionnelles dans des scénarios alternatifs.

### IV.2.1- Consulter l'ensemble des demandes

<b>Sommaire</b>	<b>Scénario nominal "Consulter l'ensemble des demandes"</b>
Description	1. L'utilisateur accède à la page 2. Le système récupère l'ensemble des demandes en cours auprès de la base de données 3. Le système affiche une miniature de chaque demande en cours
<b>Sommaire</b>	<b>Scénario alternatif "Consulter l'ensemble des demandes"</b>
Description	SA1 : L'utilisateur applique un filtre SA1 : Démarre au point 3 du scénario nominal 4. Le système récupère l'ensemble des demandes en cours filtré auprès de la base de données 5. Le système affiche une miniature de chaque demande en cours filtré
<b>Sommaire</b>	<b>Scénario d'erreur "Consulter l'ensemble des demandes"</b>
Description	

## IV.2.2- Consulter une demande

<b>Sommaire</b>	<b>Scénario nominal "Consulter une demande"</b>
Description	1. L'utilisateur accède à la page 2. Le système récupère la demande auprès de la base de données 3. Le système affiche la demande
<b>Sommaire</b>	<b>Scénario alternatif "Consulter une demande"</b>
Description	
<b>Sommaire</b>	<b>Scénario d'erreur "Consulter une demande"</b>
Description	

## IV.2.3- S'authentifier

<b>Sommaire</b>	<b>Scénario nominal "S'authentifier"</b>
Description	1. L'utilisateur accède à la page 2. L'utilisateur saisit son identifiant et son mot de passe 3. Le système vérifie la syntaxe des champs 4. Le système vérifie l'identifiant et le mot de passe à l'aide de la base de données 5. Le système valide la connexion
<b>Sommaire</b>	<b>Scénario alternatif "S'authentifier"</b>
Description	
<b>Sommaire</b>	<b>Scénario d'erreur "S'authentifier"</b>
Description	SE1 : l'identifiant et/ou le mot de passe ont une mauvaise syntaxe SE1 démarre au point 2 du scénario nominal 3. Le système indique à l'utilisateur que son identification a échoué Le scénario nominal reprend au point 1  SE2 : l'identifiant et/ou le mot de passe sont erronés SE2 démarre au point 3 du scénario nominal 3. Le système indique à l'utilisateur que son identification a échoué Le scénario nominal reprend au point 1

## IV.2.4- Créer une demande

<b>Sommaire</b>	<b>Scénario nominal "Créer une demande"</b>
Description	1. Le demandeur saisit les informations relatives à sa demande 2. Le demandeur valide 3. Le système vérifie si l'ensemble des champs obligatoires ont été remplis 4. Le système transmet les informations à la base de données afin d'être enregistrées 5. Le système indique à l'utilisateur que les informations sont enregistrées
<b>Sommaire</b>	<b>Scénario alternatif "Créer une demande"</b>
Description	
<b>Sommaire</b>	<b>Scénario d'erreur "Créer une demande"</b>
Description	SE1 : Les informations ne sont pas correctes SE1 démarre au point 3 du scénario nominal 5. Le système indique à l'utilisateur qu'une partie des informations sont erronées

## IV.2.5- Éditer une demande

<b>Sommaire</b>	<b>Scénario nominal "Éditer une demande"</b>
Description	1. Le système vérifie si c'est l'utilisateur courant qui a créé la demande 2. Le demandeur appuie sur le bouton "édite une demande" 3. Le demandeur modifie les informations 4. Le demandeur valide sa demande 5. Le système vérifie l'ensemble des informations 6. le système enregistre les informations dans la base de données 7. Le système affiche la demande modifiée
<b>Sommaire</b>	<b>Scénario alternatif "Éditer une demande"</b>
Description	
<b>Sommaire</b>	<b>Scénario d'erreur "Éditer une demande"</b>
Description	SE1 : L'utilisateur courant n'est pas le demandeur SE1 départ au point 1 du scénario nominal Le cas d'utilisation se termine en échec (l'objectif n'est pas atteint)  SE2 : Les informations ne sont pas correctes SE2 départ au point 5 du scénario nominal 6. Le système indique à l'utilisateur qu'une partie des informations est erronée Le cas d'utilisation se termine en échec (l'objectif n'est pas atteint)

## IV.2.6- Répondre à une demande

<b>Sommaire</b>	<b>Scénario nominal "Répondre à une demande"</b>
Description	1. L'aidant saisit un message à destination du demandeur 2. L'aidant valide son message en appuyant sur le bouton "proposer une demande" 3. Le système enregistre le message dans la base de données 4. Le système enregistre l'aidant dans la demande
<b>Sommaire</b>	<b>Scénario alternatif "Répondre à une demande"</b>
Description	SA1 : L'aidant est déjà enregistré dans la demande SA1 démarre 3 au point du scénario nominal Le scénario se termine
<b>Sommaire</b>	<b>Scénario d'erreur "Répondre à une demande"</b>
Description	

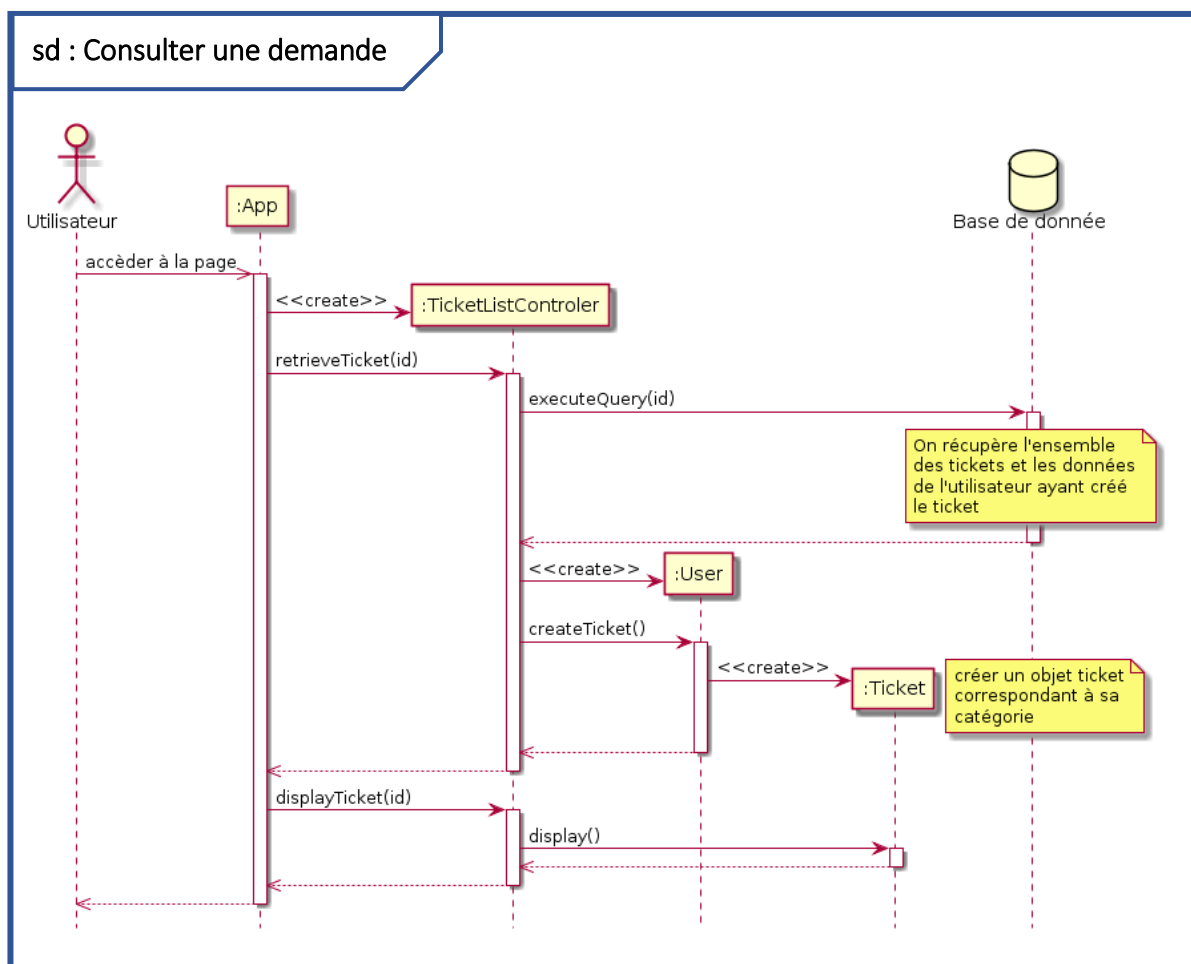
## IV.2.7- Supprimer une demande

<b>Sommaire</b>	<b>Scénario nominal "Supprimer une demande"</b>
Description	1. Le système vérifie si c'est l'utilisateur courant qui a créé la demande 2. Le demandeur appuie sur le bouton "supprimer" 3. le système supprime les informations relatives à la demande dans la base de données 4. Le système affiche la page d'accueil
<b>Sommaire</b>	<b>Scénario alternatif "Supprimer une demande"</b>
Description	
<b>Sommaire</b>	<b>Scénario d'erreur "Supprimer une demande"</b>
Description	

## IV-3 Diagrammes de séquences

La troisième étape dans la création d'un système consiste à formaliser les scénarios. Pour cela, on utilise les diagrammes de séquence qui ont donc pour but de montrer comment les objets qui participent au scénario communiquent pour réaliser les fonctionnalités souhaitées. Un diagramme de séquence correspond à un scénario.

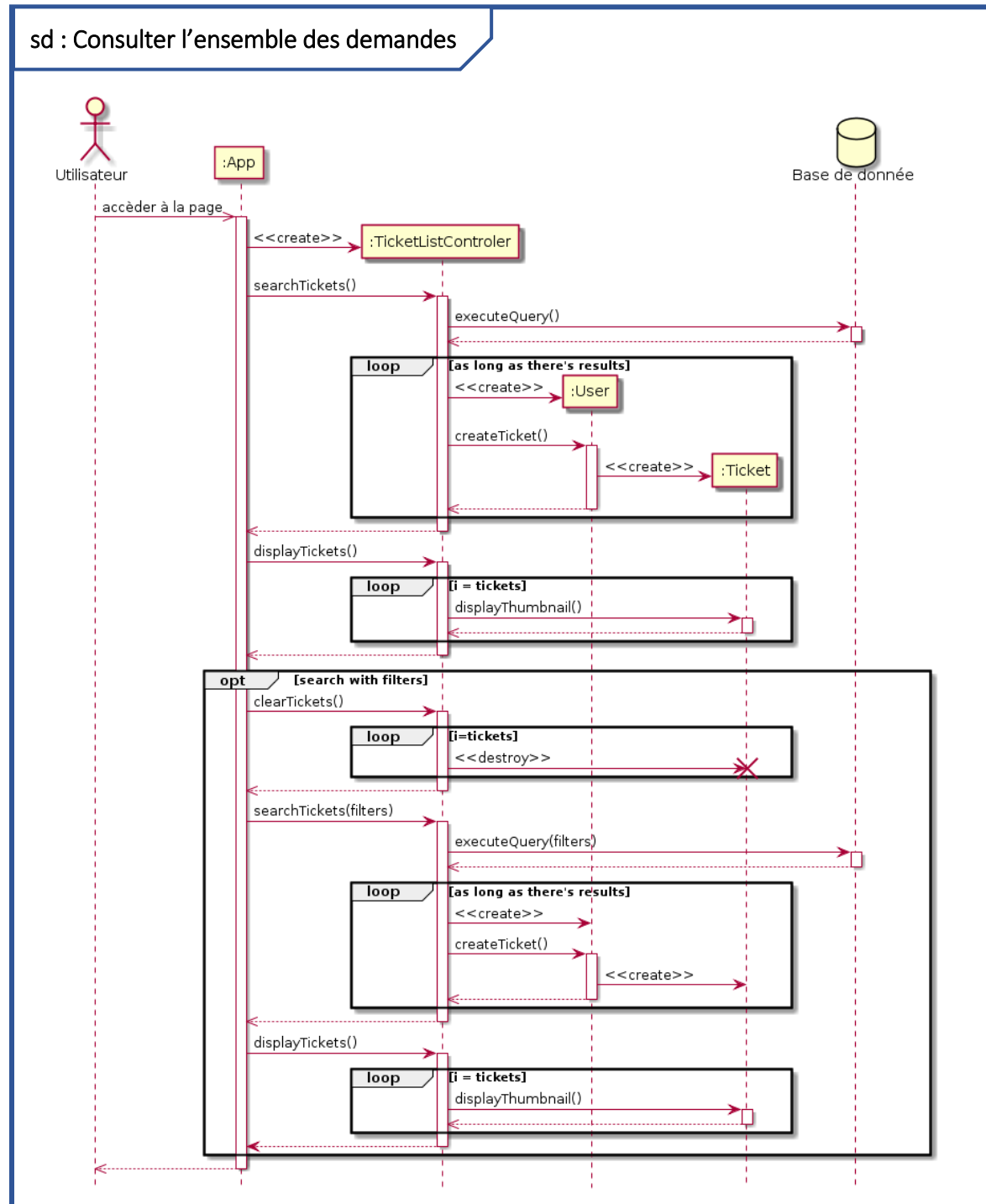
### IV.3.1- Consulter une demande



Ce diagramme présente les interactions entre la base de données et l'utilisateur lorsque celui-ci souhaite consulter une demande en cours. Quand l'utilisateur souhaite accéder à une demande le site récupère l'information auprès de la base de données puis génère une page à afficher sur l'interface utilisateur.



### IV.3.2- Consulter l'ensemble des demandes

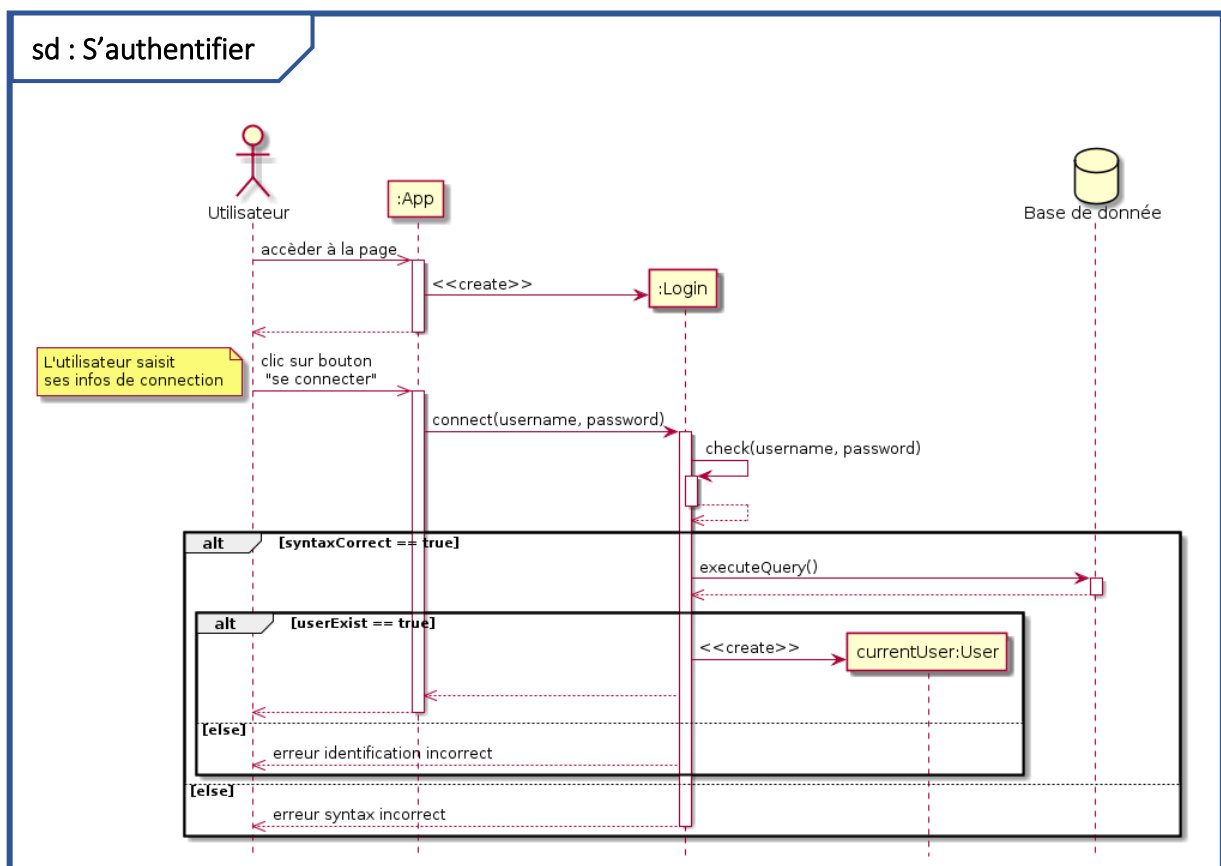


Ce diagramme présente les interactions entre la base de données et l'utilisateur lorsque celui-ci souhaite consulter l'ensemble des demandes en cours. Quand l'utilisateur accède à la

page d'accueil du site, le site récupère l'ensemble des demandes auprès de la base de données puis génère une page à afficher sur l'interface utilisateur.

Lorsque l'utilisateur souhaite filtrer les demandes le site récupère l'ensemble des demandes en cours filtrées auprès de la base de données puis génère une page à afficher sur l'interface utilisateur avec les demandes filtrées.

### IV.3.3- S'authentifier

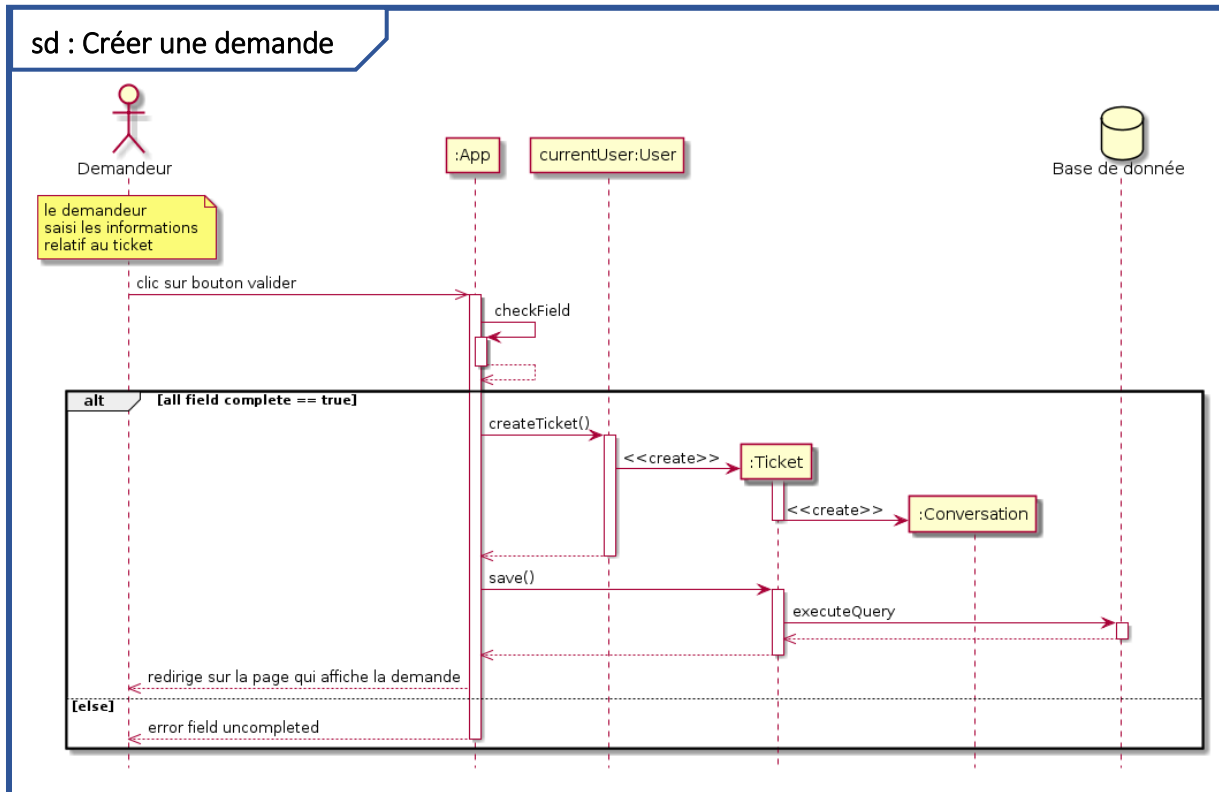


Ce diagramme présente les interactions entre la base de données et l'utilisateur lorsque celui-ci souhaite s'authentifier. Après avoir saisi son identifiant et son mot de passe une première vérification syntaxique est réalisée. Lorsque cette vérification est effectuée avec succès, le système récupère les informations auprès de la base de données.

Par contre si une erreur syntaxique est détectée le système indique à l'utilisateur qu'il y a un problème. Lorsque le système a correctement récupéré l'information auprès de la base de données, il affiche sur l'interface utilisateur que l'authentification a été réalisée avec succès.

Si le système ne trouve pas les informations dans la base de données, il indique une erreur.

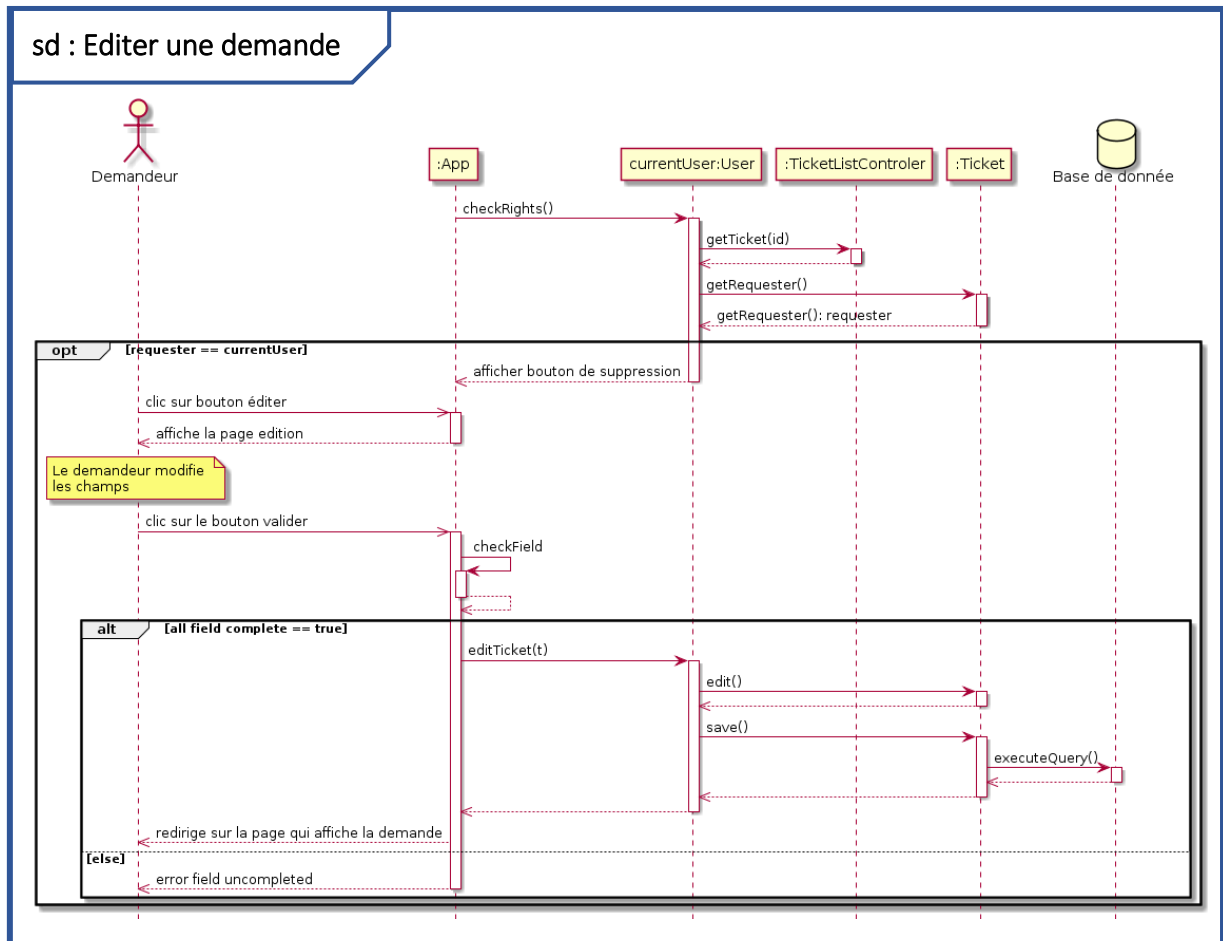
#### IV.3.4- Créer une demande



Ce diagramme présente les interactions entre la base de données et le demandeur lorsque celui-ci souhaite créer une demande d'aide. Après l'appui sur le bouton « création de demande », l'application affiche sur l'interface utilisateur un ensemble de champs à remplir concernant sa demande.

La cohérence des informations est ensuite vérifiée par l'application. Si ces informations sont cohérentes, le système enregistre la demande, sinon il l'indique à l'utilisateur par un message d'erreur.

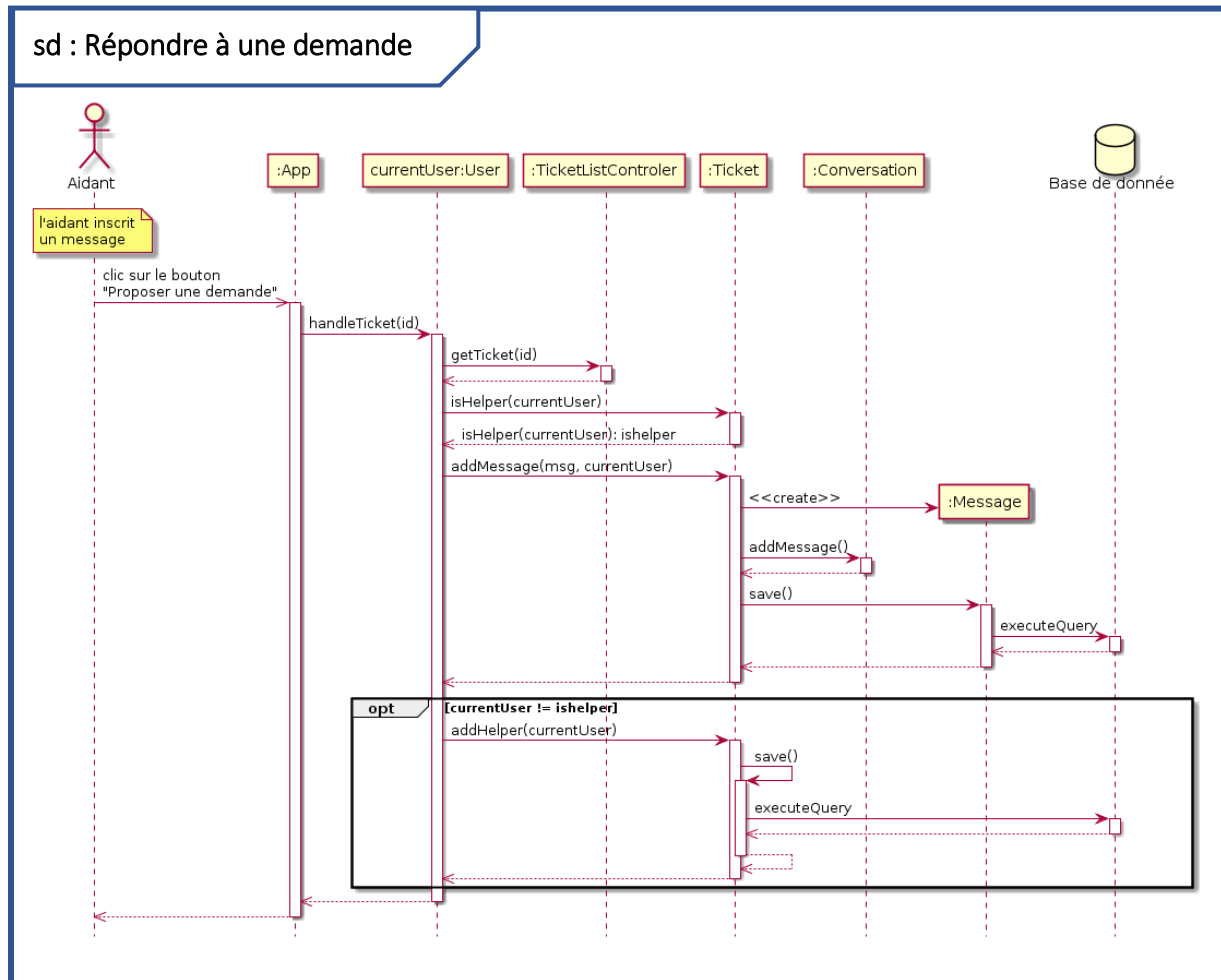
### IV.3.5- Éditer une demande



Ce diagramme présente les interactions entre la base de données et le demandeur lorsque celui-ci souhaite éditer une demande d'aide qu'il a émise. Après l'appui sur le bouton « éditer une demande », l'application affiche sur l'interface utilisateur l'ensemble des champs à remplir lors de la création de la demande.

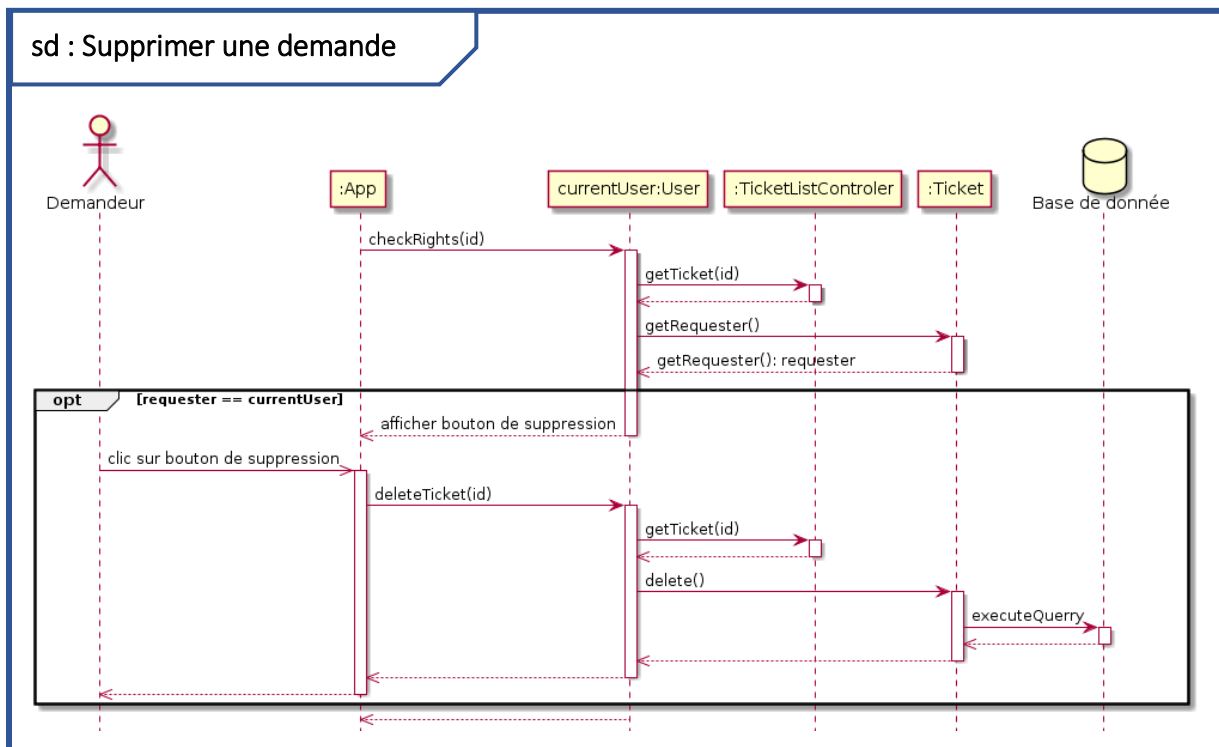
La cohérence de cette information est ensuite vérifiée par l'application. Si ces informations sont cohérentes, le système enregistre la demande dans la base de données, sinon il l'indique à l'utilisateur par un message d'erreur.

#### IV.3.6- Répondre à une demande



Ce diagramme présente les interactions entre la base de données et l'aidant lorsque celui-ci souhaite répondre à une demande. Après la saisie d'un message le système enregistre le message dans la base de données. Il enregistre aussi l'utilisateur en tant qu'aidant dans le ticket si cela n'a pas été fait.

### IV.3.7- Supprimer une demande

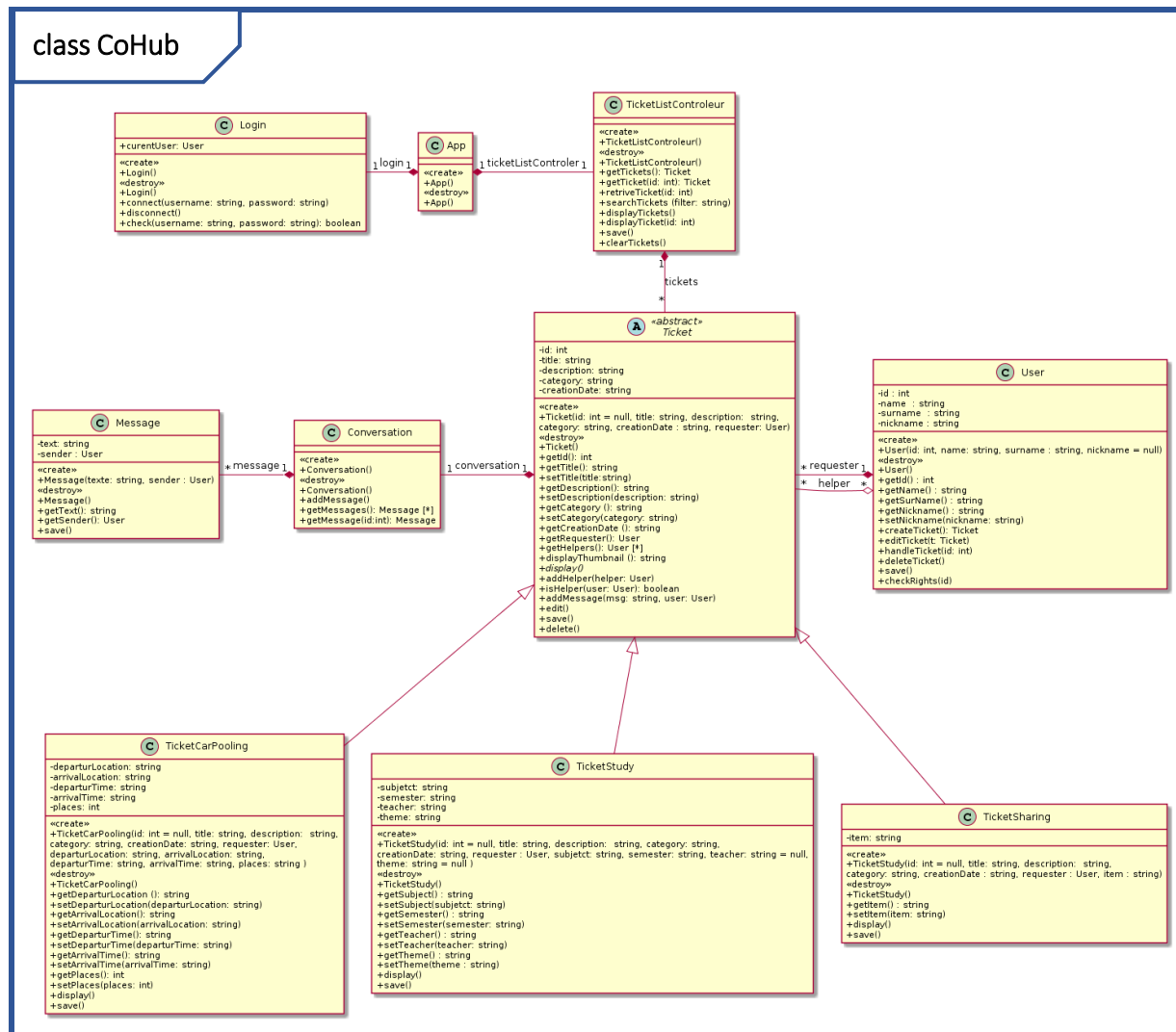


Ce diagramme présente les interactions entre la base de données et le demandeur lorsque celui-ci souhaite supprimer une demande d'aide qu'il a émis. Après l'appui sur le bouton « supprimer », l'application supprime la demande dans la base de données.

## IV-4 Diagrammes de classe

Dans un quatrième temps nous avons réalisé le diagramme de classe de CoHub.

Le diagramme de classe permet de décrire clairement la structure de l'application d'un point de vue statique. Il permet de décrire les classes et les relations entre elles.



Le site CoHub peut être consulté par tous librement. Cependant les actions d'émissions, de réponses, d'édition et de suppression d'une demande nécessitent une connexion.

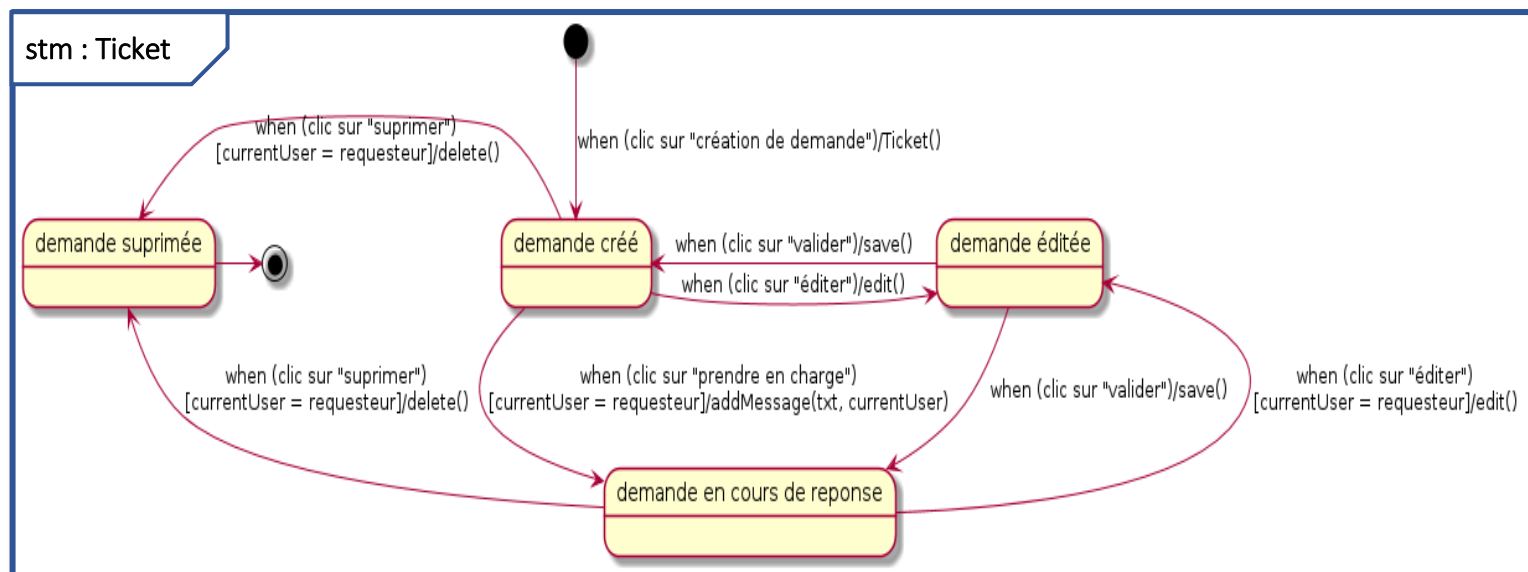
La classe Login contient une méthode permettant de s'authentifier et de se déconnecter à CoHub. CoHub contient une classe abstraite Ticket qui correspond à une demande émise par le « requester » (demandeur) qui est une instance de la classe User.

Les classes TicketCarPooling, TicketStudy et TicketSaring spécialisent la classe Ticket. Un ou plusieurs User peuvent répondre à une demande, on les a appelés les « helper » (aidant). Ils sont aussi une instance de la classe User.

Lorsque qu'un aidant souhaite répondre à une demande d'aide, il envoie un message sur une conversation liée à la demande.

## IV-5 Diagrammes d'états-transitions

Le diagramme d'état-transition permet de modéliser le comportement d'un objet parmi ses différents états. Ici, il s'agit de l'état d'un ticket au cours de l'exécution de l'application.

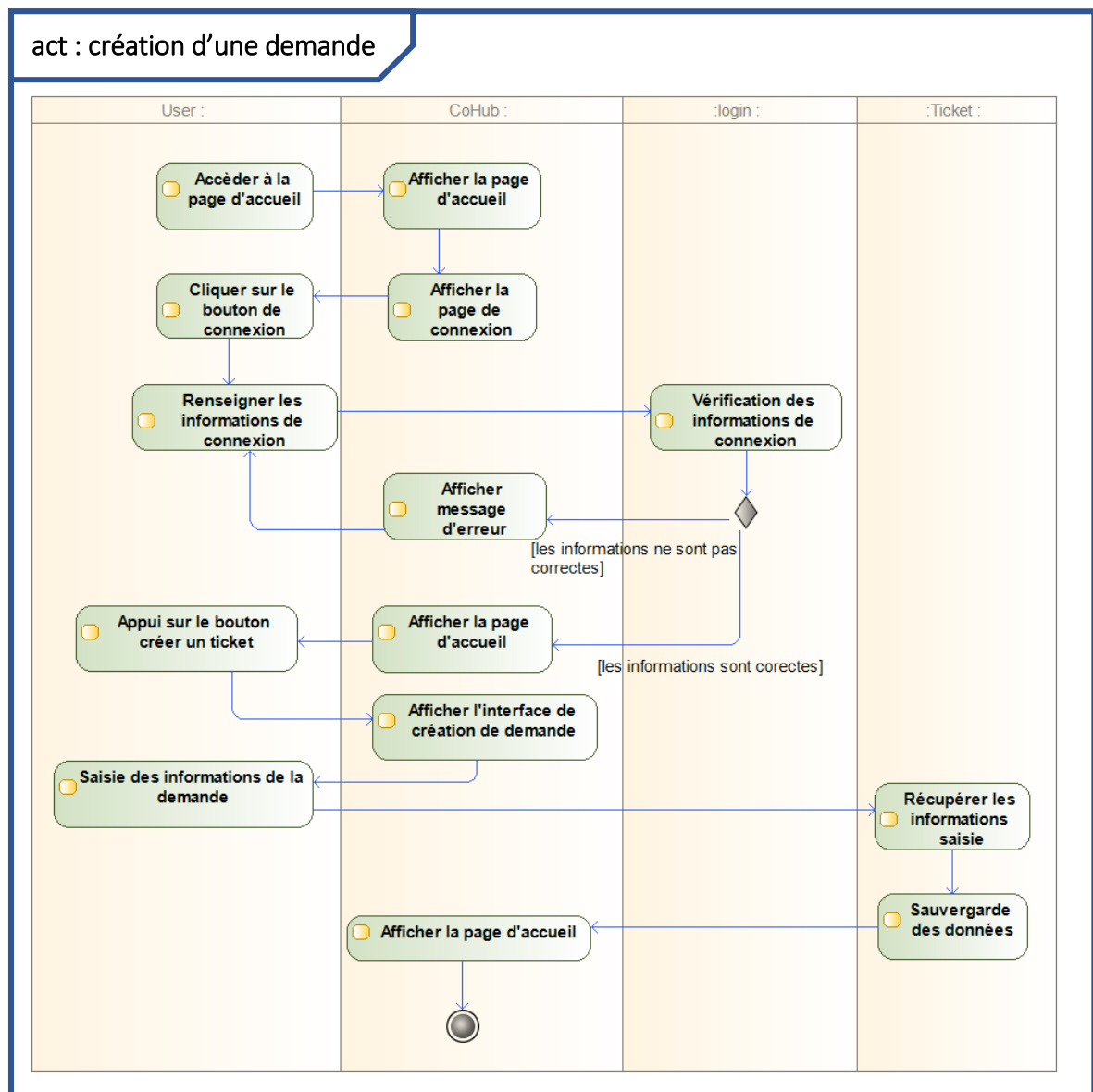




## IV-6 Diagrammes d'activité

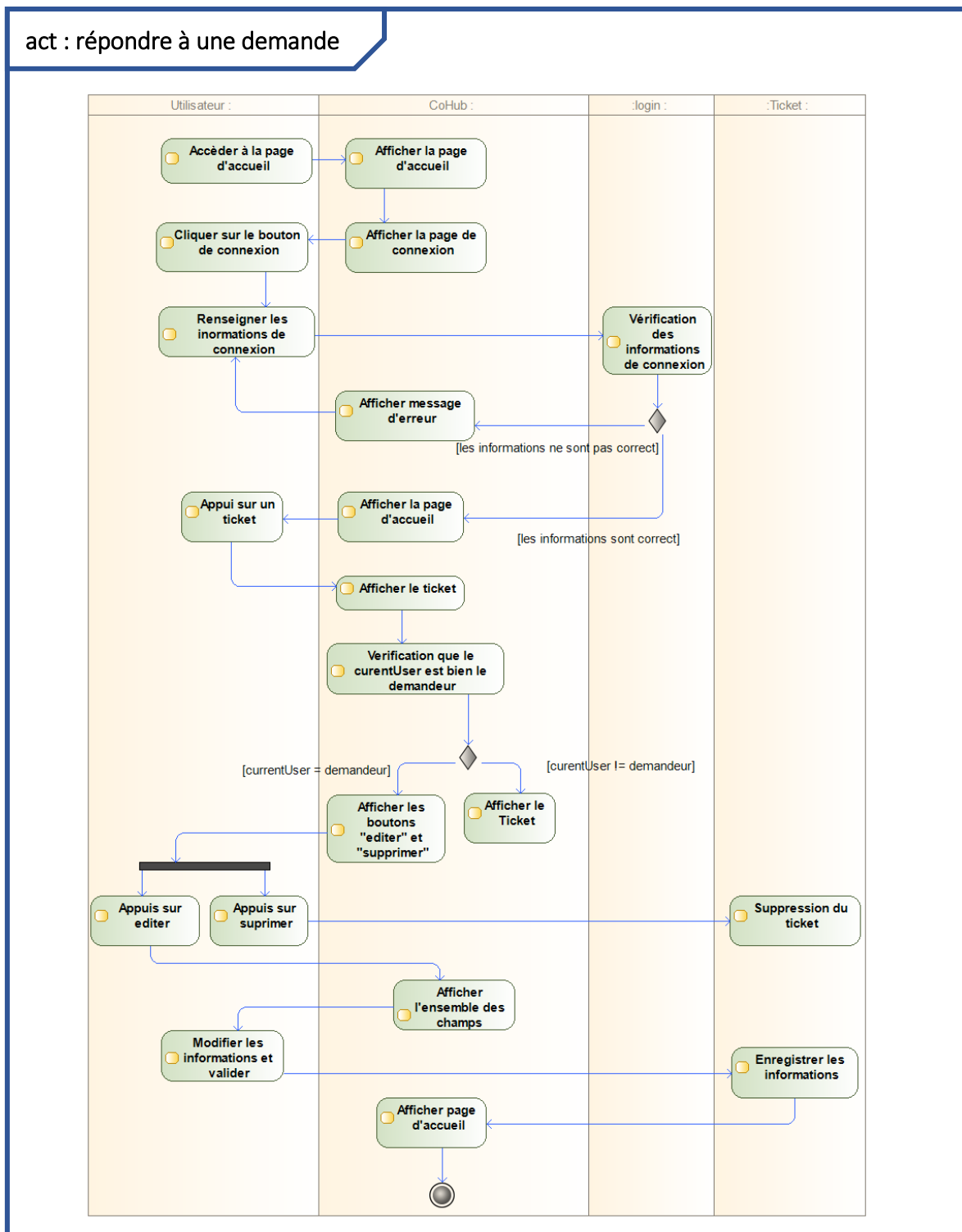
Le diagramme d'activité permet de décrire le comportement des classes entre elles.

### IV.6.1- Activité de création d'une demande



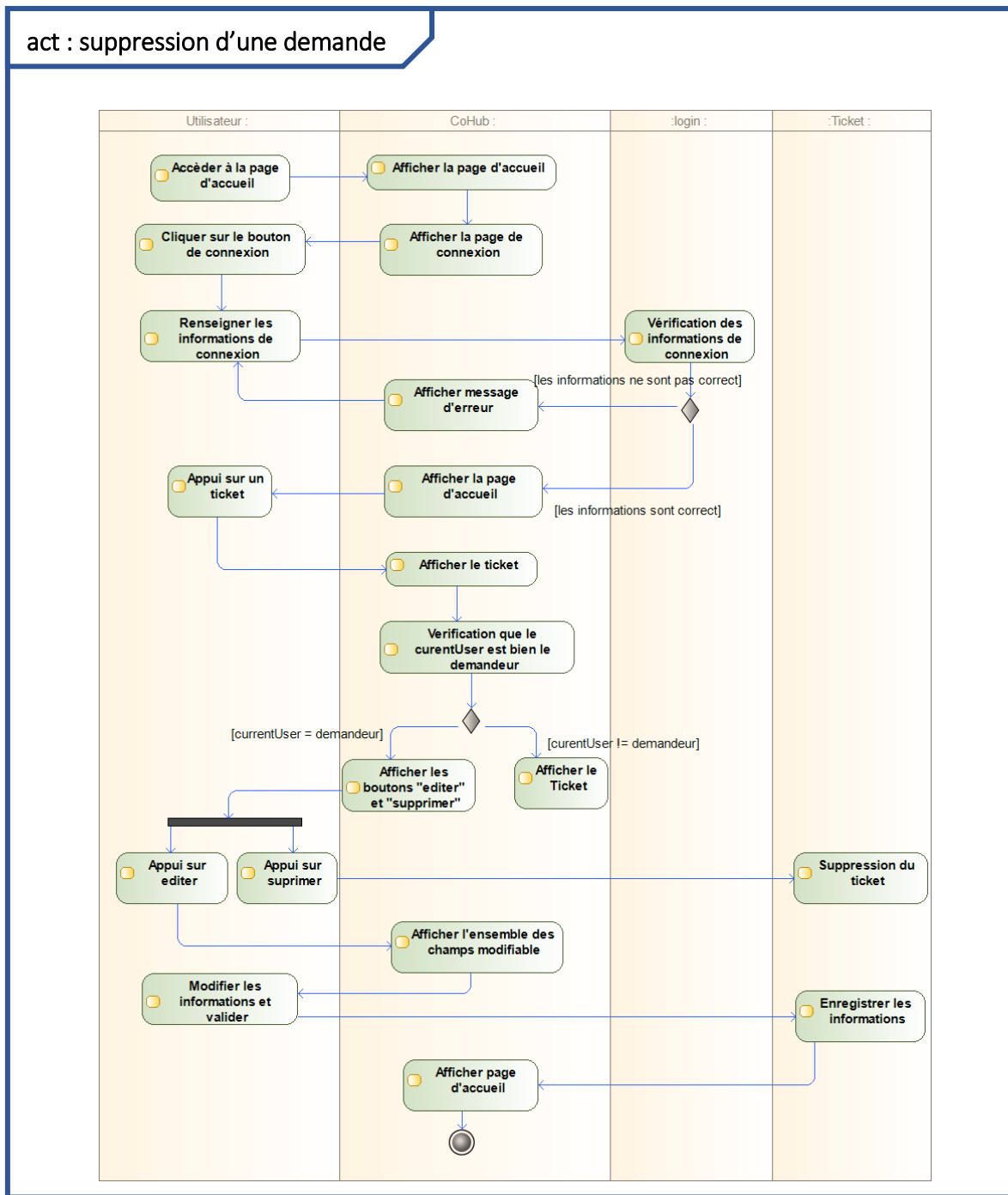
Ce diagramme décrit le processus de création de tickets. Lorsque l'utilisateur renseigne les informations nécessaires pour la demande, celles-ci sont enregistrées dans la base de données.

## IV.6.2- Activité de répondre à une demande



Ce diagramme décrit les étapes d'aide d'une demande. Un utilisateur ne possédant pas la demande peut ajouter un message au ticket. Celui-ci sera enregistré dans la base de données.

### IV.6.3- Activité de suppression d'une demande



Ce diagramme décrit les étapes d'édition ou de suppression d'une demande. Pour ce faire l'utilisateur est vérifié par l'application afin de s'assurer que la demande lui appartient bien. L'utilisateur peut choisir l'édition ou la suppression. La demande est alors supprimée de la base de données ou modifiée dans la base de données.

# V- Conclusion

Au travers de ces diagrammes UML, nous avons pu montrer comment nous souhaitons implémenter notre vision d'une solution de partage, de communication et d'échange. L'utilisation d'un site web et d'une base de données nous permettent une bonne accessibilité et ergonomie.

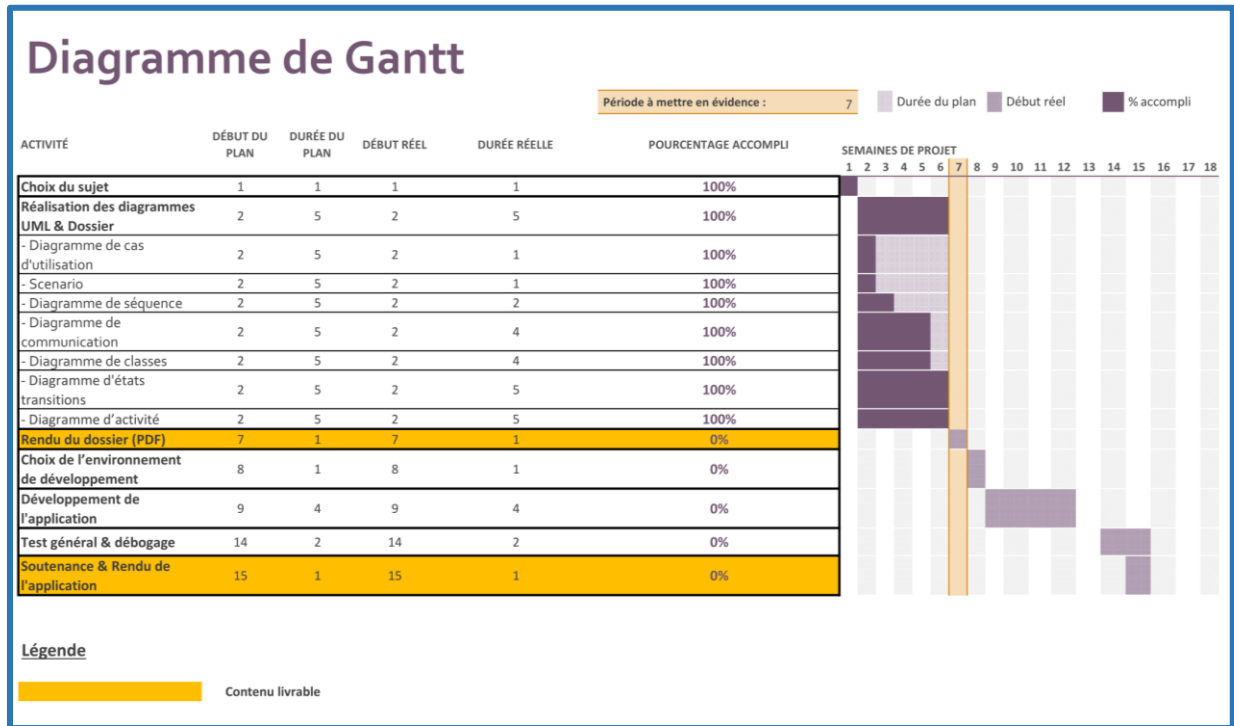
Nous espérons que la solution que nous vous proposons, vous permettra d'améliorer votre expérience à l'ENIB. Un coup de main, une demande de compétence ou une demande de prêt vous seront désormais accessibles plus facilement de n'importe où.

L'ensemble des diagrammes UML que nous avons conçu, nous ont permis d'obtenir une vision claire de la structure globale de notre futur programme. L'intégralité des problèmes rencontrés lors de cette phase de conception nous ont permis d'éviter une perte de temps lors du développement.

Par la même occasion de nous pencher un peu plus sur une partie importante de la programmation : le Web.

# VI- Annexes

## VI.1- Diagramme de Gantt



## VI.2- Présentation de l'équipe

L'équipe est composée de quatre membres :

- CHEVALLIER Anouk. L'informatique n'est pas son domaine de prédilection mais sa rigueur et son travail permettent une très bonne gestion de projet.
- DELABRE Jarod. Maîtrise d'adobe XD pour le design de la maquette. Intérêt pour l'informatique. Polyvalence.
- EVEN Elouan. Capacité d'adaptation. Bon correcteur orthographique. Polyvalence. Préfère la programmation à l'UML.
- LEVEQUE Dorian. L'un de ses langages préférés ? Le JavaScript. Son expérience en programmation a été très utile lors de la création des diagrammes UML.

Même si nous n'envisageons pas tous de choisir un module de spécialité informatique, la conception du site a été réalisée en équipe, ce qui a entretenu la motivation de chacun et nous a permis individuellement et collectivement de progresser.

### VI.3- Pourquoi CoHub ?

Nous avons choisi d'appeler notre site CoHub car c'est la concaténation -c'est-à-dire le regroupement- de Co et de Hub qui illustrent les deux idées directrices de notre projet.

Co pour coopération, coordination, collaboration et Hub pour organisation mettant en œuvre une connexion centrale ou une plateforme de correspondance.

Le logo, avec ces deux mains, exprime ces idées de "coup de main", de main tendue et de "give me 5" de la réussite partagée.



# CoHub