

Conception : Gestion d'une médiathèque

CHAPELLE Cédric
DE ALMEIDA David
MAHDAVI Shahram

Choix des langages

Comparaison des deux solutions

Afin de choisir la solution la plus adaptée à nos besoins, nous avons effectué différentes recherches pour pouvoir comparer les deux langages web mais aussi leurs framework respectifs.

Comparaison des langages

Critère	JAVA	PHP
Paradigme	Orientée objet, structurée, impérative	Impératif, objet, fonctionnel, procédural, réflexif. Langage interprété
Typage	Statique, fort, sûr, nominatif	Dynamic, Faible
Version	8	5.5.11
Compatibilité ascendante	OUI	Pas toujours
Multiplateforme	OUI	OUI
Type d'application cible	Toutes	Web
Gestion UTF-8	OUI	Partielle
Slogan	Write once, run anywhere (WORA)	PHP is a popular general-purpose scripting language that is especially suited to web development.
Popularité (selon l'index TIOBE)	16,9%	3,38%
Dynamisme (selon l'index TIOBE)	-0,01%	-2,4%
Pré-requis	Une JVM	Un interpréteur PHP et un serveur web (Souvent Apache)
Lourdeur de mise en place	Moyen	Faible
Environnement de développement	Eclipse, IntelliJ, NetBeans, JDeveloper...	Simple éditeur de texte, ou IDE intégré

Source : <https://blog.axopen.com/2014/06/java-vs-php-creation-dune-application-web-site-web-en-2014/>

Comparaison de la maintenance

Le langage JAVA est compilé avant d'être exécuté ce qui permet d'éviter un grand nombre de bug. Cependant lorsqu'une erreur d'exécution est détectée alors l'application cesse de fonctionner. Quant à PHP, le langage n'est pas compilé ce qui oblige le développeur à exécuter le programme afin d'y déceler les anomalies potentielles. De plus, l'anomalie sera plus difficilement identifiable car la qualité des logs PHP est plus faible que celle présente en JAVA.

Comparaison des serveurs

JEE nécessite un serveur d'application ou au moins un conteneur de servlet de type Tomcat pour déployer sa solution. Pour héberger un site web, il existe un large panel d'offre mettant à la disposition du client un répertoire et un serveur apache. Le déploiement d'un site web est donc beaucoup plus simple et moins coûteux que celui d'une application web.

Comparaison de l'évolutivité

Le framework JEE évolue beaucoup moins vite que les framework PHP. De plus, il s'engage à toujours être rétroactive contrairement à ses opposants. Cette non-rétroactivité oblige les développeurs PHP à effectuer une recherche et une modification fastidieuse du code source à chaque montée de versions. Il en va de même pour chaque nouvelle version de la librairie. Il faut aussi prendre en compte que la majorité des applications mobile Android sont développés en JAVA ce qui implique une meilleure portabilité du code JAVA.

Au niveau sécurité

Il est plus facile de sécuriser une application web JAVA qu'un site web développé en PHP. En effet, le framework JSF (JavaServer Faces) implémenté dans JEE permet de se prémunir contre certaines attaques telles que l'injection SQL. De plus, le contrôle et la validation des données est directement incluse au cœur du langage contrairement au PHP où les variables ne sont pas typées.

Conclusion

Dans un premier temps nous voulions développer un site web en PHP car nous pensions que le développement et le déploiement de l'application serait plus simple. Cependant, après avoir effectué les recherches pour la comparaison des deux solutions, nous avons estimé que les avantages de l'application web JAVA étaient plus significatives que ceux du site web PHP. Le critère de sécurité des données a été un poids majeur dans notre décision. C'est pourquoi nous allons développer l'application web à l'aide du framework JEE.

Diagramme de classes

Classe Utilisateur :

Id : Elle représente l'identifiant de l'utilisateur.

Elle lui permettra de se connecter au site web.

Elle est unique pour chaque utilisateur.

Nom : Le nom de l'utilisateur ne peut être mais n'est pas unique.

Prénom : Le prénom de l'utilisateur ne peut être mais n'est pas unique.

Update() :

Paramètre : Nom et prénom

Description : Modifie les données d'un utilisateur

Classe Membre :

Solde : Elle représente le solde de l'utilisateur et ne pourra être négatif.

Défaillant : Si le booléen est vrai alors l'adhérent ne pourra réserver aucune ressource.

Update() :

Paramètre : Nom et prénom et solde

Description : Redéfinition de l'update de l'utilisateur en rajoutant en paramètre le solde

Bannir() :

Paramètre :

Description : Cette procédure permettra de rendre un adhérent défaillant

Créditer() :

Paramètre : un montant

Description : Crédite un adhérent du montant donné

Classe Gestionnaire :

Publier() :

Paramètre : Un titre et un texte

Description : Publie un nouvel article à la date du jour

Classe Informations générales :

Id : Un nombre auto-incrémenté afin de rendre chaque identifiant unique.

datePublication : Date à laquelle l'article est publié ou à sa dernière modification

titre : Titre de l'article

texte : Contenu de l'article

Update () :

Paramètre : Un titre et un texte

Description : Modifie un article et sa date de publication

Classe Réservation :

Id : Un nombre auto-incrémenté afin de rendre chaque identifiant unique.

dateRéservation : Date à laquelle l'adhérent a réservé sa ressource

dateRetour : Null tant que l'adhérent n'as pas rendu la ressource. La date ne peut être inférieure à la date de réservation

Retour() :

Paramètre :

Description : La date de retour sera enregistrée.

Classe Ressource :

Id : Un nombre auto-incrémenté afin de rendre chaque identifiant unique

nbrExemplaire : Nombre d'exemplaire au total dont dispose la médiathèque. Il ne peut être négatif.

Caution : Acompte à payer pour pouvoir réserver le livre. Il ne peut être négatif.

nbrJourRetour : Nombre de jour dont dispose l'adhérent pour rendre la ressource. Il ne peut être inférieur à 3 jours.

Déréféréncé : Champ booléen permettant de savoir si la ressource est toujours référencée. Si le nombre d'exemplaire est de 0 alors la ressource sera déréféréncée.

Achat() :

Paramètre : nombre d'exemplaire acheté

Description : Additionne le nombre d'exemplaire acheté au nombre d'exemplaire déjà en stock

Perte() :

Paramètre : L'origine de la perte

Description : Décrémente de 1 le nombre d'exemplaire et créer une perte

Referencement() :

Paramètre :

Description : Donne la valeur FALSE au champ déréféréncé

Dereferencement() :

Paramètre :

Description : Donne la valeur TRUE au champ déréférencé

Update() :

Paramètre : Une caution et un nombre de jour pour le retour

Description : Modifie la caution et le nombre de jour pour le retour

Classe Perte

Id : Un nombre auto-incrémenté afin de rendre chaque identifiant unique

originePerte : Il existe 3 catégories d'origine d'une perte :

« Dégradation naturelle », « Dégradation volontaire » et « Vol »

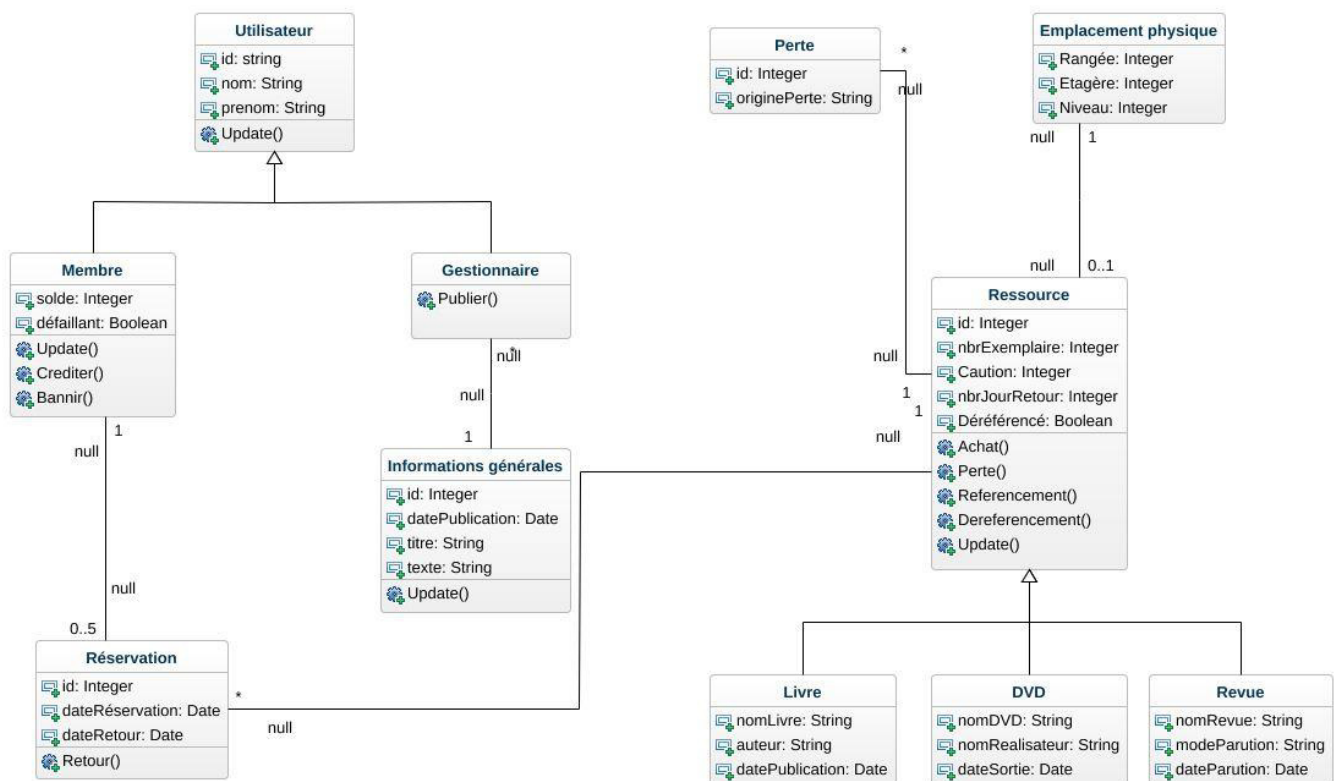
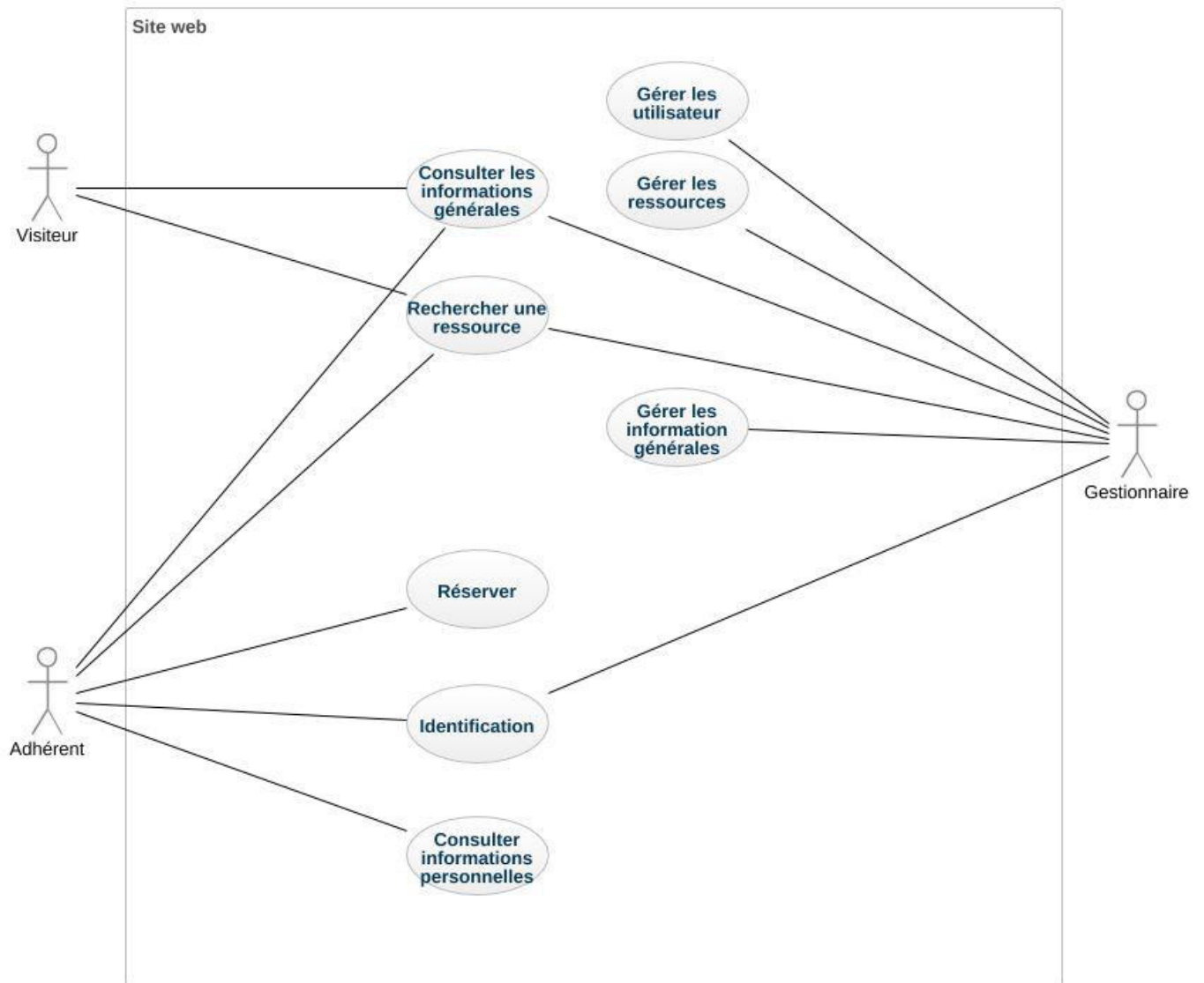


Diagramme des cas d'utilisation



Diagrammes de séquence

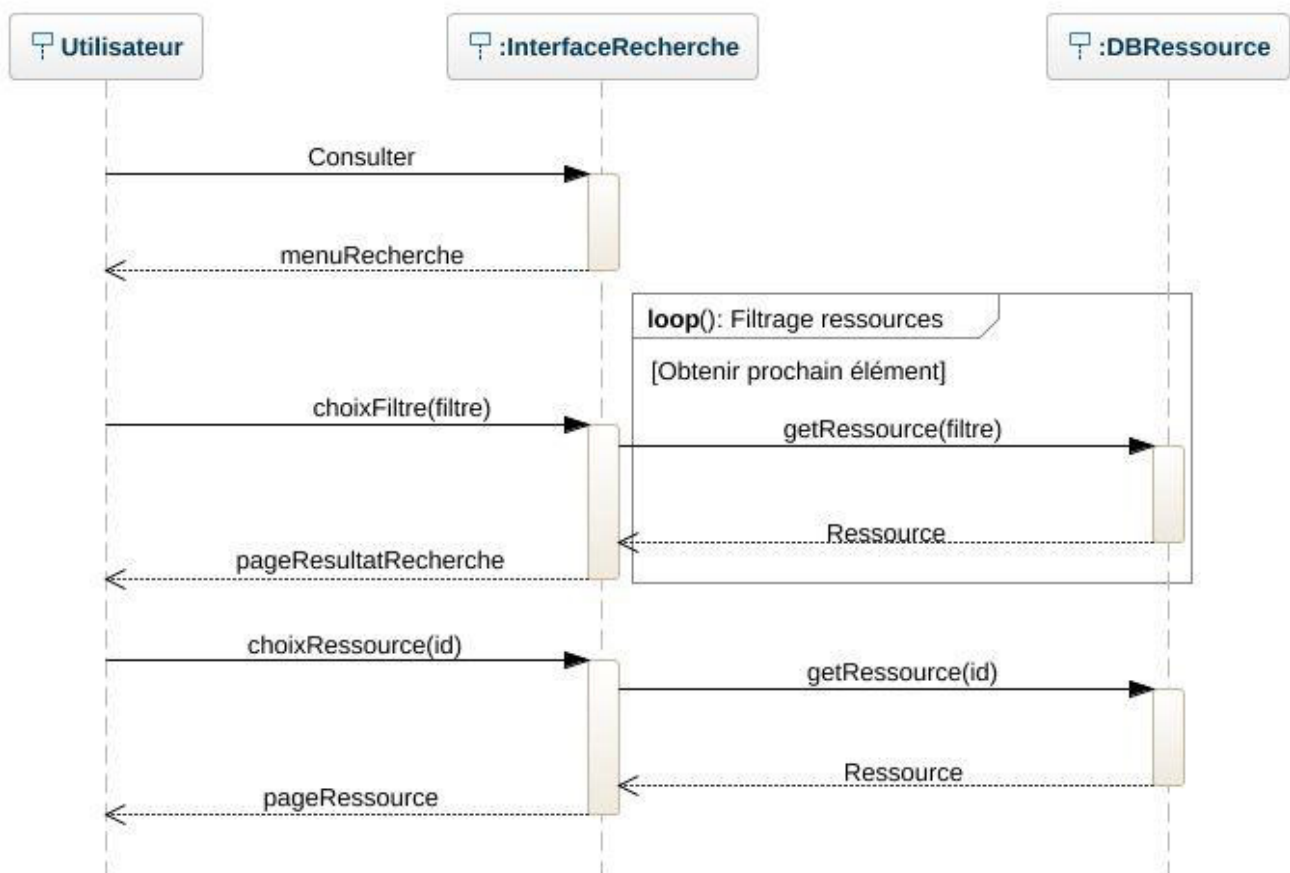
Un utilisateur recherche un ouvrage et lit sa présentation.

L'utilisateur effectue une recherche avancée en demandant au site web l'accès à la page lui permettant de filtrer les ressources selon différents critères.

Une fois les filtres choisis, l'utilisateur confirme et envoie ceux-ci au site web qui effectue une requête SQL à la base de données afin de recevoir les résultats correspondants.

Le site affiche les résultats à l'écran de l'utilisateur et celui-ci sélectionne la présentation l'intéressant.

Une requête SQL est envoyée à la base de données par le site afin de récupérer les informations de la présentation pour l'afficher.

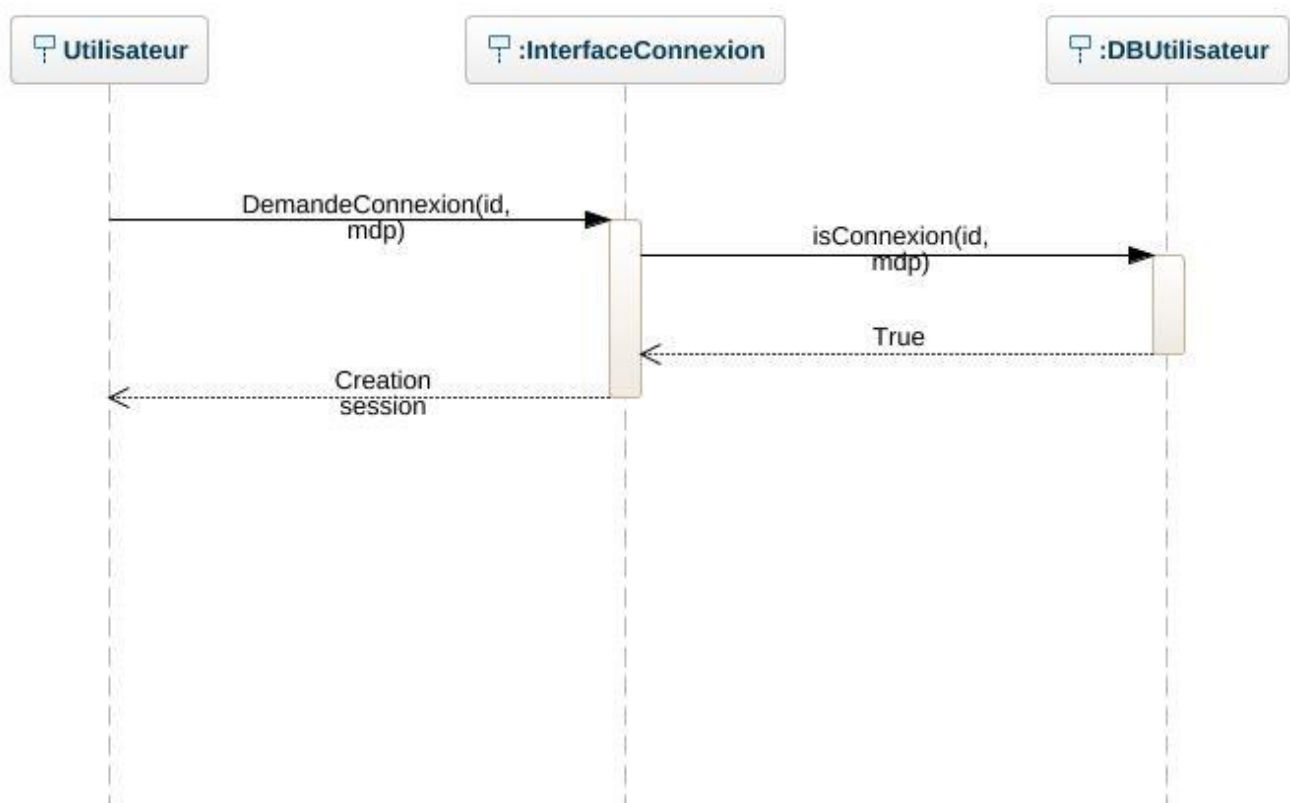


Connexion d'un utilisateur réussie.

Un utilisateur demande à se connecter en entrant ses informations de connexion (identifiant et mot de passe) et confirme sa demande.

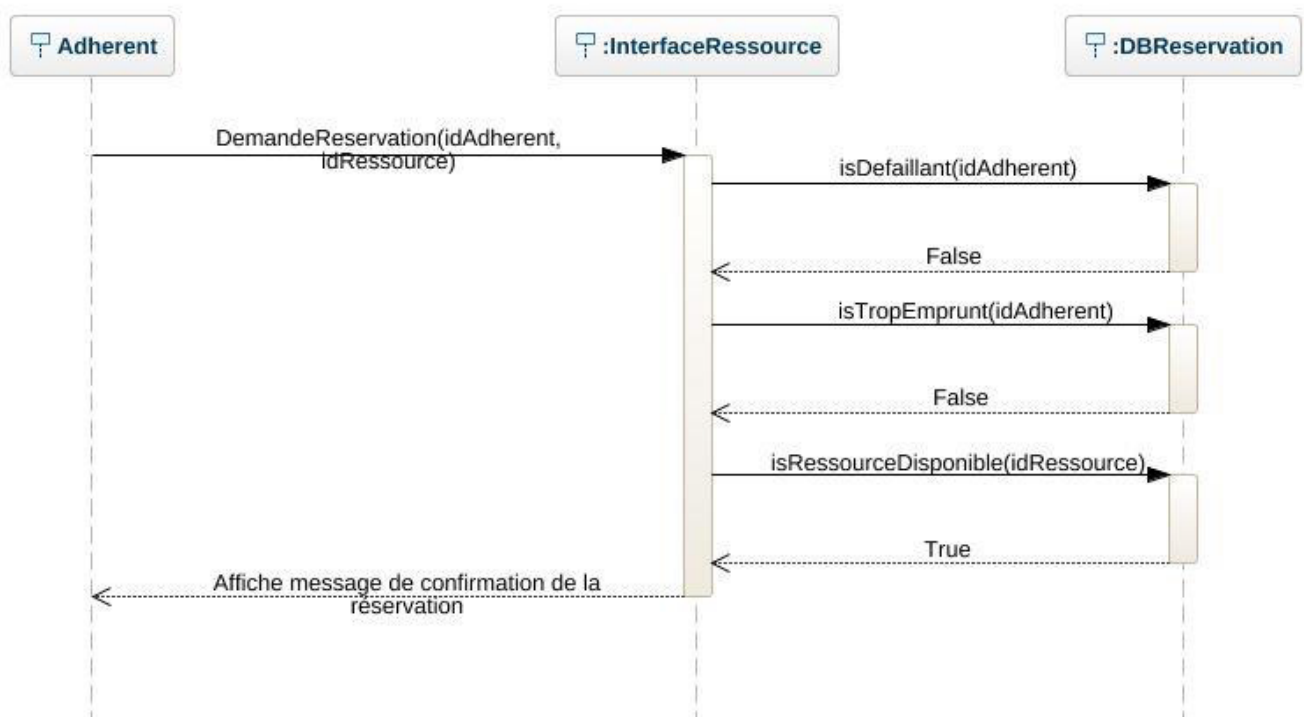
Le site envoie une requête à la base de données afin de vérifier que ces deux informations existent et correspondent à un compte.

C'est le cas donc la BDD renvoie une réponse positive et le site créer une session pour l'utilisateur.



Un adhérent réserve un ouvrage.

Un adhérent demande une réservation sélectionnant la ressource qui l'intéresse. Le site va demander à la base de données si l'adhérent ayant demandé la réservation est défaillant, puis s'il n'a actuellement plus de 5 ressources en cours d'emprunts et en réservation et enfin si la ressource est disponible. L'adhérent a passé tous les tests et un message confirmant la réservation est affiché.



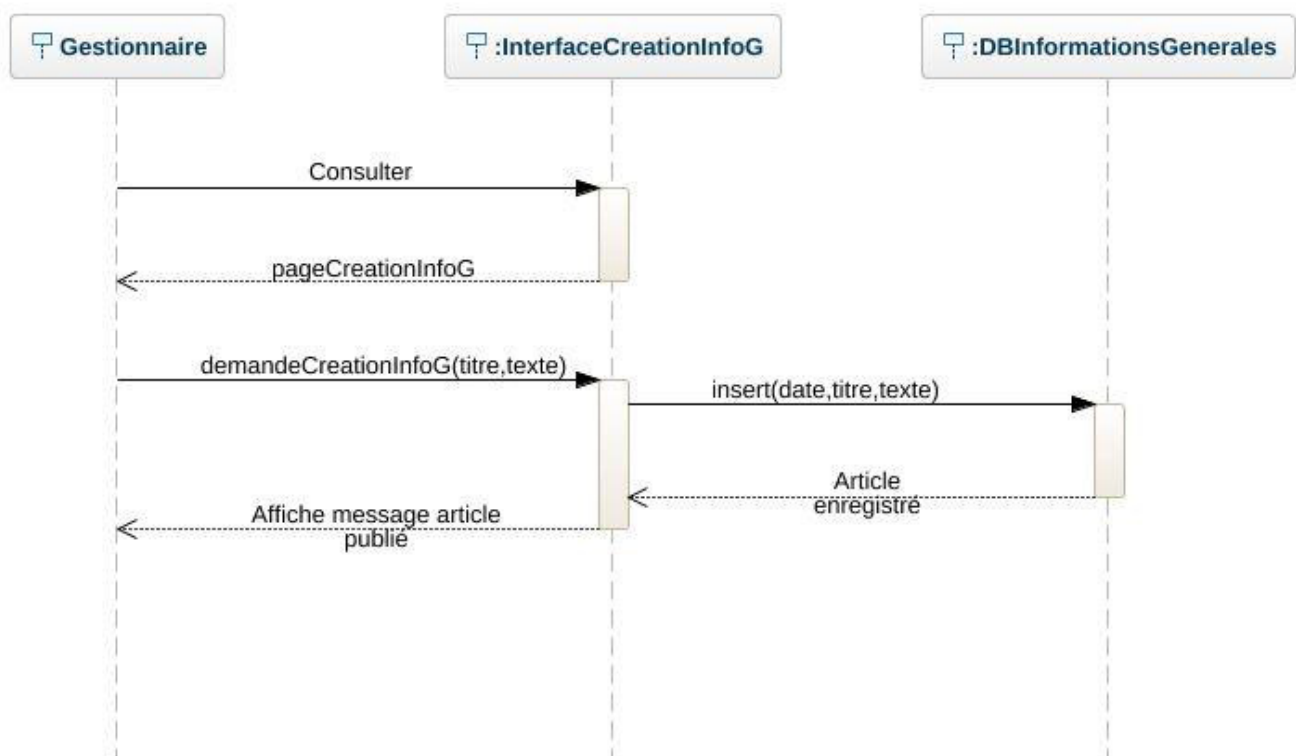
Un gestionnaire créer un nouvel article.

Le gestionnaire demande l'accès à la page de création d'article.

Une fois sur celle-ci, il saisit l'article en y indiquant toutes les informations nécessaire.

Une requête contenant toutes les informations est envoyée à la BDD afin de tout y enregistrer.

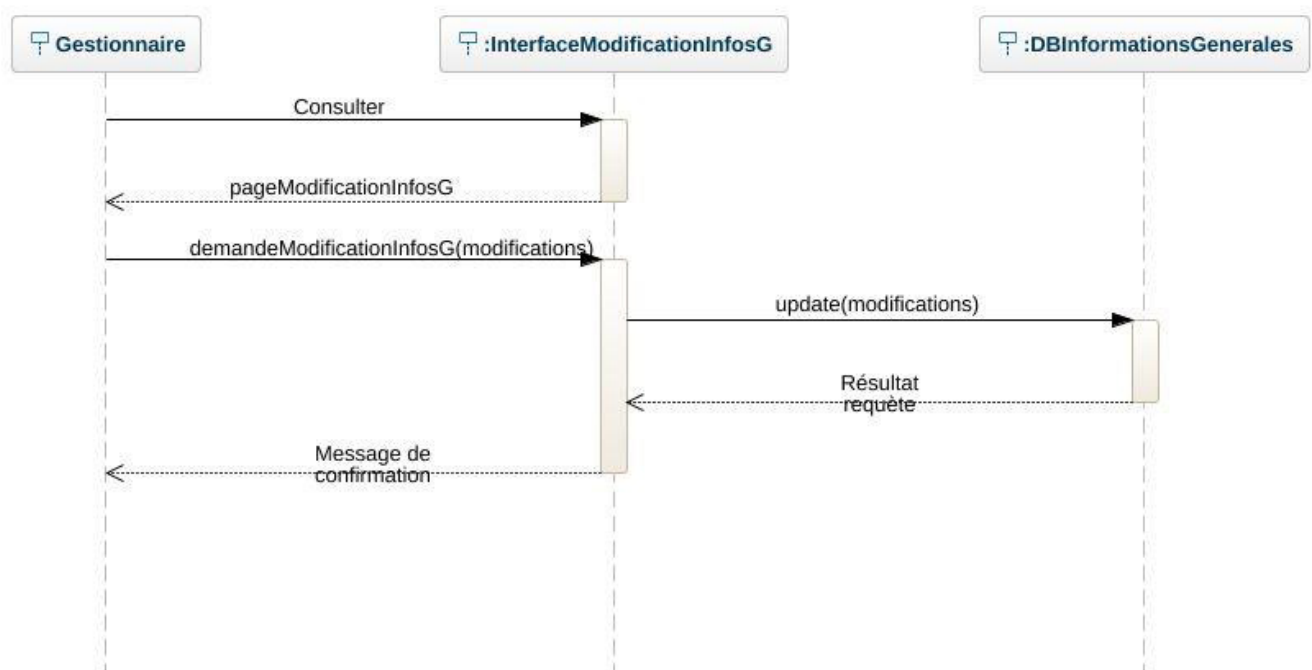
Un message confirmation la publication de l'article est affiché.



Un gestionnaire modifie un article.

Un gestionnaire accède à la page de modification des articles.

Une fois sur cette page, il peut sélectionner l'article à modifier et y saisir les nouvelles informations. Ces informations sont envoyées à la BDD afin de l'enregistrer et un message de confirmation de la modification est affiché.

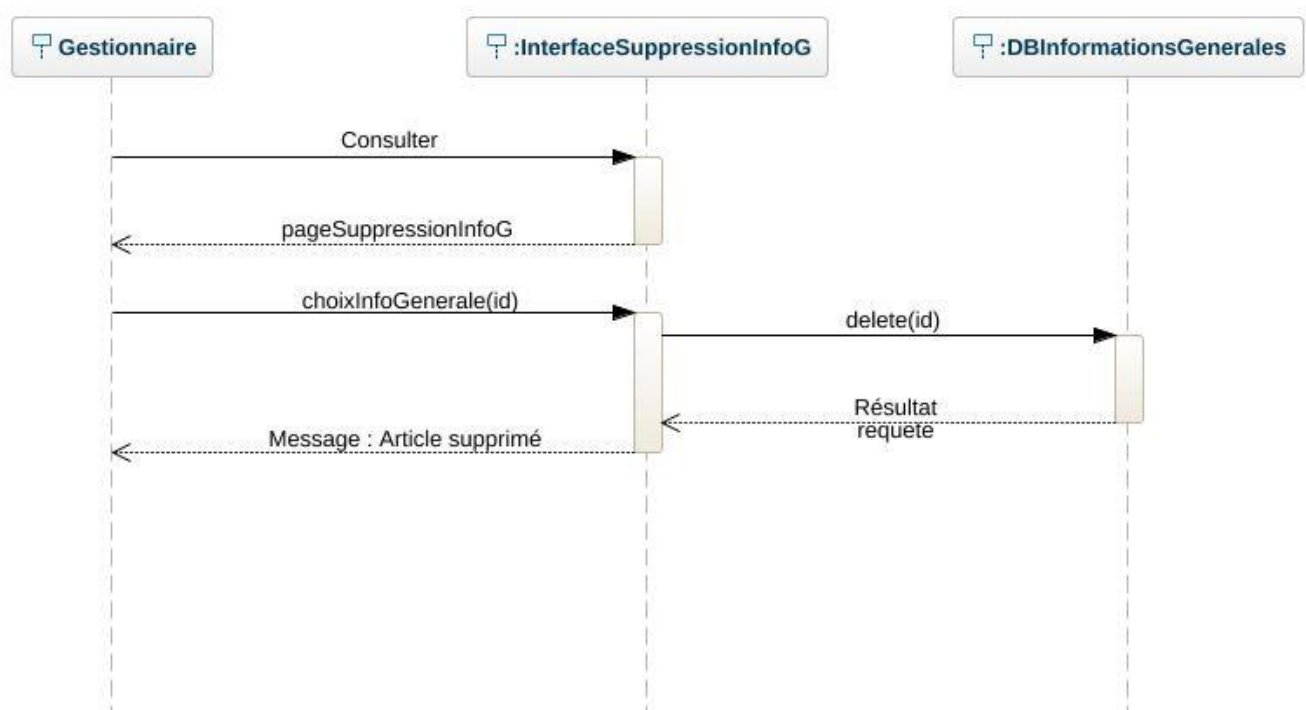


Un gestionnaire supprime un article.

Le gestionnaire accède à la page de suppression des articles.

Sur celle-ci, il choisit l'article à supprimer et une requête est envoyée à la base de données afin de supprimer cet article.

Une fois la suppression dans la BDD effectuée, un message confirmant la suppression est affiché.

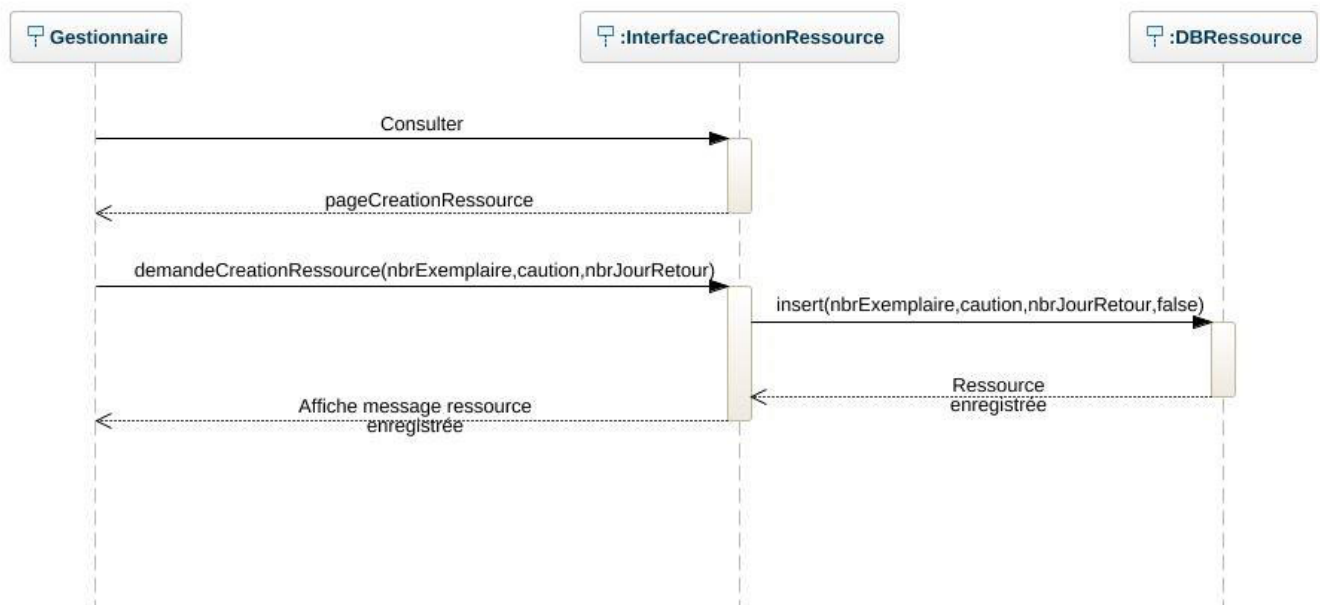


Un gestionnaire créer une nouvelle ressource.

Un gestionnaire accède à la page de création de ressource.

Une fois sur cette page, il entre les informations de la nouvelle ressource et le site envoie une requête afin d'ajouter la nouvelle ressource dans la base de données.

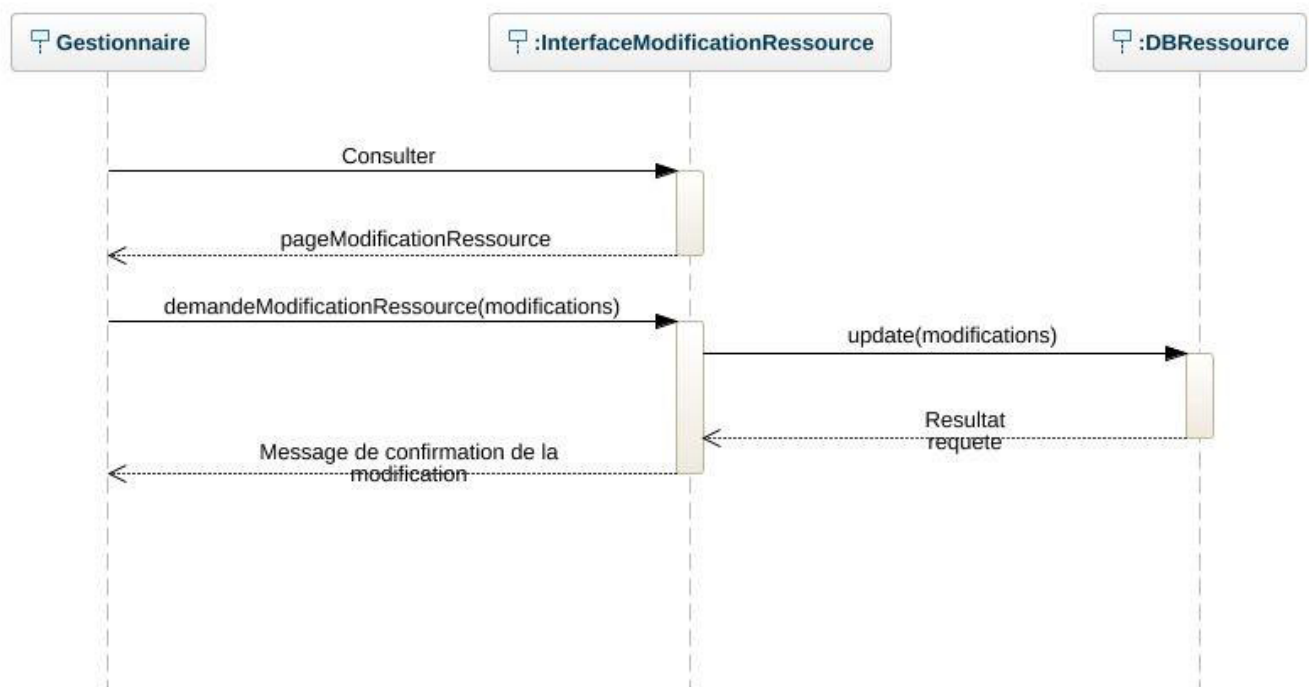
Un message confirmant la création de la ressource est affiché.



Un gestionnaire modifie une ressource.

Un gestionnaire accède à la page de modification des ressources.

Une fois sur cette page, il peut sélectionner la ressource à modifier et y saisir les nouvelles informations. Ces informations sont envoyées à la BDD afin de l'enregistrer et un message de confirmation de la modification est affiché.

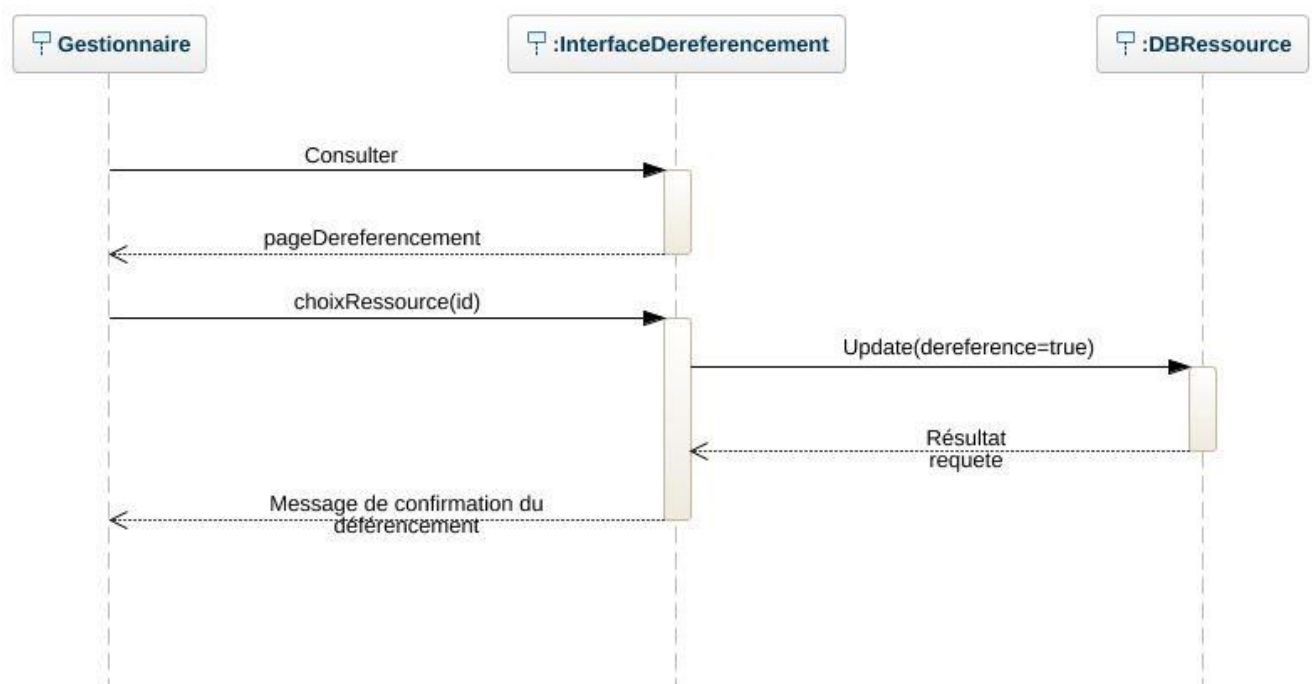


Un gestionnaire déréférence une ressource.

Le gestionnaire accède à la page de déréférencement des ressources.

Sur celle-ci, il choisit la ressource à déréférencer et une requête est envoyée à la base de données.

Une fois la requête effectuée dans la BDD, un message confirmant le déréférencement est affiché.

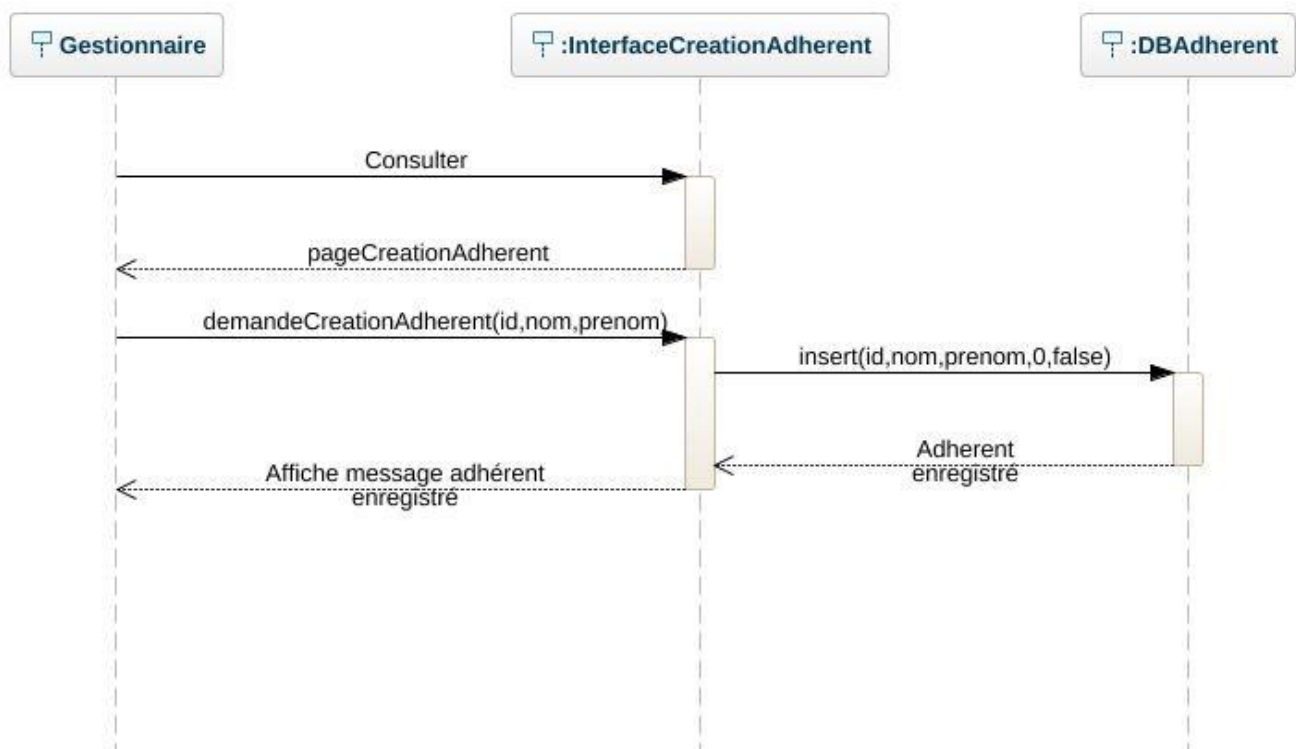


Un gestionnaire créer un nouveau compte d'adhérent.

Un gestionnaire accède à la page de création des comptes d'adhérents.

Une fois sur cette page, il entre les informations de l'adhérent et le site envoie une requête afin d'ajouter celui-ci dans la BDD.

Un message confirmant la création de l'adhérent est affiché.

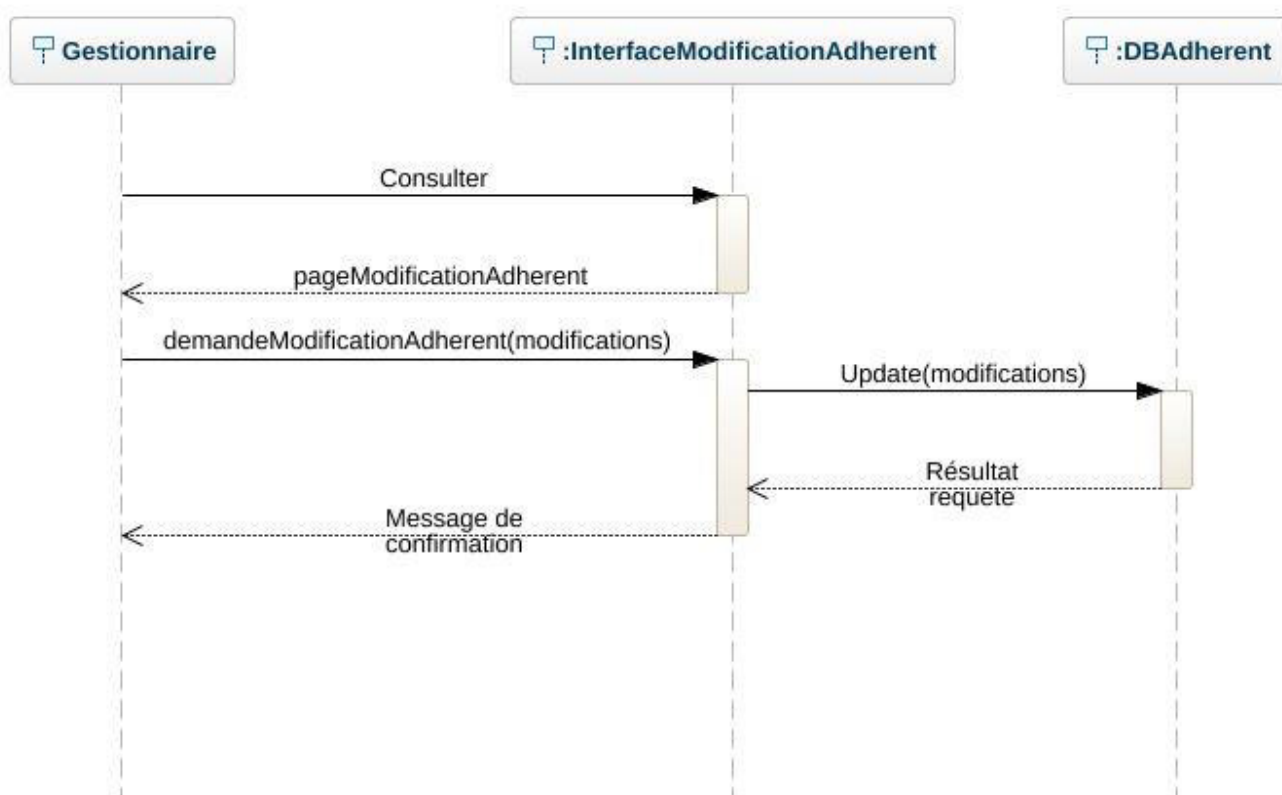


Un gestionnaire modifie les données personnelles d'un adhérent.

Le gestionnaire accède à la page de modification des adhérents.

Sur la page, il choisit l'adhérent à modifier et une requête est envoyée à la base de données.

Une fois la requête effectuée dans la base, un message confirmant la modification est affiché.



Un gestionnaire rend un adhérent « défaillant ».

Le gestionnaire accède à la page « adhérent défaillant ».

Sur la page, il choisit l'adhérent et une requête est envoyée afin de rendre l'utilisateur défaillant.

Un message confirmant que l'action a bien été effectuée est affiché.

