

Projet de modélisation UML et développement JEE d'une application de gestion de matériel.

DOUADY Tom

DUSE Mathieu

GUILLOUX Nathan

LE FORESTIER Erwan

PONTACQ Hugo

KILOUPRESQUETOUT

ENSSAT
LANNION

UNIVERSITÉ DE
RENNES 1

IMR2

Projet de modélisation UML et développement JEE d'une application de gestion de matériel.	1
Cadrage	4
Contexte du projet	4
Objectifs	4
Besoins	4
Solutions	5
Modélisation UML	5
Diagramme de cas d'utilisation	5
Gestion des utilisateurs	5
Gestion du matériel	6
Gestion des prêts	7
Diagrammes d'activités	8
S'authentifier	8
Supprimer un utilisateur / Ajouter un utilisateur	9
Ajouter matériel / Supprimer matériel	10
Rechercher matériel	11
Modifier matériel	12
Nouveau prêt	13
Modifier prêt	14
Consulter prêts en cours	15
Consulter utilisateurs en retards	15
Diagrammes de séquences	29
Supprimer un utilisateur / Ajouter un utilisateur	29
Ajouter matériel	29
Supprimer matériel	30
Rechercher matériel	31
Modifier matériel	32
Nouveau prêt	33
Modifier prêt	34
Retourner prêt	35
Lister les prêts en cours	35
Lister les prêts en retard	36
Diagramme d'état transition	37

Diagramme de classe	39
Implémentation	40
Architecture	40
MVC	40
DAO	41
Le modèle DAO permet aux contrôleurs Java de faire abstraction du modèle SQL. Les contrôleurs vont simplement demander aux classes DAO de stocker des données et de les récupérer.	
Dans notre cas, chaque type d'équipement dispose d'un DAO propre qui correspond à ses caractéristiques.	41
Navigation	41
Un servlet permet de faire le lien entre les URL et leur vue et contrôleur associé.	
Chaque contrôleur dispose également d'une méthode doGet qui vérifie que l'utilisateur qui demande a voir la page dispose des droits nécessaires pour avoir accès à la page.	
La méthode doPost est optionnelle en fonction des pages, elle permet d'envoyer des informations saisies par l'utilisateur dans différents formulaires aux contrôleurs correspondants et donc d'effectuer les actions nécessaires.	41
Fonctions implémentées	42
Authentification	42
Gestion des utilisateurs	42
Gestion du matériel	42
Gestion des prêts	42
Fonctions non implémenté / en cours d'implémentation	43
Mails	43
Outils utilisés	43
En Pratique	44
Accueil	44
Les profil utilisateurs	44
Gestion du matériel	45
Gestion des Emprunts / Prêts	47
Annexes	49
Maquettes	49
Base de données	50
Dump de la base	53

1. Cadrage

a. Contexte du projet

KilouPresqueTout est une société souhaitant moderniser son suivi de prêt matériel. Dans ce contexte, le projet répond à ce besoin par une application de gestion de prêt de matériel. Dans ce logiciel on peut louer des ordinateurs, véhicules jusqu'à des vidéoprojecteurs.

b. Objectifs

L'objectif est de permettre aux membres d'une entreprise d'emprunter du matériel via notre logiciel. La gestion des emprunts et des retards éventuels doit être prise en compte. Une gestion des utilisateurs doit être mise en place car seuls certains rôles disposent des autorisations nécessaires pour créer un emprunt ou éditer le matériel en stock. Un administrateur pourra également avoir accès aux comptes utilisateurs et superviser toutes les autres opérations.

2. Besoins

Les besoins de la société KilouPresqueTout sont multiples. Tout d'abord une application regroupant une gestion d'utilisateurs et une gestion de matériels. Puis pour faire le lien entre les deux, il faut créer une gestion d'emprunt. Toutes les fonctionnalités doivent être disponibles via une interface qui permet à l'utilisateur de naviguer et d'utiliser ces dernières.

Le tout doit être en cohérence et respecter le cahier des charges défini par le client.

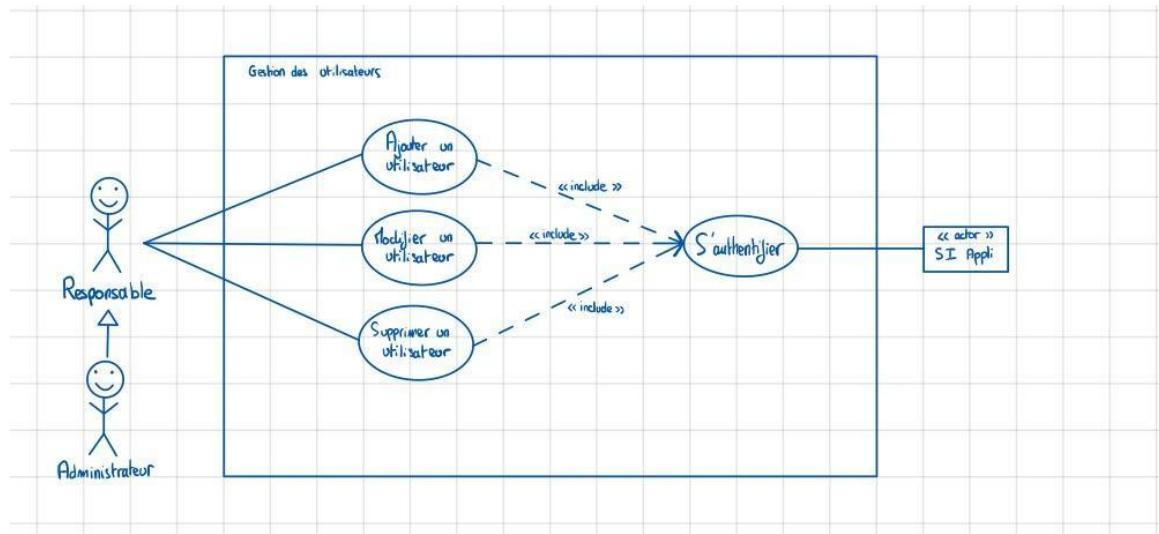
3. Solutions

a. Modélisation UML

Afin de partir sur une base solide, notre équipe suit des concepts et méthodologies spécifiques à l'UML. Nous avons donc réalisé de nombreux diagrammes identifiant les acteurs et les cas d'utilisations.

b. Diagramme de cas d'utilisation

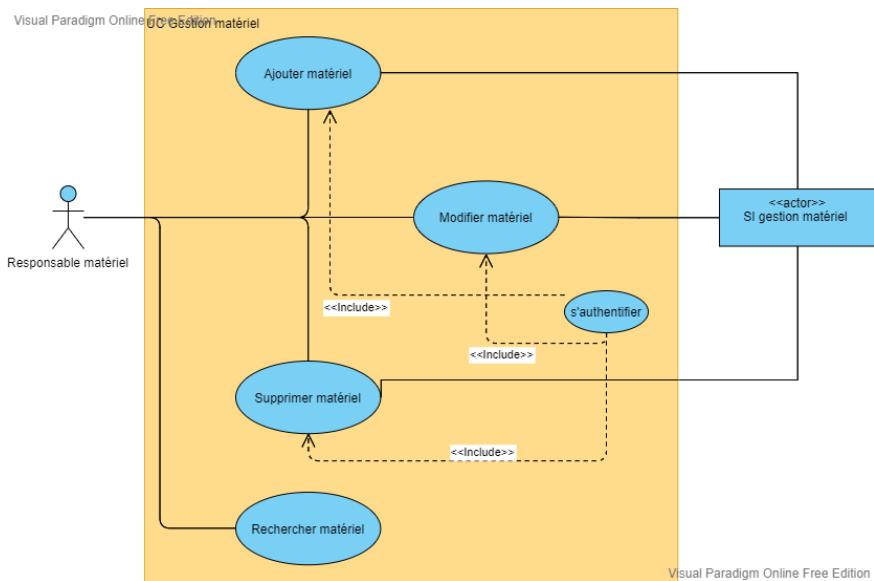
Gestion des utilisateurs



Justification :

On a choisi de définir deux rôles (administration et responsable) afin de pouvoir déterminer qui peut ajouter, modifier et supprimer un ou des utilisateurs. Comme dans les autres cas d'utilisation il faut se connecter afin de pouvoir accéder aux fonctionnalités.

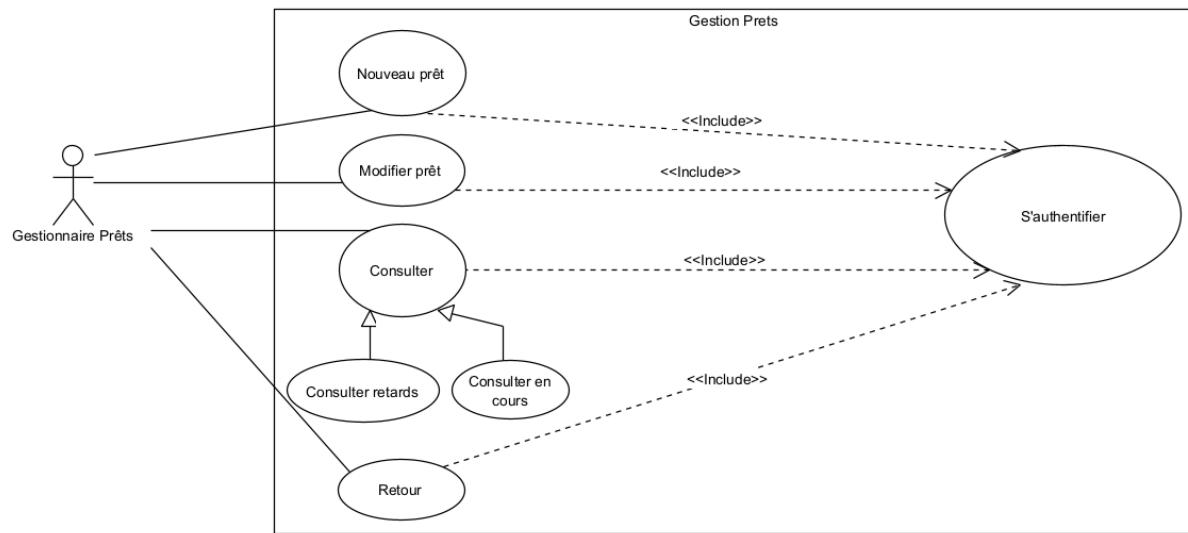
Gestion du matériel



Justification :

Un seul rôle peut interagir avec la gestion du matériel. On a choisi de réaliser un CRUD classique à savoir ajouter, modifier, supprimer un matériel et on a ajouté une fonctionnalité de recherche de matériel. Tout comme les autres cas d'utilisation, il faut se connecter afin de pouvoir accéder aux différentes fonctionnalités.

Gestion des prêts

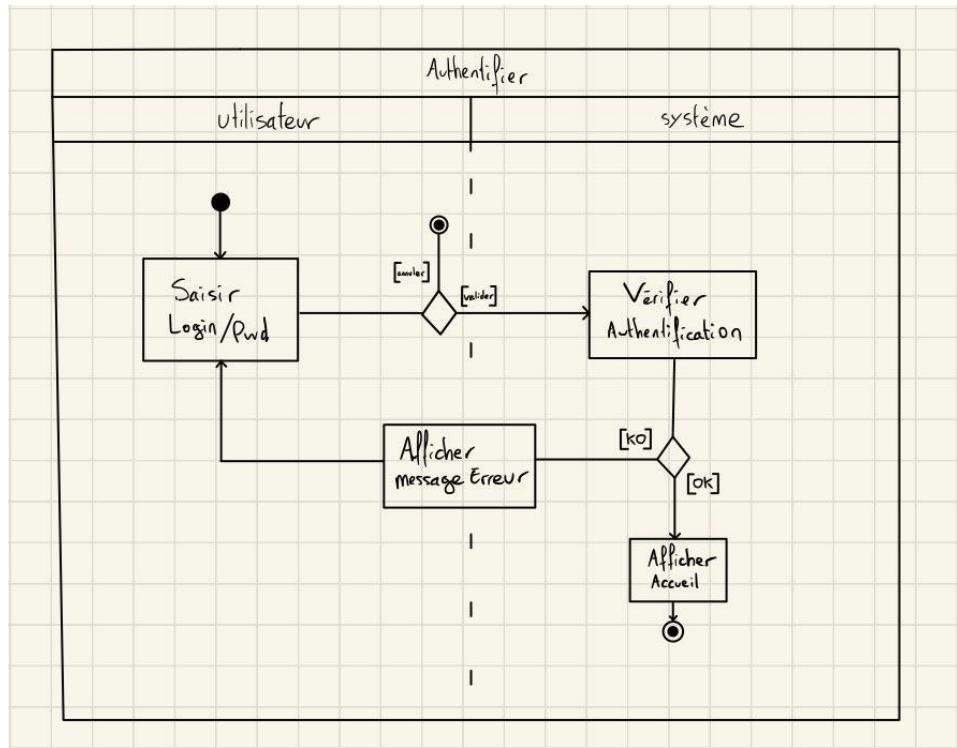


Justification :

Un seul rôle peut interagir avec la gestion du prêt. On a décidé contrairement au CRUD de ne pas supprimer les prêts. Pour se faire, on a simplement implémenté les fonctionnalités de création, modification et consultation de prêt.

c. Diagrammes d'activités

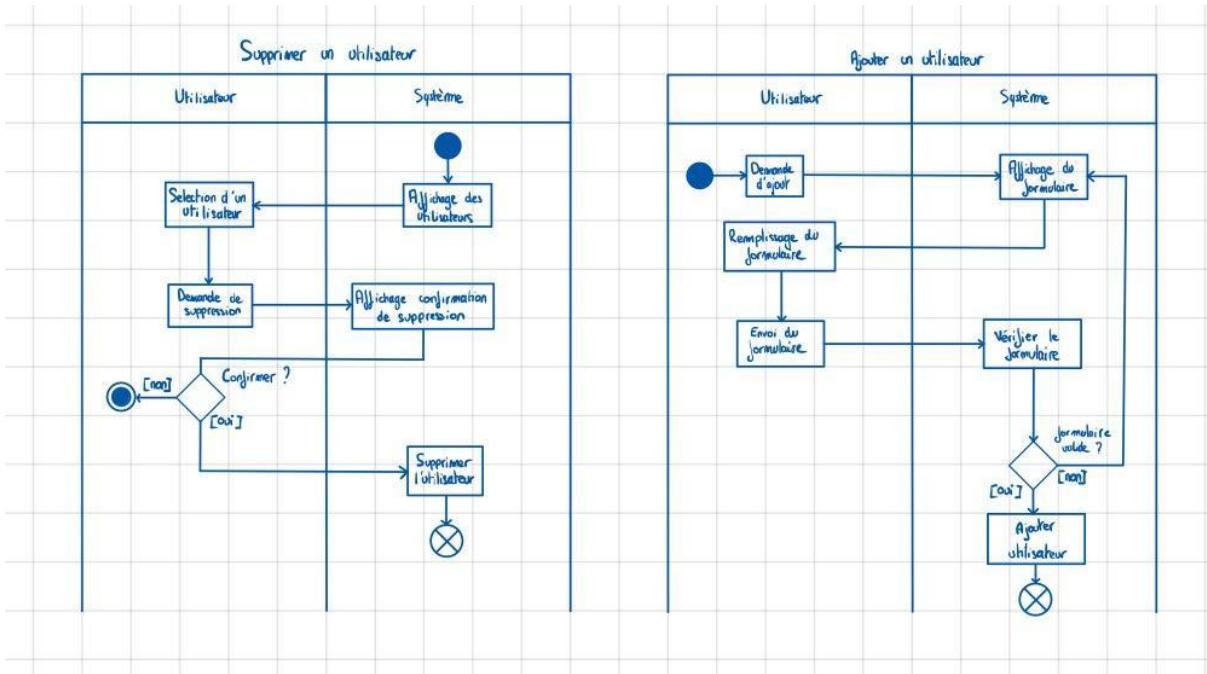
S'authentifier



Justification :

Pour ce diagramme d'activité, on a décidé de rester basique et faire en sorte qu'on puisse annuler ou valider l'authentification à tout moment. Si l'utilisateur entre de mauvaises coordonnées, on affiche un message d'erreur afin que l'utilisateur sache pourquoi il n'arrive pas à se connecter.

Supprimer un utilisateur / Ajouter un utilisateur

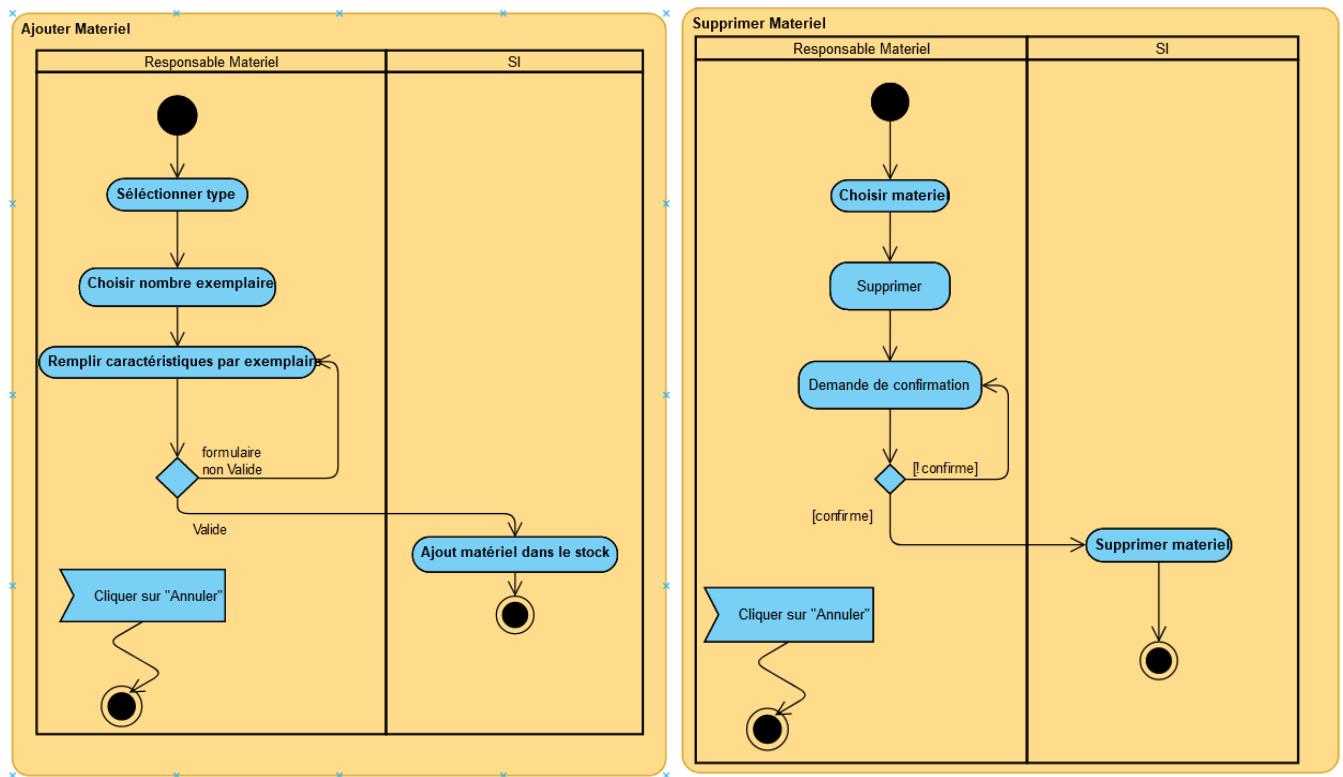


Justification :

Pour supprimer un utilisateur on a choisi qu'il fallait le sélectionner donc la suppression multiple n'est pas disponible.

Pour l'ajout d'utilisateur, de même que pour la suppression on ne peut pas ajouter plusieurs utilisateurs en même temps. C'est un choix que nous avons fait pour la lisibilité et la cohérence dans l'interface avec le reste.

Ajouter matériel / Supprimer matériel

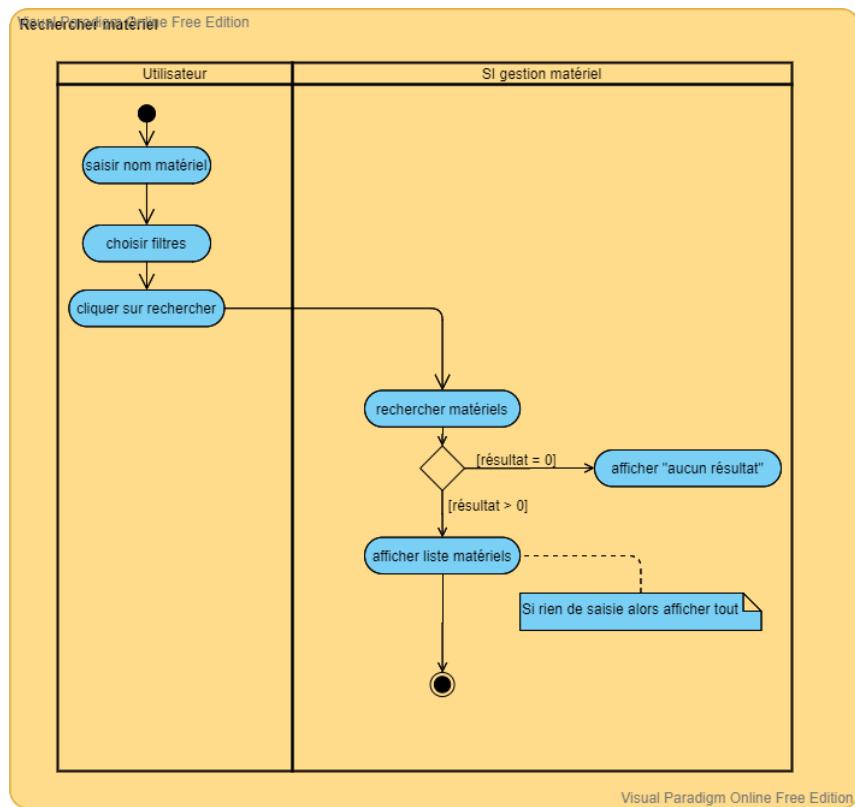


Justification :

Pour ajouter un matériel, il faut choisir un type qui le catégorise pour la suite. On réalise une vérification sur les données pour garder une cohérence dans notre schéma de données. On a choisi qu'on puisse annuler à tout moment le processus.

De la même manière afin de supprimer un matériel on en sélectionne un puis un message de confirmation s'affiche qu'il faut accepter si on souhaite continuer. Cela permet d'éviter des manipulations ratées qui résulteraient dans des pertes de données involontaires.

Rechercher matériel

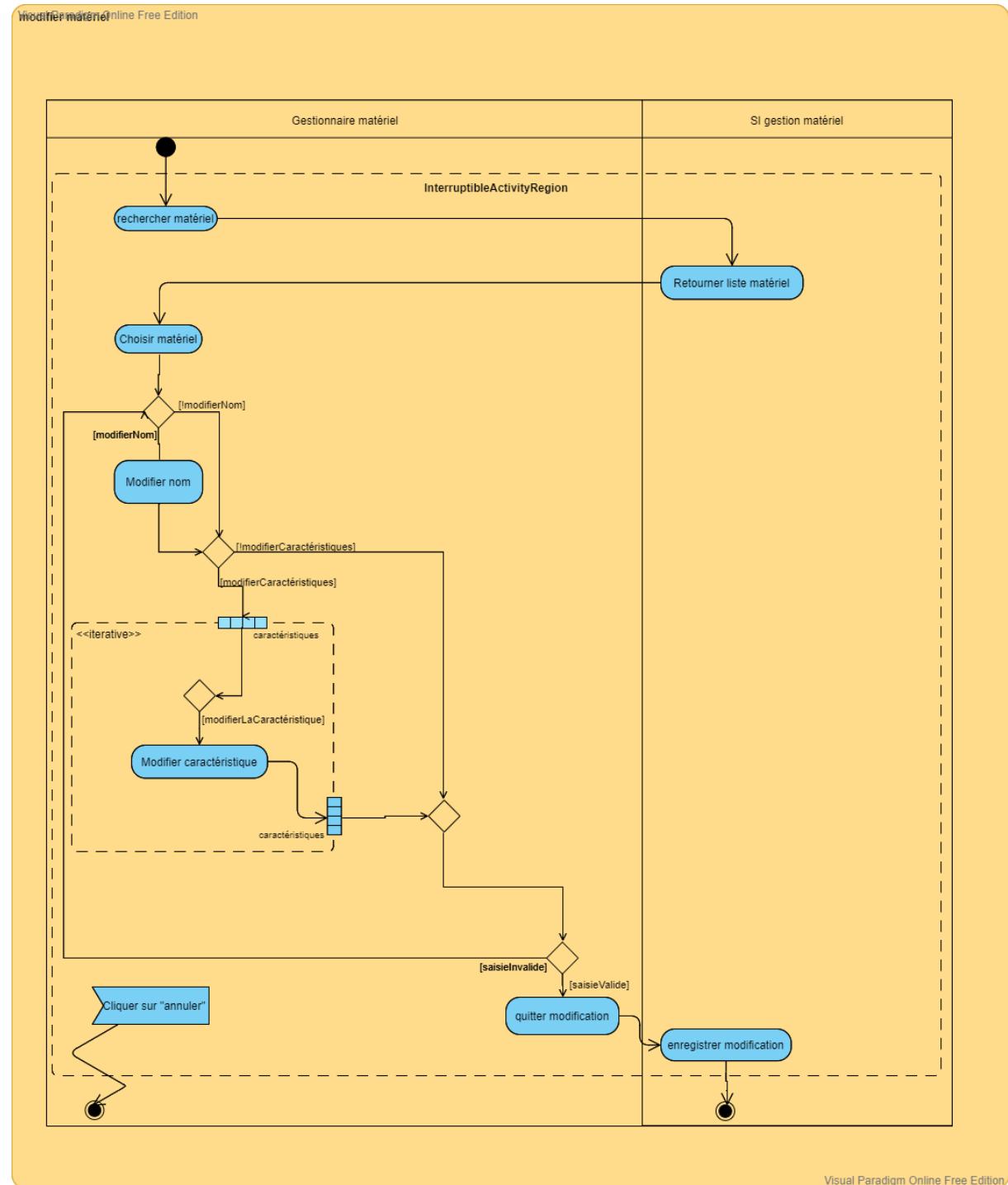


Justification :

Quand on recherche du matériel, on peut effectuer la recherche sur le nom du matériel et on peut également utiliser des filtres. Ce choix est pour faciliter l'utilisation pour le gestionnaire notamment en filtrant sur des catégories mentionnées précédemment.

On a décidé de tout afficher quand on ne rentre aucun nom ni aucun filtre pour éviter d'avoir une page vide qui peut porter à confusion l'utilisateur.

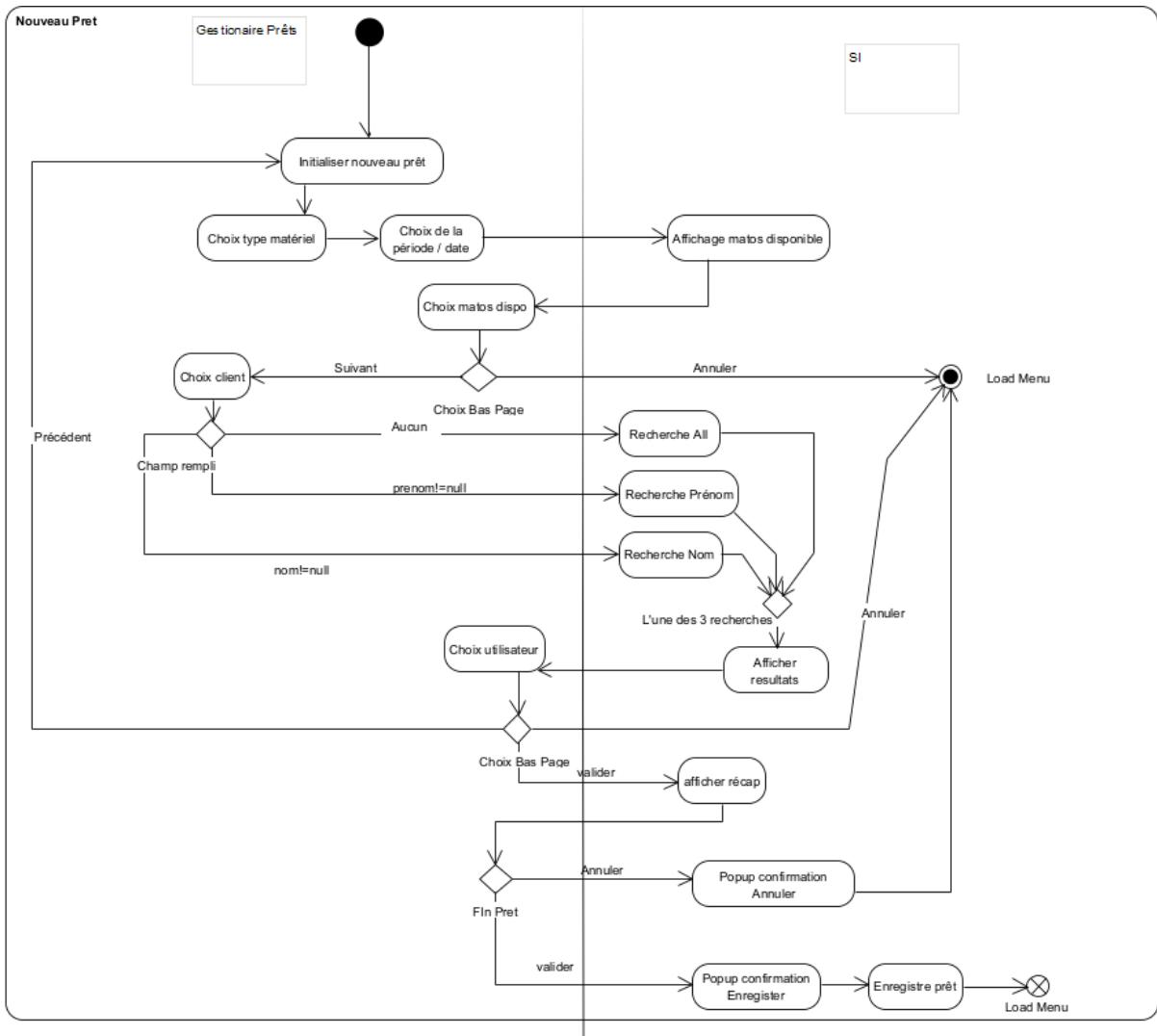
Modifier matériel



Justification :

La modification d'un matériel prend en compte qu'un matériel existe déjà. Ensuite on modifie ses caractéristiques. On peut finalement valider ou annuler. On a décidé de rajouter l'annulation à tout moment.

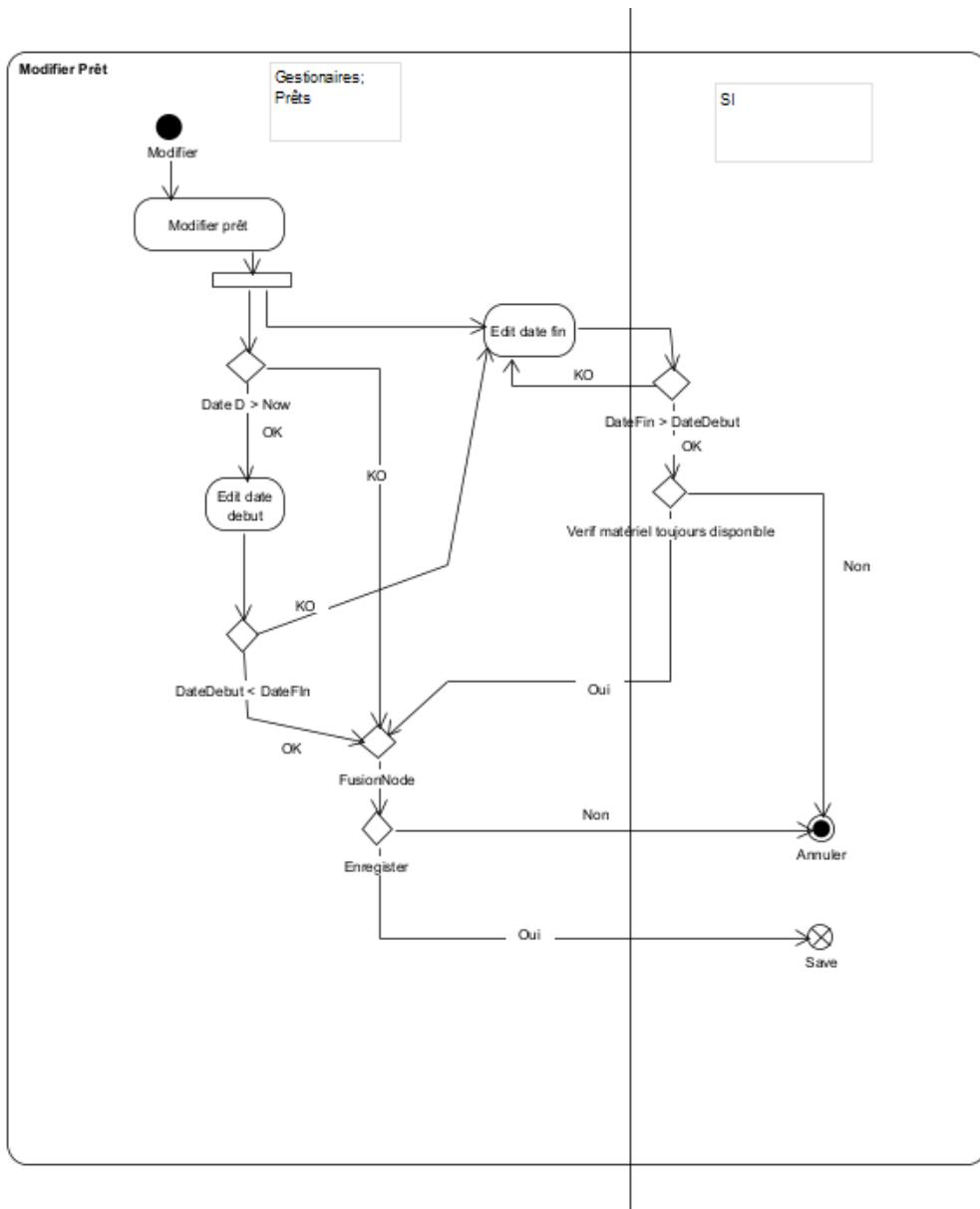
Nouveau prêt



Justification :

Le cas d'utilisation nouveau prêt consiste à choisir un matériel, un utilisateur, une date puis on valide. On fait appel à la recherche de matériel et à la recherche d'utilisateurs pour rendre ce cas ergonomique. On a décidé de choisir la date avant de choisir l'utilisateur pour n'afficher que les matériels disponibles afin d'éviter au gestionnaire de devoir revenir au tout début choisir un matériel puis un utilisateur.

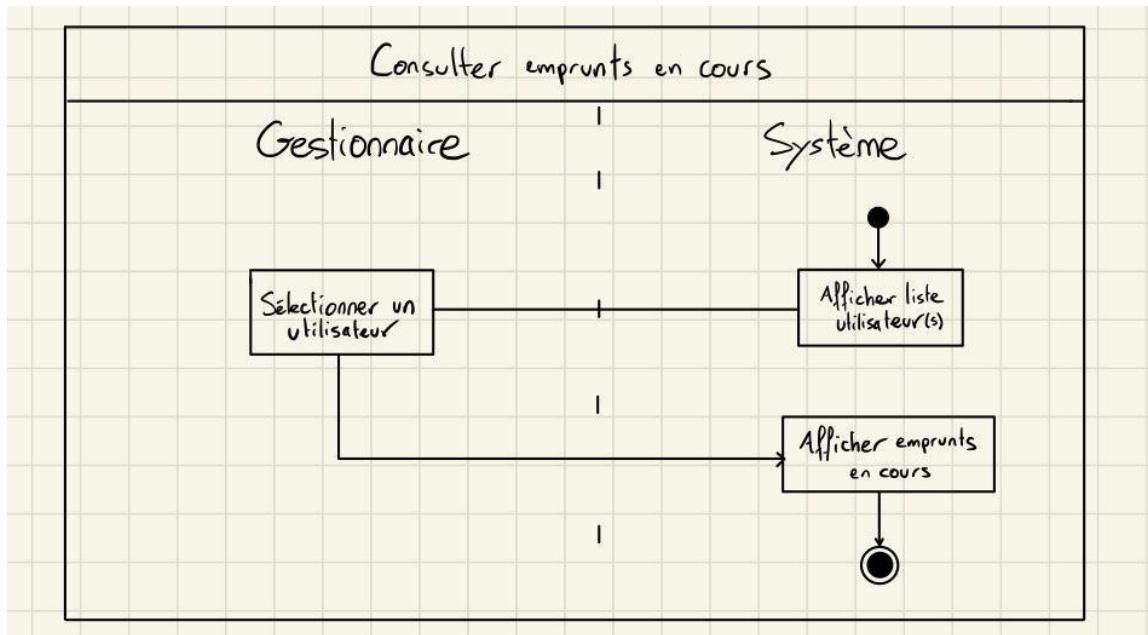
Modifier prêt



Justification :

Modifier un prêt s'assimile à la modification d'un matériel. On peut modifier les dates de début et de fin d'un prêt. Pour chaque modification on vérifie la cohérence des dates et si le matériel est toujours disponible dans l'intervalle de jours choisi.

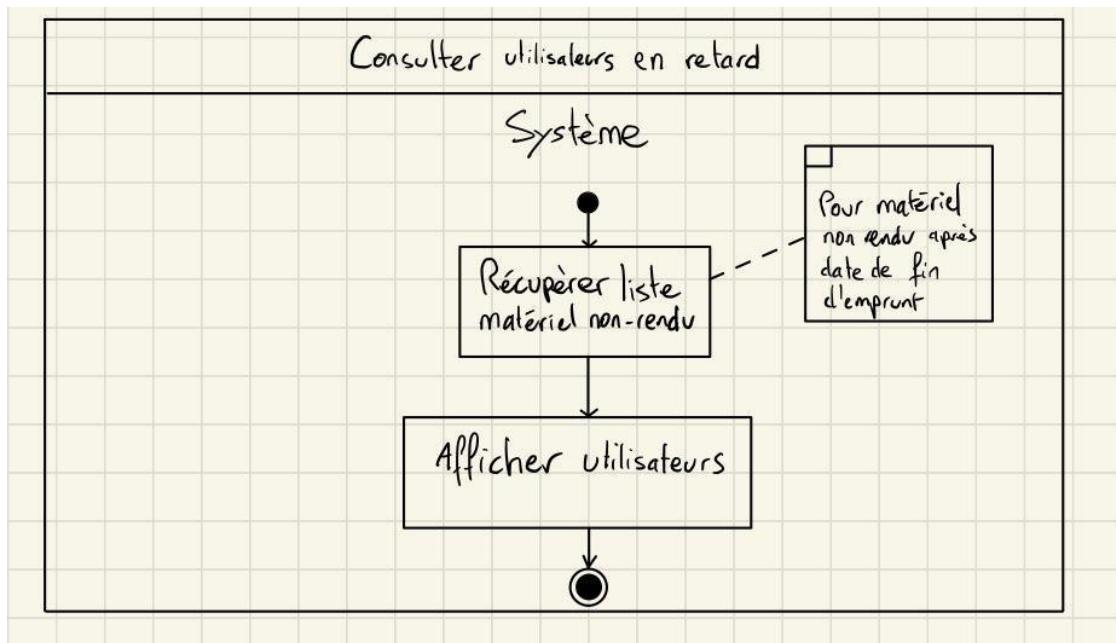
Consulter prêts en cours



Justification :

La consultation d'un prêt en cours consiste à sélectionner dans une liste un utilisateur et de voir ses prêts en cours.

Consulter utilisateurs en retards



Justification :

La consultation d'utilisateurs en retard affiche la liste des utilisateurs qui n'ont pas rendu leurs matériels avant la fin du rendu.

d. Diagrammes textuels

Cas d'utilisation : Ajouter un utilisateur

A. Description

Ce cas d'utilisation concerne l'ajout d'un utilisateur, avec pour objectif de l'ajouter à la base de données des utilisateurs, pour ensuite pouvoir utiliser l'application.

B. Enchaînements d'évènements

1. Enchaînement de base

- 1.1 Le responsable demande l'ajout d'un utilisateur
- 1.2 Le système affiche le formulaire de création
- 1.3 Le responsable remplit le formulaire
- 1.4 Le responsable envoie le formulaire
- 1.5 Le Système d'information vérifie le formulaire
- 1.6 Le Système d'information ajoute le nouvel utilisateur
- 1.7 Le Système d'information informe le responsable du succès de l'ajout

2. Enchaînement alternatif

- 2.1. Enchaînement alternatif 1 : A l'étape 1.5 : Le formulaire n'est pas rempli ou incomplet
 - 2.1.1. Le système d'information envoie un message d'erreur lors de l'ajout
 - 2.1.2. Le responsable reste sur la page du formulaire pour finir de le compléter

3. Enchaînement d'exception

- 3.1. Le responsable « clique » sur annuler
 - 3.1.1. Le formulaire se ferme après une demande de confirmation

C. Exigences particulières

Aucunes exigences particulières à déclarer.

D. Pré-conditions

- L'utilisateur doit être authentifié en tant que responsable ou administrateur de l'application.

E. Post-conditions

- L'utilisateur a bien été ajouté à la base de données.

F. Liste des acteurs participants

1. Acteurs principaux

- Un administrateur / responsable du système.
- Le système de gestion des utilisateurs.

Cas d'utilisation : Supprimer un utilisateur

A. Description

Ce cas d'utilisation concerne la suppression d'un utilisateur, avec pour objectif de le retirer de la base de données des utilisateurs.

B. Enchaînements d'évènements

1. Enchaînement de base

- 1.1. Le système affiche une liste des utilisateurs
- 1.2. Le responsable sélectionne un utilisateur
- 1.3. Le responsable clique sur supprimer
- 1.4. Le responsable confirme la suppression
- 1.5. Le Système d'information supprime l'utilisateur
- 1.6. Le Système d'information notifie le responsable du succès de la suppression

2. Enchaînement alternatif

- 2.1. Enchaînement alternatif 1 : A l'étape 1.5 : Le responsable annule la suppression
 - 2.1.1. Le responsable reste sur la page des utilisateurs et l'utilisateur n'est pas supprimé

3. Enchaînement d'exception

- 3.1. L'utilisateur ne peut pas être supprimé car il a un emprunt en cours (en partant du principe qu'un utilisateur peut aussi être client)
 - 3.1.1. Le Système renvoie un message d'erreur et la suppression est annulée

C. Exigences particulières

Aucunes exigences particulières à déclarer.

D. Pré-conditions

- L'utilisateur doit être authentifié en tant que responsable ou administrateur de l'application.

E. Post-conditions

- L'utilisateur a bien été supprimé de la base de données.

F. Liste des acteurs participants

1. Acteurs principaux

- Un administrateur / responsable du système.
- Le système de gestion des utilisateurs.

Cas d'utilisation : Ajout de matériel

A. Description

Ce cas d'utilisation concerne l'ajout de matériel dans l'objectif d'ajouter à la base de données différents matériels, pour ensuite pouvoir le(s) prêter aux utilisateurs.

B. Enchaînements d'évènements

1. Enchaînement de base

- 1.1 Une fois sur la page d'ajout, le responsable de matériel sélectionne le type du matériel
- 1.2 Le responsable matériel choisi le nombre d'exemplaire du matériel
- 1.3 Le responsable matériel doit remplir tous les champs d'un formulaire (qui diffère en fonction du type)
- 1.4 Le responsable matériel « click » sur valider
- 1.5 Le Système d'information reçoit le formulaire
- 1.6 Le Système d'information ajoute le(s) nouveau(x) matériel(s) à son inventaire
- 1.7 Envoi que l'ajout a bien été effectué par le Système d'information au responsable matériel

2. Enchaînement alternatif

- 2.1. Enchaînement alternatif 1 : Le formulaire n'est pas rempli ou incomplet
 - 2.1.1. Le système d'information envoie un message d'erreur lors de l'ajout
 - 2.1.2. Le responsable matériel reste sur la page du formulaire pour finir de le compléter
- 2.2. Enchaînement alternatif 2 : Le responsable de matériel « click » sur annuler
 - 2.2.1. Le formulaire se ferme après une demande de confirmation

3. Enchaînement d'exception

- 3.1. Enchaînement d'exception 1 : Le Système d'information n'arrive pas à ajouter de nouveaux matériels dans la base de données
 - 3.1.1. Le système d'information envoie un message d'erreur.

C. Exigences particulières

Aucunes exigences particulières à déclarer.

D. Pré-conditions

- L'utilisateur doit être authentifié en tant qu'administrateur du système de gestion de matériel.

E. Post-conditions

- Le(s) matériel(s) doivent bien être ajouté(s) à la base de données et bien ordonné(s)

F. Liste des acteurs participants

1. Acteurs principaux

- Un administrateur du système de gestion du matériel
- Le système de gestion du matériel

Cas d'utilisation : Suppression de matériel

A. Description

Ce cas d'utilisation concerne la suppression de matériel dans l'objectif d'enlever du stock et de la base de données différents matériels. Pour ensuite que le prêt aux utilisateurs ne soit plus possible.

B. Enchaînements d'évènements

1. Enchaînement de base

- 1.1 Recherche de matériel et choix de celui que le responsable de matériel veut supprimer
- 1.2 Le responsable matériel choisi la suppression de matériel
- 1.3 Une description du matériel est envoyée avec une demande de confirmation
- 1.4 Le responsable matériel « click » sur valider
- 1.5 Le Système d'information reçoit le(s) matériel(s) à supprimer
- 1.6 Le Système d'information fait passer le matériel à l'état « supprimé »
- 1.7 Envoi que la suppression a bien été effectué par le Système d'information au responsable matériel

2. Enchaînement alternatif

- 2.1. Enchaînement alternatif 1 : Le responsable de matériel « click » sur annuler
 - 2.1.1. Le formulaire se ferme après une demande de confirmation

3. Enchaînement d'exception

- 3.1. Enchaînement d'exception 1 : Le Système d'information n'arrive pas à supprimer le(s) matériel(s)
 - 3.1.1. Le système d'information envoie un message d'erreur

C. Exigences particulières

Aucunes exigences particulières à déclarer.

D. Pré-conditions

- L'utilisateur doit être authentifié en tant qu'administrateur du système de gestion de matériel.
- Le(s) matériel(s) doivent exister au préalable dans la base de données.

E. Post-conditions

- L'état de(s) matériel(s) doivent bien être passé à l'état « supprimé »

F. Liste des acteurs participants

1. Acteurs principaux

- Un administrateur du système de gestion du matériel
- Le système de gestion du matériel

Cas d'utilisation : rechercher matériel

A. Description

Ce cas d'utilisation concerne la recherche d'un matériel dans l'objectif d'obtenir des informations sur un ou plusieurs matériels, de le(s) modifier ou bien de le(s) supprimer.

B. Enchaînements d'évènements

1. Enchaînement de base

- 1.1 L'utilisateur saisit un nom de matériel à rechercher
- 1.2 L'utilisateur choisit les filtres désirés
- 1.3 L'utilisateur clique sur « rechercher »
- 1.4 Le système de gestion du matériel récupère le nom recherché ainsi que les filtres désirés puis il effectue la recherche.
- 1.5 Le système de gestion du matériel affiche les résultats de la recherche

2. Enchaînement alternatif

- 2.1 Enchaînement alternatif 1 : L'utilisateur ne saisit rien et effectue la recherche
 - 2.1.1 Le système de gestion retourne tous les matériels existants.
- 2.2 Enchaînement alternatif 2 : Aucun résultat n'est retourné par le SI
 - 2.2.1 Le SI affiche un message indiquant à l'utilisateur qu'aucun résultat ne correspond à la recherche.

3. Enchaînement d'exception

Aucun enchaînement d'exception à déclarer.

C. Exigences particulières

Aucune exigence particulière à déclarer.

D. Pré-conditions

- L'utilisateur doit être authentifié en tant qu'administrateur du système de gestion de matériel.
- La base de données doit contenir la liste des matériels.

E. Post-conditions

- La liste des résultats retournée doit correspondre à la recherche effectuée par l'utilisateur.

F. Liste des acteurs participants

1. Acteurs principaux

- Un administrateur du système de gestion du matériel
- Le système de gestion du matériel

Cas d'utilisation : modifier matériel

A. Description

Ce cas d'utilisation permet de modifier un matériel. La recherche du matériel a été inclus dans la séquentialité des actions.

B. Enchaînements d'évènements

1. Enchaînement de base

- 1.1 Rechercher matériel
 - 1.1.1 Le système d'information retourne la liste des résultats
 - 1.1.2 L'utilisateur choisit le matériel qu'il veut modifier
- 1.2 Modification du matériel
 - 1.2.1 L'utilisateur modifie le nom du matériel si voulu.
 - 1.2.2 Pour chaque caractéristique l'utilisateur modifie la caractéristique en question.
- 1.3 Cliquer sur «enregistrer modification »
- 1.4 Enregistrer les modifications
- 1.5 Retour automatique au menu

2. Enchaînement alternatif

- 2.1 Enchaînement alternatif 1 : l'utilisateur quitte la page de modification
 - 2.1.1 Toutes les modifications faites aux points 1.2 ne sont pas prises en compte.
- 2.2 Enchaînement alternatif 2 : l'utilisateur annule la modification
 - 2.2.1 Toutes les modifications faites aux points 1.2 ne sont pas prises en compte.
 - 2.2.2 Retour automatique au menu

3. Enchaînement d'exception

- 3.1. Enchaînement d'exception 1 : l'utilisateur ne saisit pas tous les champs
 - 3.1.1. Au point 1.3, un message d'erreur s'affiche pour indiquer à l'utilisateur que certains champs sont invalides
 - 3.1.2. Encadrer en rouge les champs manquants
 - 3.1.3. Ne pas effectuer l'enregistrement

C. Exigences particulières

Une modification concerne un matériel à la fois. On ne peut pas modifier plusieurs matériels en même temps.

D. Pré-conditions

- L'utilisateur doit être authentifié en tant qu'administrateur du système de gestion du matériel.
- Le matériel doit exister.

E. Post-conditions

- Les modifications faites doivent être prises en compte et sauvegardées dans la base de données.
- Les données modifiées doivent rester cohérentes.

F. Liste des acteurs participants

1. Acteurs principaux

- Un administrateur du système de gestion du matériel
- Le système de gestion du matériel

Cas d'utilisation : Nouveau prêt

A. Description

Ce cas d'utilisation correspond à la création d'un nouvel emprunt / prêt de matériel pour une période et une personne donnée.

B. Enchaînements d'évènements

1. Enchaînement de base

- 1.1. Le système affiche le formulaire de création de prêt.
- 1.2. L'utilisateur (Gestionnaire prêts) choisit le type de matériel
- 1.3. L'utilisateur choisit la période d'emprunt / date début et fin.
- 1.4. Le SI affiche alors la liste du matériel disponible pour cette période.
- 1.5. L'utilisateur choisit alors le matériel à emprunter.
- 1.6. L'utilisateur passe à la seconde page uniquement si les champs sont tous remplit.
- 1.7. L'utilisateur peut filtrer les utilisateurs par nom, prénom ou les deux à la fois
- 1.8. Le SI affiche les utilisateurs en fonction des paramètres de recherche.
- 1.9. L'utilisateur sélectionne le client dans la liste déroulante.
- 1.10. Passage à la page suivante uniquement si le client a été choisi
- 1.11. Le SI affiche un récapitulatif de l'emprunt.
- 1.12. L'utilisateur valide
- 1.13. Le SI propose un popup de confirmation
- 1.14. L'utilisateur enregistre le prêt/emprunt.

2. Enchaînement alternatif

- 2.1. Enchainement alternatif 1 : A l'étape 1.4, si aucun matériel n'est disponible sur cette période

2.1.1. Le SI affiche un message et l'utilisateur peut changer ses dates ou annuler la demande de prêt.

- 2.2. Enchainement alternatif 2 : A l'étape 1.8, si l'utilisateur ne saisit rien dans la recherche des clients

2.2.1. Le SI affiche tous les utilisateurs

- 2.3. Enchainement alternatif 3 : À tout moment, l'utilisateur clique sur le bouton annuler.

2.3.1. L'emprunt n'est pas enregistré, retour à l'accueil.

- 2.4. Enchainement alternatif 4 : Sur la page 2(étape 6 à 9) L'utilisateur clique sur le bouton page précédente.

2.4.1. L'utilisateur retourne sur la page 1 (étapes 1 à 5)

- 2.5. Enchainement alternatif 5 : Sur la page 3(étape 11) L'utilisateur clique sur le bouton page précédente.

2.5.1. L'utilisateur retourne sur la page 2 (étapes 6 à 9)

3. Enchaînement d'exception

Aucun enchaînement d'exception à déclarer.

C. Exigences particulières

Aucune exigence particulière à déclarer.

D. Pré-conditions

- L'utilisateur doit être connecté en tant que gestionnaire de prêt.
- La base de données doit contenir la liste des matériels et des utilisateurs.

E. Post-conditions

- La liste du matériel affiché doit correspondre uniquement à du matériel disponible sur la période sélectionnée
- La liste des clients retournée doit correspondre à la recherche effectuée par l'utilisateur.
- La date de fin d'emprunt doit être supérieure à la date de début d'emprunt.

F. Liste des acteurs participants

1. Acteurs principaux

- Un utilisateur du système de gestion de prêts
- Le système de gestion de prêts

2. Acteurs secondaires

- Le client

Cas d'utilisation : Modifier prêt

A. Description

Ce cas d'utilisation correspond à la modification d'un emprunt / prêt de matériel pour une période et une personne donnée.

B. Enchaînements d'évènements

1. Enchaînement de base

- 1.1. Le système affiche le formulaire de modification de prêt.
- 1.2. L'utilisateur peut modifier la date de début (si celle-ci n'est pas déjà passée)
- 1.3. L'utilisateur peut modifier la date de fin.
- 1.4. Le SI affiche alors si le matériel est toujours disponible pour cette période.
- 1.5. L'utilisateur enregistre la modification du prêt/emprunt.

2. Enchaînement alternatif

- 2.1. Enchainement alternatif 1 : A l'étape 1.4, si aucun matériel n'est disponible sur cette période
 - 2.1.1. Le SI affiche un message et l'utilisateur peut changer ses dates ou annuler la demande de prêt.
- 2.2. Enchainement alternatif 2 : À tout moment, l'utilisateur clique sur le bouton annuler.
 - 2.2.1. La modification n'est pas enregistrée, retour à l'accueil.

3. Enchaînement d'exception

Aucun enchaînement d'exception à déclarer.

C. Exigences particulières

Modifier un emprunt exige qu'un emprunt existe dans l'état en cours.

D. Pré-conditions

- L'utilisateur doit être connecté en tant que gestionnaire de prêt.
- La base de données doit contenir la liste des matériels et des utilisateurs.
- La date de début d'emprunt ne doit pas être dépassé pour pouvoir modifier la date de début d'emprunt

E. Post-conditions

- Le matériel sélectionné doit demeurer disponible sur la nouvelle période sélectionnée
- La date de fin d'emprunt doit être supérieure à la date de début d'emprunt.

F. Liste des acteurs participants

1. Acteurs principaux
 - Un utilisateur du système de gestion de prêts

- Le système de gestion de prêts
2. Acteurs secondaires
- Le client

Cas d'utilisation : Consulter emprunts en cours

A. Description

Ce cas d'utilisation concerne la consultation des emprunts en cours dans l'objectif de superviser la gestion des emprunts en cours.

B. Enchaînements d'évènements

1. Enchaînement de base

- 1.1 Le système affiche la liste des utilisateurs
- 1.2 L'utilisateur sélectionne un utilisateur
- 1.3 Le système affiche les emprunts en cours pour l'utilisateur concerné

2. Enchaînement alternatif

- 2.1. Enchaînement alternatif 1 : Aucun résultat n'est retourné par le SI
 - 2.3.1 Le SI affiche un message indiquant à l'utilisateur qu'aucun emprunt est en cours pour cet utilisateur

3. Enchaînement d'exception

Aucun enchaînement d'exception à déclarer.

C. Exigences particulières

Aucune exigence particulière à déclarer.

D. Pré-conditions

- L'utilisateur doit être authentifié en tant qu'utilisateur du système de gestion de matériel.
- La base de données doit contenir la liste des matériels et des utilisateurs.

E. Post-conditions

Aucune exigence particulière à déclarer.

F. Liste des acteurs participants

1. Acteurs principaux

- Un utilisateur du système de gestion du matériel
- Le système de gestion du matériel

Cas d'utilisation : Consulter utilisateurs en retard

A. Description

Ce cas d'utilisation concerne la consultation des utilisateurs en retard dans l'objectif de superviser la gestion des emprunts en retard.

B. Enchaînements d'évènements

1. Enchaînement de base

- 1.1 Le système récupère la liste des matériels non rendus après date de fin d'emprunt
- 1.2 Le système affiche les utilisateurs liés à la liste récupérée à l'étape 1.1

2. Enchaînement alternatif

- 2.1 Enchaînement alternatif 1 : Aucun résultat n'est retourné par le SI

- 2.3.1 Le SI affiche un message indiquant à l'utilisateur qu'aucun emprunt non rendu après de fin d'emprunt n'a été trouvé

3. Enchaînement d'exception

Aucun enchaînement d'exception à déclarer.

C. Exigences particulières

Aucune exigence particulière à déclarer.

D. Pré-conditions

- L'utilisateur doit être authentifié en tant que gestionnaire du système de gestion de matériel.
- La base de données doit contenir la liste des matériels et des utilisateurs.

E. Post-conditions

Aucune exigence particulière à déclarer.

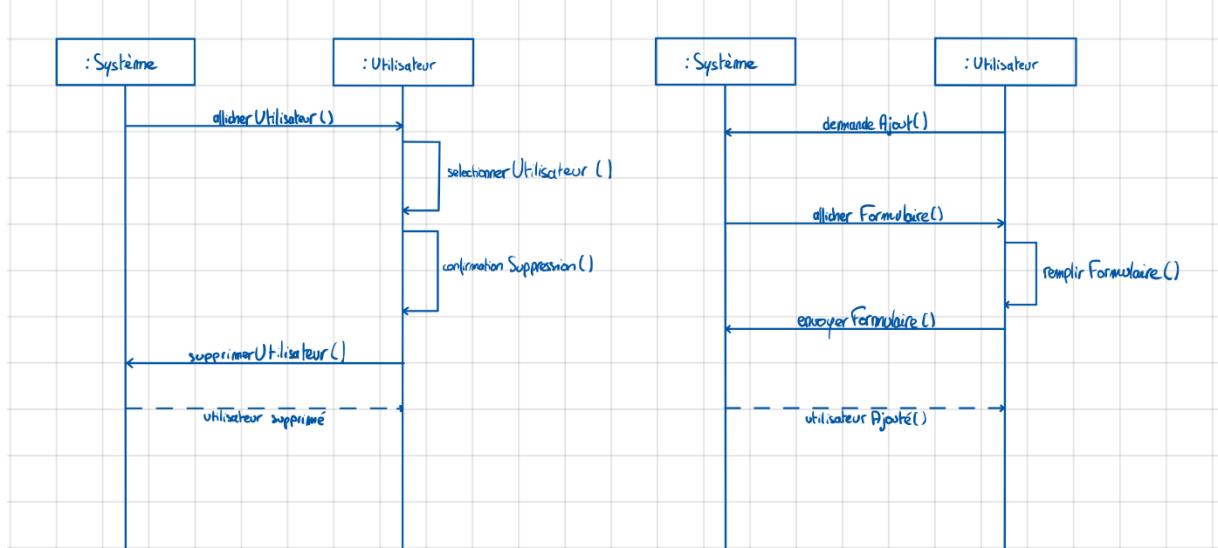
F. Liste des acteurs participants

1. Acteurs principaux

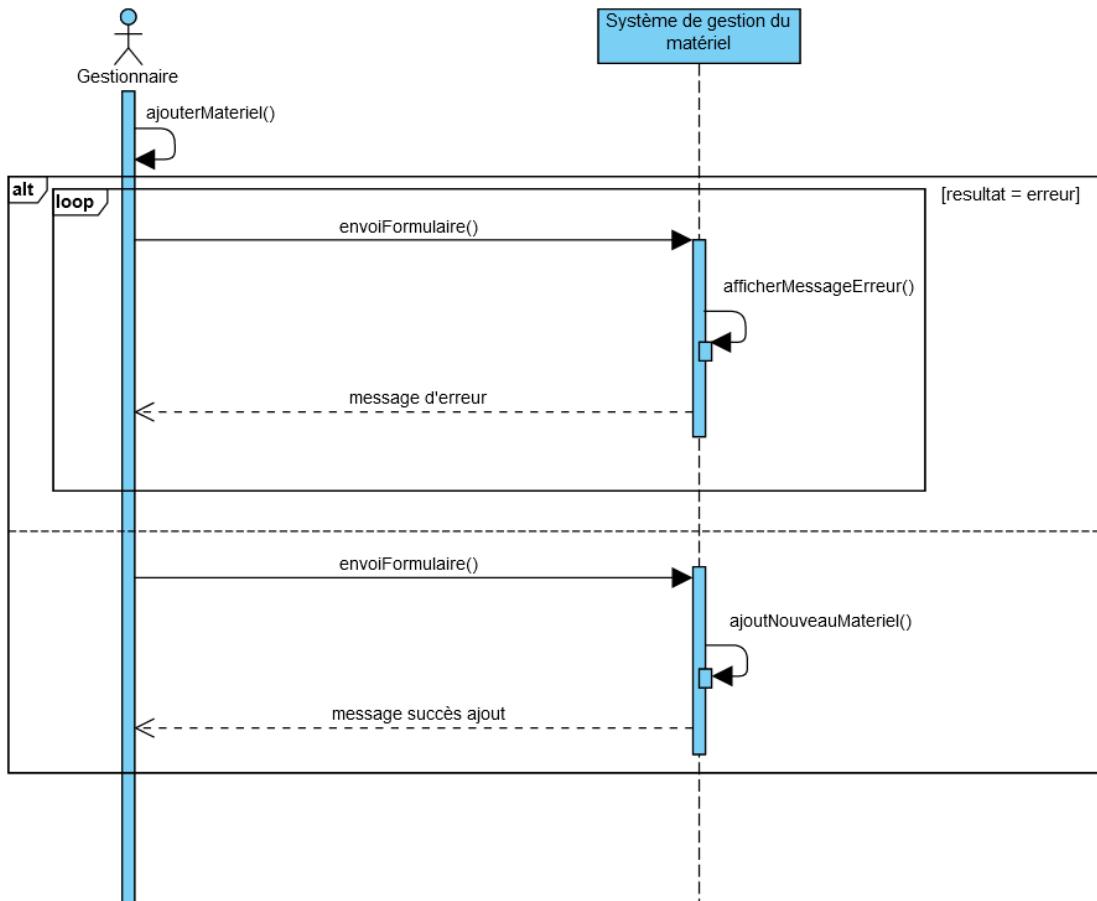
- Un utilisateur du système de gestion du matériel
- Le système de gestion du matériel

e. Diagrammes de séquences

Supprimer un utilisateur / Ajouter un utilisateur



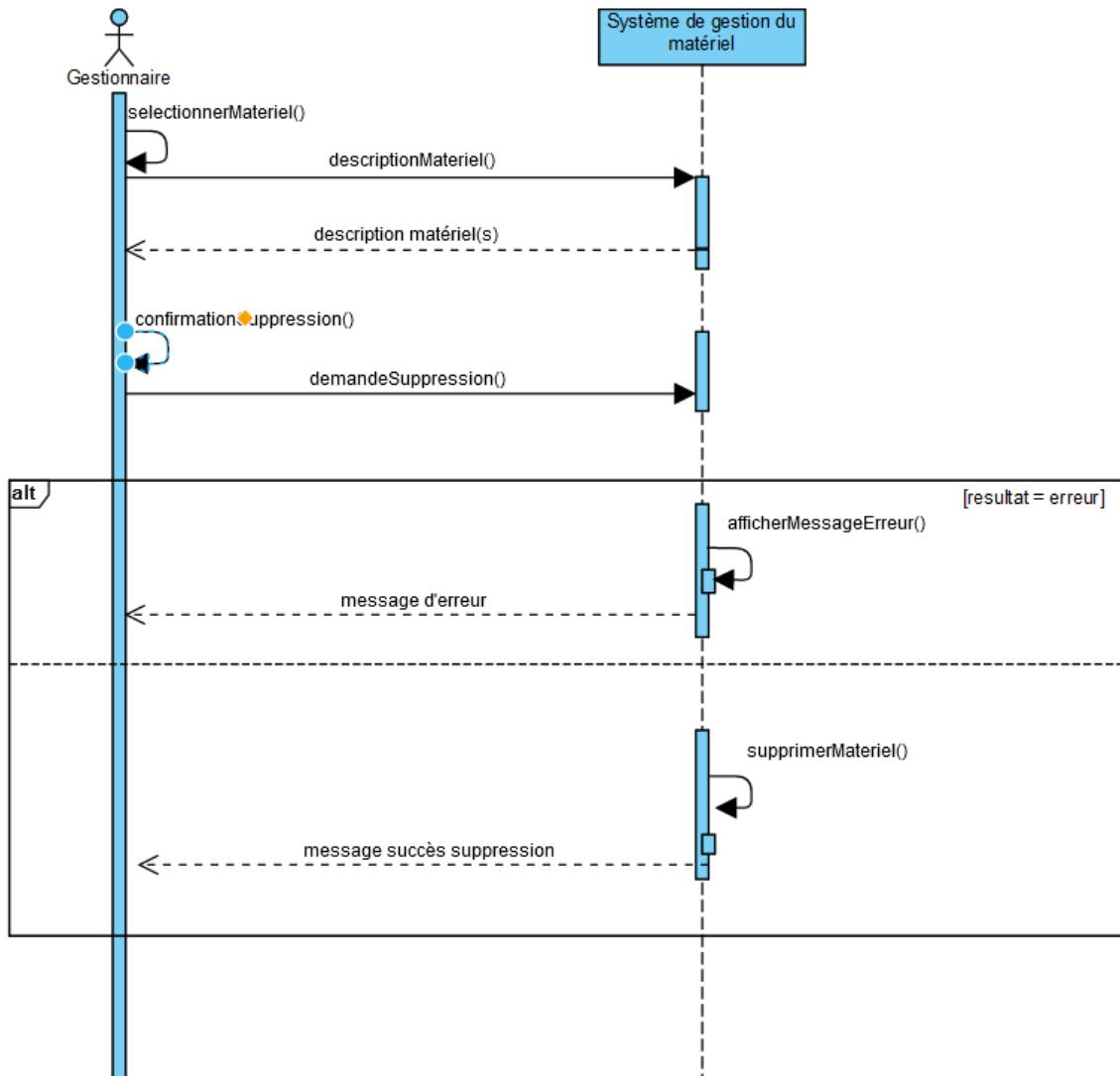
Ajouter matériel



Justification :

Le diagramme de séquence du cas ajouter un matériel propose un cas alternatif dans le cas où une erreur est détectée dans le résultat. Cela peut par exemple être un champ obligatoire du formulaire qui est vide lors de la validation.

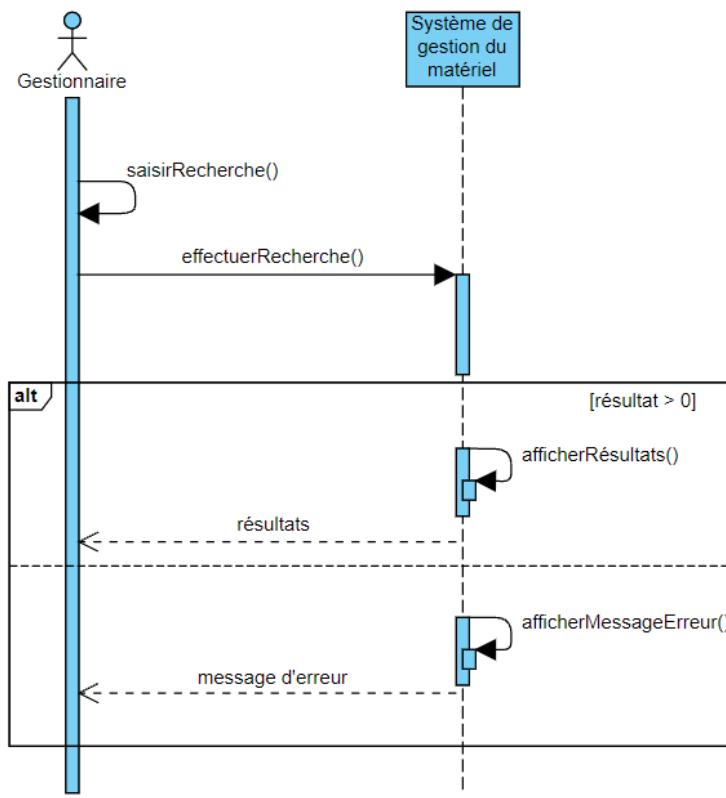
Supprimer matériel



Justification :

De la même manière que pour ajouter un matériel, pour supprimer un matériel nous avons choisi de mettre un cas alternatif afin de guider l'utilisateur dans son parcours.

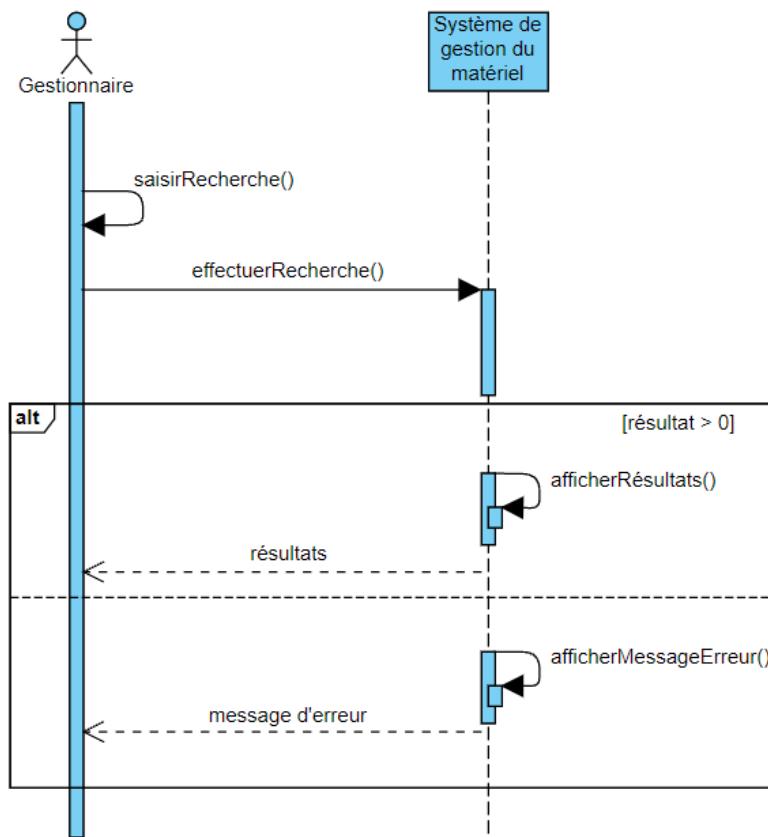
Rechercher matériel



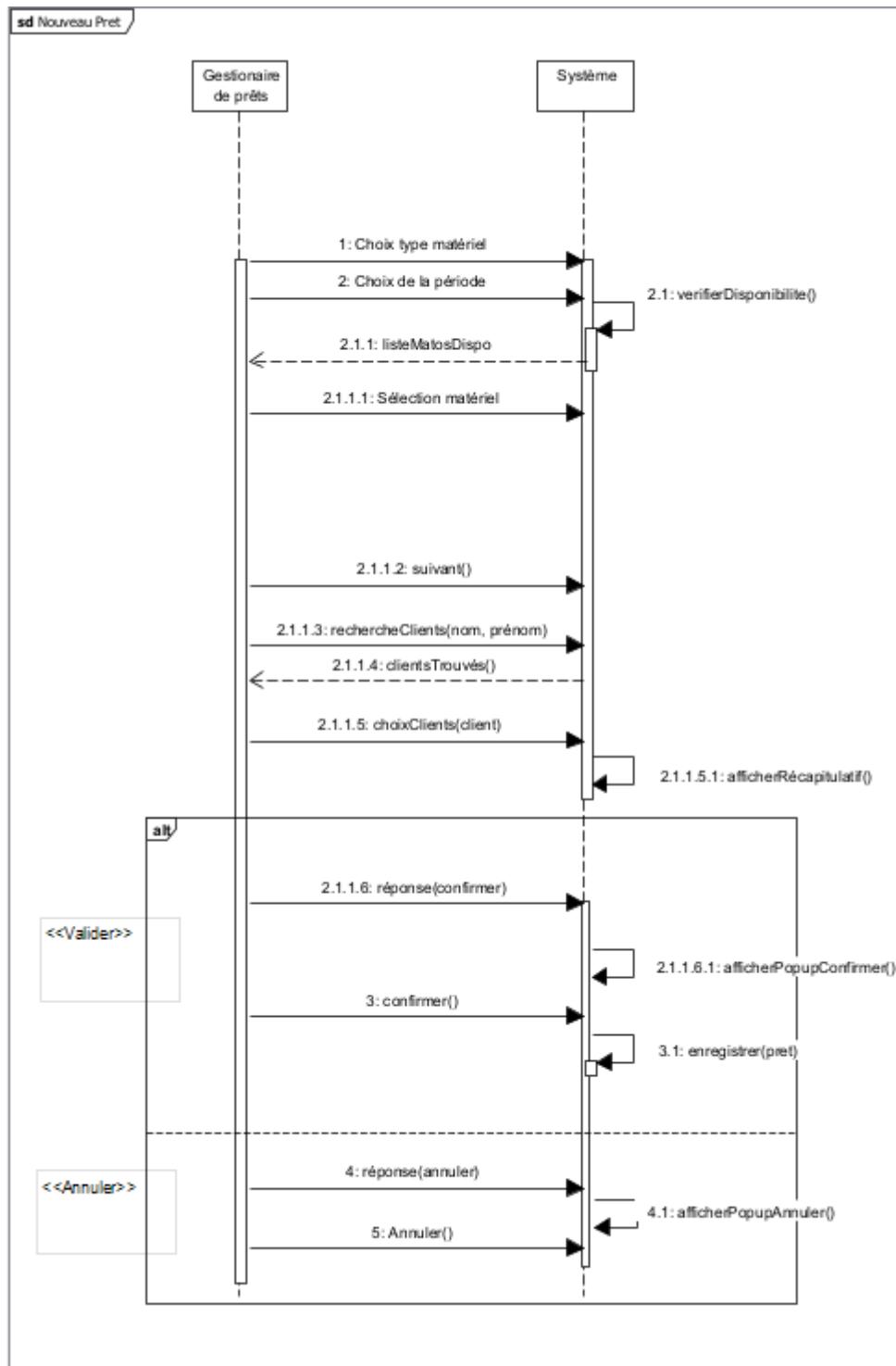
Justification :

Lorsqu'on recherche un matériel on a un cas alternatif, on affiche les résultats s'il y en a sinon on affiche un message d'erreur.

Modifier matériel



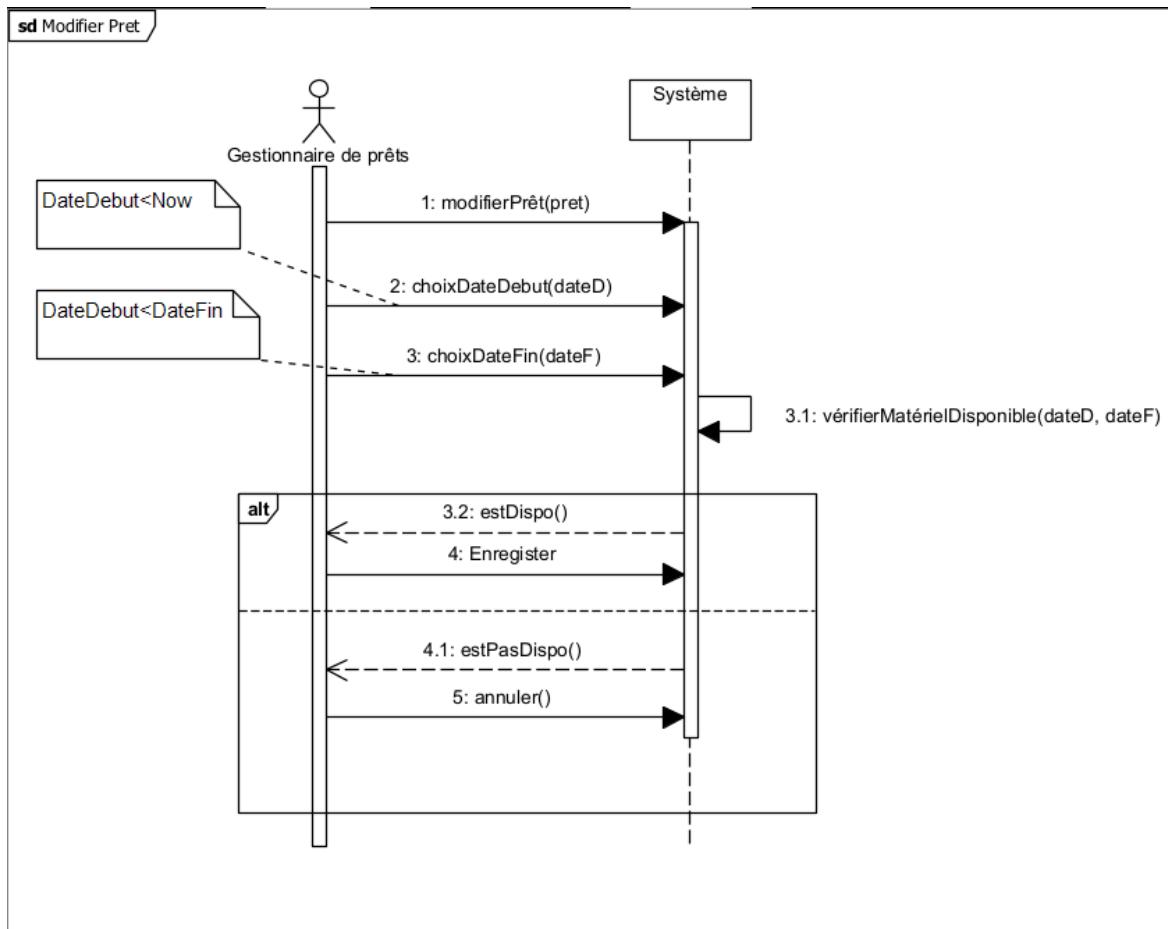
Nouveau prêt



Justification :

On a décidé de faire deux cas alternatifs à savoir valider et annuler. Quand on valide on affiche un message pour confirmer la validation puis on enregistre le nouveau prêt. Lors d'une annulation de la même manière qu'avant on affiche une popup pour confirmer.

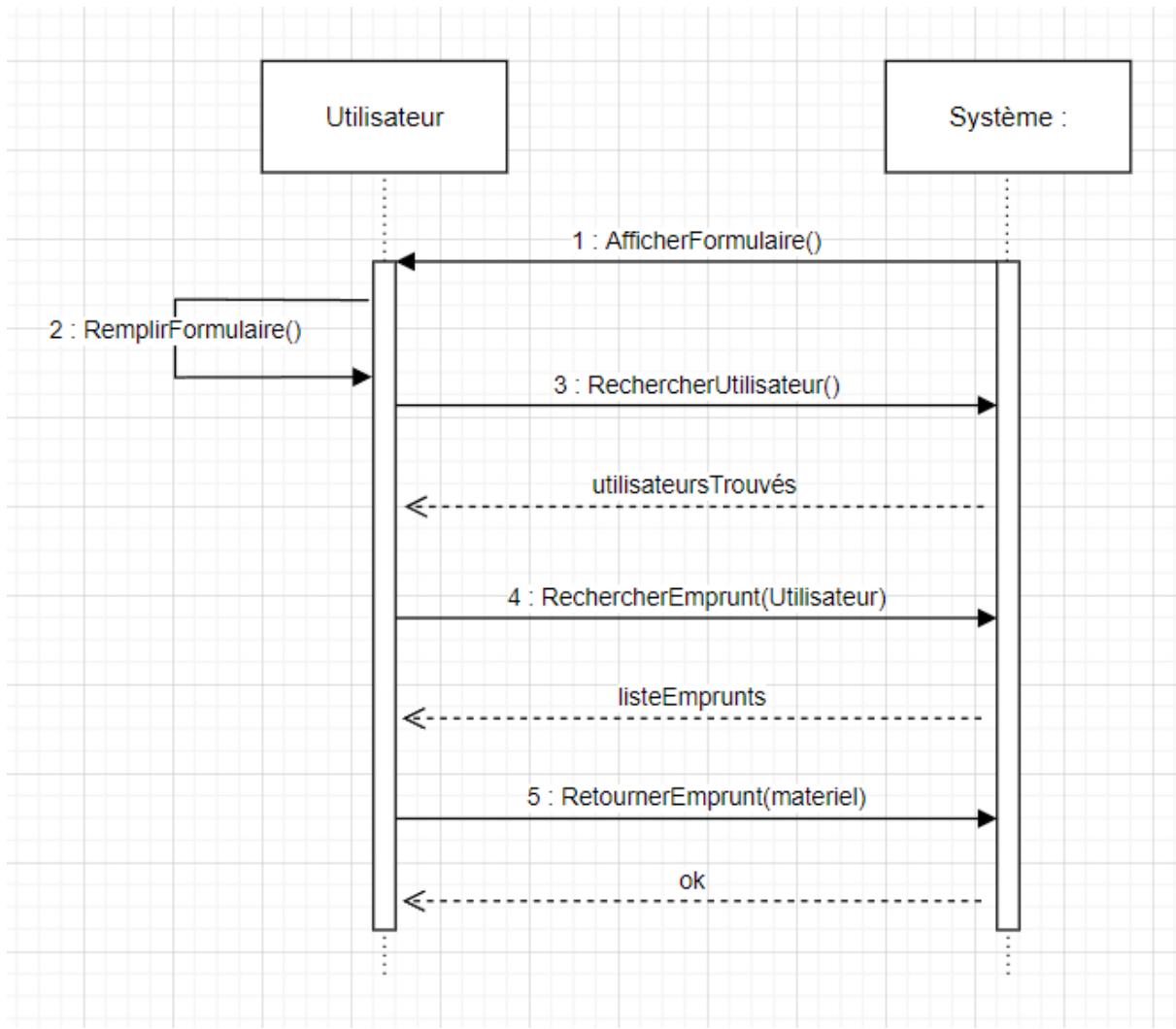
Modifier prêt



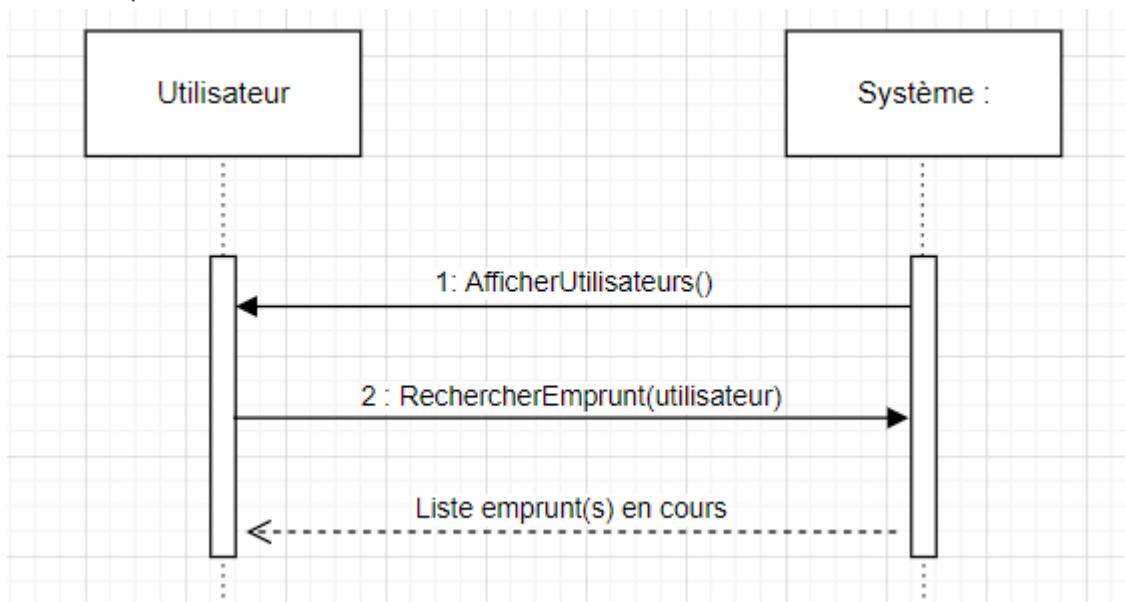
Justification :

En cohérence avec les schémas précédents on vérifie la condition d'une modification notamment au niveau des dates.

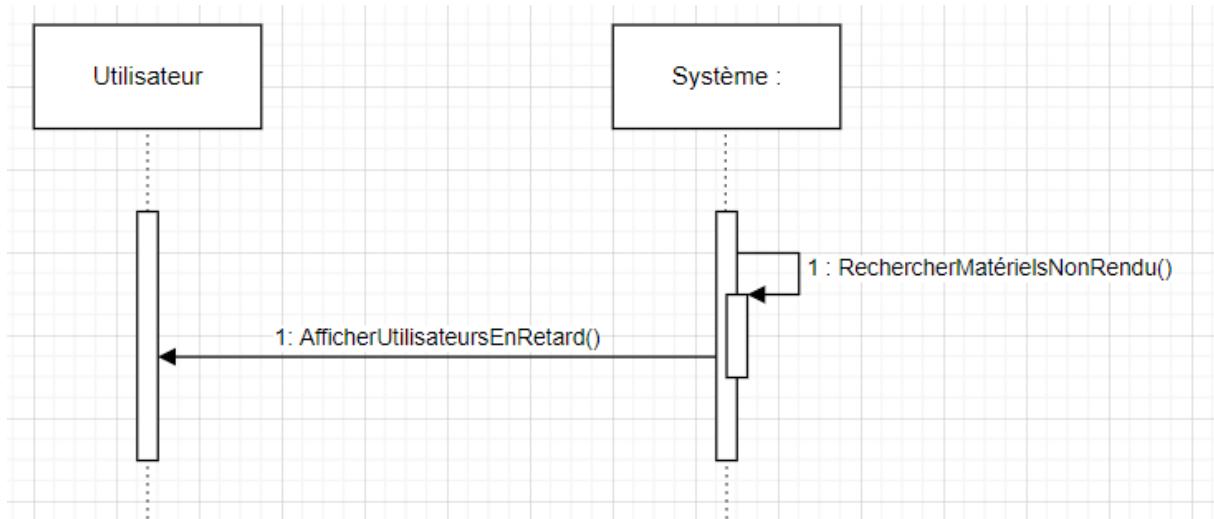
Retourner prêt



Lister les prêts en cours



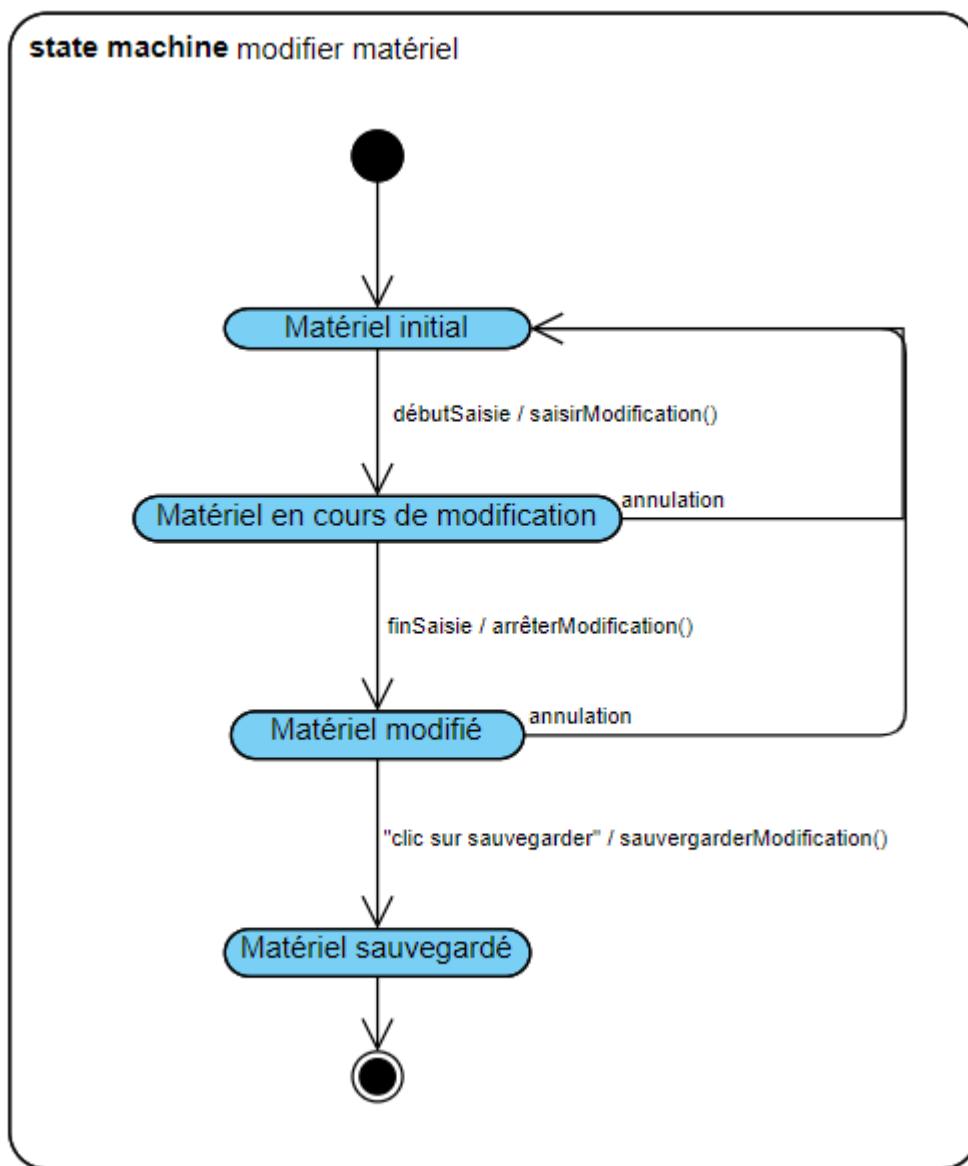
Lister les prêts en retard



Justification :

Pour tous les schémas sur les prêts, “utilisateur” veut dire gestionnaire de prêt.

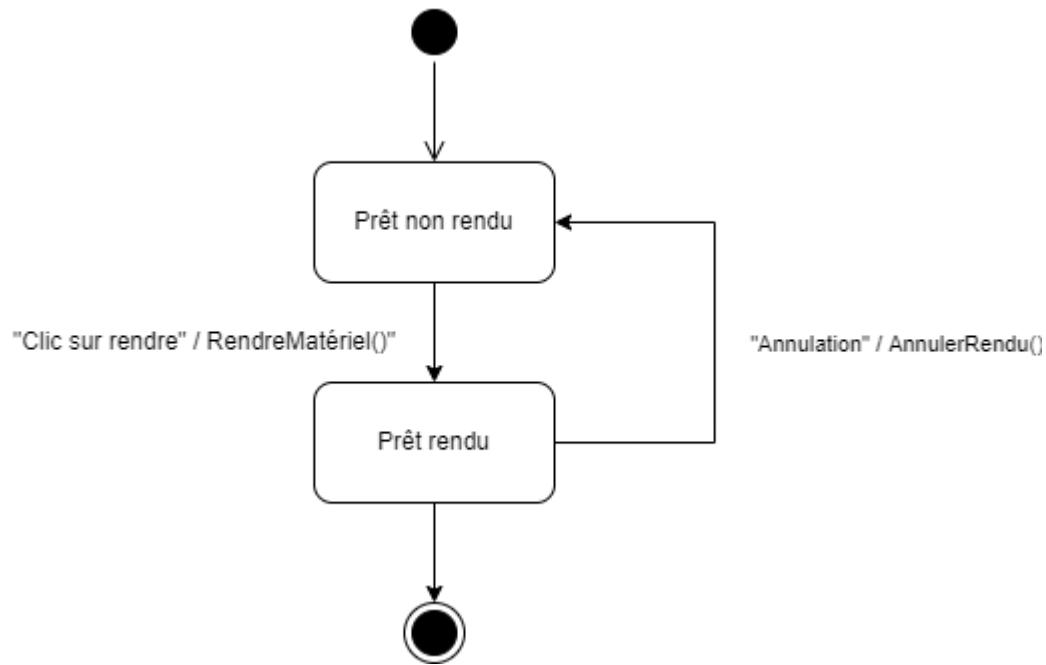
f. Diagramme d'état transition



Justification :

Le diagramme d'état transition pour le cas Modifier matériel consiste à un état initial. Une fois qu'on le modifie, soit on annule toute modification et le matériel revient à son état initial, soit on valide et le matériel change d'état pour être modifié et sauvegardé.

state machine rendre prêt



Justification :

Ici, on rend un matériel dans un état initial “non-rendu”. Une fois avoir cliqué sur rendre, le prêt change d’état en “rendu” sauf si on annule durant le message de confirmation.

g. Diagramme de classe

<https://online.visual-paradigm.com/community/share/diagramme-de-classe-po46xd5lv>

Justification :

On a décidé de faire deux classes principales qui stockent les utilisateurs et les matériels. Le lien entre ces deux classes est une classe qui représente les emprunts. Chaque type de matériel est une sous-classe de matériel, de même pour la table utilisateur. On a choisi de travailler avec des classes “principales” afin de rendre la maintenance plus facile puisque dans le futur on pourra ajouter de nouveaux matériels sans devoir modifier le lien avec les utilisateurs via les emprunts.

4. Implémentation

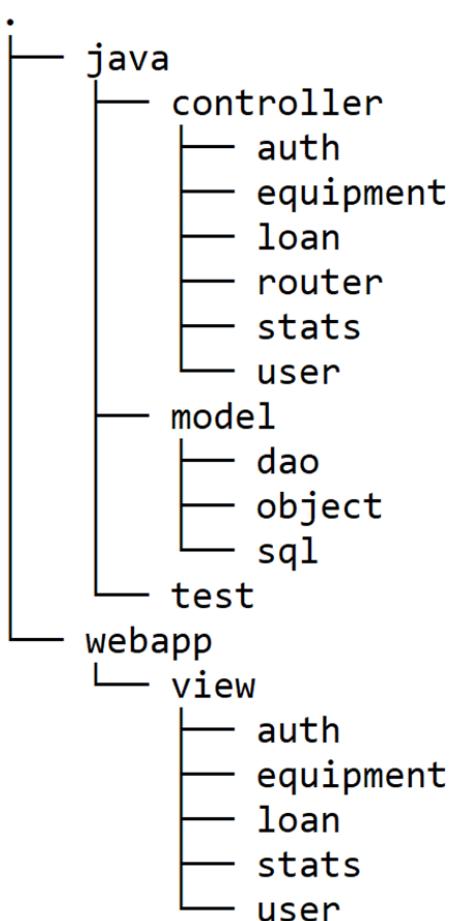
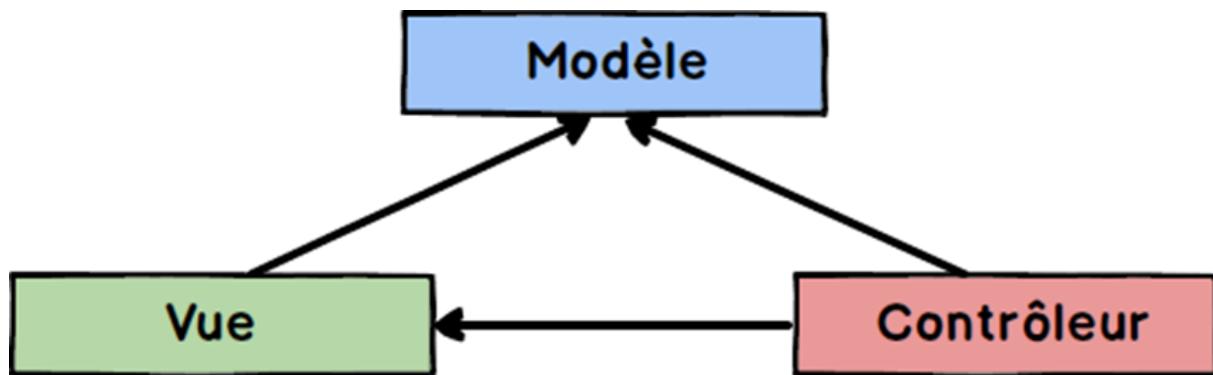
a. Architecture

MVC

L'ensemble de l'application fonctionne sous un pattern MVC, c'est-à-dire que l'interface utilisateur, la vue affiche les informations du modèle.

La vue détient uniquement ce que le contrôleur lui envoie.

C'est le contrôleur qui manipule les informations du modèle.



L'arborescence du projet est relativement simple :

- Le répertoire /java est composé de toutes les classes et interfaces nécessaires au développement backend de l'application dont le modèle.
- Le répertoire /webapp contient les ressources telles que les vues ou les images.

DAO

Le modèle DAO permet aux contrôleurs Java de faire abstraction du modèle SQL. Les contrôleurs vont simplement demander aux classes DAO de stocker des données et de les récupérer.

Dans notre cas, chaque type d'équipement dispose d'un DAO propre qui correspond à ses caractéristiques.

Navigation

Un servlet permet de faire le lien entre les URL et leur vue et contrôleur associé.

Chaque contrôleur dispose également d'une méthode doGet qui vérifie que l'utilisateur qui demande à voir la page dispose des droits nécessaires pour avoir accès à la page.

La méthode doPost est optionnelle en fonction des pages, elle permet d'envoyer des informations saisies par l'utilisateur dans différents formulaires aux contrôleurs correspondants et donc d'effectuer les actions nécessaires.

b. Fonctions implémentées

Authentification

Chaque utilisateur du site doit se connecter afin d'avoir accès aux différentes fonctionnalités en fonction des restrictions mises en place.

Pour ce faire, plusieurs rôles existent :

- L'utilisateur lambda peut accéder à ses emprunts en cours sur son profil.
- Le rôle EQUIPEMENT_ADMIN permet de modifier la liste des matériels disponibles (Ajout, modification et suppression).
- Le rôle LOAN_ADMIN permet la création de prêts et leur retour.
- Le rôle ADMIN permet d'ajouter des utilisateurs et dispose de tous les rôles précédents.

Gestion des utilisateurs

L'ADMIN peut créer des utilisateurs, en les ajoutant en base de données et également leur donner un rôle si besoin.

Gestion du matériel

Plusieurs pages sont disponibles afin d'ajouter, lister, modifier et supprimer du matériels

Gestion des prêts

Le gestionnaire des prêts peut sélectionner le matériel et une date d'emprunt afin d'assigner un prêt à un utilisateur.

Si le prêt n'est pas marqué comme rendu, avant la date de fin, il sera déclaré en retard et visible dans la liste des prêts en retard.

c. Fonctions non implémenté / en cours d'implémentation

Mails

Un système d'envois de mail pourrait être bénéfique.

Il pourrait permettre d'avoir accès à une fonctionnalité de mot de passe oublié avec réinitialisation par mail.

Ou simplement pour notifier l'utilisateur que la date de fin d'emprunt arrive ou qu'il est en retard.

d. Outils utilisés

Modélisation UML - Visual Paradigm

Maquettes graphiques - Figma

Développement - Eclipse JEE et serveur Tomcat

Base de données : MySQL

Utilisation du pattern DAO

Pas de frameworks utilisé

Utilisation de librairies pour les vues : JQuery, Bootstrap (datepicker) et Chart.js

5. En Pratique

a. Accueil

La page d'accueil propose si l'on dispose des droits suffisant d'ajouter un emprunt, un matériel, ou un utilisateur.

La barre de navigation en haut de l'écran permet de naviguer entre les différentes pages.

Chaque bouton envoie une action qui sera traitée par un DoPost.



b. Les profil utilisateurs

La liste des utilisateurs permet de rechercher un utilisateur et de connaître ses emprunts en cours, en retard et même à venir.

UTILISATEURS

	Prénom et/ou Nom	<input type="checkbox"/>	
	angela.merkel@gmail.com		
	emmanuel.macron@gmail.com		
	root@gmail.com		
	vladimir.putin@gmail.com		

La page de recherche affiche tous les utilisateurs possibles en effectuant une requête à la base de données à l'ouverture de la page. Afin de prendre en compte le nom et/ou le prénom de l'utilisateur que l'on recherche du JavaScript intervient et filtre en fonction des caractères saisis dans le champ de recherche.

Une fois un utilisateur choisi, on peut voir ses informations personnelles et également ses emprunts en cours, à venir, et en retard.

UTILISATEUR

Angela Merkel
secrétaire

Login : Deutsch92
Adresse : 1 root de Germany
Téléphone : 0102030405
Mail : angela.merkel@gmail.com
N° Bureau : 2
Profil : LOAN_ADMIN
Service : secrétariat

Emprunts

Projecteur - En cours [Restituer](#)

Fiat 500 rouge - À venir

Harley Davidson Sportster 2 pla - À venir

c. Gestion du matériel

L'utilisateur responsable du matériel dispose d'un accès à l'ajout de matériel. A l'aide de différents formulaires, il peut ajouter le bon type de matériel et toutes les informations nécessaires. Par

exemple, un véhicule dispose d'un kilométrage, d'une plaque d'immatriculation, etc. Un ordinateur est composé d'un sous matériel en base de donnée, le processeur.

La recherche d'équipements/matériels fonctionne similaire à la recherche d'utilisateurs.

ÉQUIPEMENTS

	Nom équipement	Type d'équipement
		Tout afficher ▼

VOITURES



MOTOS



Chacun des équipements disponible sur le site peut être modifié afin de mettre à jour les informations le concernant.

La modification permet à l'utilisateur de mettre à jour les champs dont il souhaite. L'envois du formulaire déclenche la méthode doPost qui récupère tout dans une Hashmap puis à l'aide du DAO on met à jour l'objet en question en base de données.

EQUIPMENT/VÉHICULE



Kilométrage : 1000
Kilométrage de renouvellement : 6000
Marque : Yamaha
Modèle : MT07
Immatriculation : AE-0X0-AA
État : Bon état
Vitesse maximale : 500
Puissance : 1200

d. Gestion des Emprunts / Prêts

Le responsable des emprunts peut associer un matériel à une période d'emprunt et un utilisateur. Cet emprunt est limité dans le temps et tout emprunt non rendu en temps est en heure sera signalé comme en retard.

Le matériel choisi pourra être emprunté uniquement si il est libre, l'outil de sélection de date empêche la sélection dans les cases ou le matériel est déjà emprunté.

Une fois l'emprunt saisi, on peut le retrouver dans la liste et ainsi obtenir plus d'informations sur l'emprunt.

EMPRUNTS

EMPRUNTS EN RETARD

Il n'y a pas d'emprunts en retard.

EMPRUNTS EN COURS

Utilisateur	Matériel	
angela.merkel@gmail.com	Projecteur	Détail
emmanuel.macron@gmail.com	Souris Logitech	Détail

EMPRUNTS À VENIR

Utilisateur	Matériel	
angela.merkel@gmail.com	Fiat 500 rouge	Détail
angela.merkel@gmail.com	Harley Davidson Sportser 2 pla	Détail

Les emprunts sont triés par types, en retard, en cours et à venir, on peut afficher plus de détails sur chacun des emprunts. L'affichage des informations d'un emprunt permet aussi de rendre le matériel emprunté.

INFORMATION SUR UN EMPRUNT

Date d'emprunt : 2021-11-21	-	Date de rendu : 2021-11-28
-----------------------------	---	----------------------------

 **Emmanuel Macron**

Mail : emmanuel.macron@gmail.com
Téléphone : 0102030405
Adresse : 1 root de Paris

 **Souris Logitech**

[Modifier](#) [Signaler comme rendu](#)

6. Annexes

a. Maquettes

Liens des maquettes sur figma :

<https://www.figma.com/file/3tkBDepEMjU6iN8EaQlqJ8/JEE?node-id=0%3A1>

b. Base de données

```

DROP TABLE IF EXISTS Loan;
DROP TABLE IF EXISTS SuitableVehicle;
DROP TABLE IF EXISTS
Computer;
DROP TABLE IF EXISTS Car;
DROP TABLE IF EXISTS Bike;
DROP TABLE IF EXISTS
Employee;
DROP TABLE IF EXISTS User;
DROP TABLE IF EXISTS Processor;
DROP TABLE IF EXISTS
GraphicCard;
DROP TABLE IF EXISTS ComputerAccessory;
DROP TABLE IF EXISTS
VehicleAccessory;
DROP TABLE IF EXISTS Vehicle;
DROP TABLE IF EXISTS Equipment;

-- #region User
CREATE TABLE User (
    mail VARCHAR(100) PRIMARY KEY ,
    name VARCHAR(30) NOT
NULL,
    firstname VARCHAR(30) NOT NULL,
    login VARCHAR(200) NOT NULL,
    password
VARCHAR(200) NOT NULL, -- add SHA function
    address VARCHAR(200),
    phoneNumber
VARCHAR(20),
    isReal BOOLEAN DEFAULT 1
);

CREATE TABLE Employee (
    mail VARCHAR(100)
PRIMARY KEY,
    employeeFunction VARCHAR(30) NOT NULL,
    employeeService VARCHAR(30) NOT
NULL,
    deskNumber INT,
    profil enum('ADMIN', 'EQUIPMENT_ADMIN', 'LOAN_ADMIN') NOT NULL,
    FOREIGN KEY (mail) REFERENCES User(mail)
);
-- #endregion User

-- #region
Equipment

CREATE TABLE Equipment (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(30)
NOT NULL,
    available BOOLEAN DEFAULT 1,
    imageURL VARCHAR(300),
    canBeLoaned BOOLEAN
DEFAULT 1
);

CREATE TABLE Processor (
    id INT PRIMARY KEY,
    name VARCHAR(30) NOT
NULL,
    brand VARCHAR(30) NOT NULL,
    numberOFCores INT,
    frequency FLOAT
);

CREATE
TABLE GraphicCard (
    id INT PRIMARY KEY,

```

```

        name VARCHAR(30) NOT NULL,
        brand VARCHAR(30)
NOT NULL ,
        frequency FLOAT
);

CREATE TABLE ComputerAccessory (
    id INT PRIMARY KEY,
    FOREIGN KEY (id) REFERENCES Equipment(id)
);

CREATE TABLE VehicleAccessory (
    id INT
PRIMARY KEY,
    FOREIGN KEY (id) REFERENCES Equipment(id)
);

CREATE TABLE Vehicle (
    id
INT PRIMARY KEY,
    kilometers INT NOT NULL,
    renewalKilometers INT NOT NULL,
    brand
VARCHAR(30) NOT NULL,
    model VARCHAR(30) NOT NULL,
    registrationNumber VARCHAR(30) NOT
NULL,
    state enum('GOOD', 'AVERAGE', 'BAD'),
    maxSpeed INT,
    numberOfspeeds INT,
    power
INT,
    FOREIGN KEY (id) REFERENCES Equipment(id)
);

CREATE TABLE Car (
    id INT PRIMARY
KEY,
    numberOfeats INT,
    FOREIGN KEY (id) REFERENCES Vehicle(id)
);

CREATE TABLE Bike
(
    id INT PRIMARY KEY,
    numberOfcylinders INT,
    FOREIGN KEY (id) REFERENCES
Vehicle(id)
);

CREATE TABLE SuitableVehicle (
    accessoryId INT ,
    vehicleId INT ,
    PRIMARY KEY (accessoryId, vehicleId),
    FOREIGN KEY (accessoryId) REFERENCES
VehicleAccessory(id),
    FOREIGN KEY (vehicleId) REFERENCES Vehicle(id)
);

CREATE TABLE
Computer (
    id INT PRIMARY KEY,
    brand VARCHAR(30) NOT NULL,
    model VARCHAR(30) NOT
NULL,
    serialNumber VARCHAR(30) NOT NULL,
    memorySize INT ,
    isLaptop BOOLEAN NOT NULL,
    screenSize INT ,
    purchaseDate DATE NOT NULL,
    renewalDate DATE NOT NULL,
    processorId

```

```

    INT NOT NULL,
    graphicCardId INT NOT NULL,
    FOREIGN KEY (id) REFERENCES Equipment(id),
    FOREIGN KEY (processorId) REFERENCES Processor(id),
    FOREIGN KEY (graphicCardId) REFERENCES GraphicCard(id)
);

-- #endregion Equipment

-- #region Loan

CREATE TABLE Loan (
    id INT PRIMARY KEY,
    equipmentId INT NOT NULL,
    userMail VARCHAR(100) NOT NULL,
    beginningDate DATE NOT NULL,
    endDate DATE NOT NULL,
    isOver BOOLEAN NOT NULL,
    FOREIGN
    KEY (equipmentId) REFERENCES Equipment(id),
    FOREIGN KEY (userMail) REFERENCES User(mail)
);

-- #endregion Loan

```

c. Dump de la base

```

INSERT INTO User (mail, name, firstname, login, password, address, phoneNumber) VALUES
("root@gmail.com", "Biden", "Joe", "root",
SHA1("toor"), "1 root des USA", "0102030405");
INSERT INTO User
(mail, name, firstname, login, password, address, phoneNumber) VALUES
("xi.jiping@gmail.com", "Jiping", "Xi", "xj",
SHA1("toor"), "1 root de Chine", "0102030405");
INSERT INTO User
(mail, name, firstname, login, password, address, phoneNumber) VALUES
("vladimir.putin@gmail.com", "Putin", "Vladimir",
"vlad", SHA1("toor"), "1 root de Mother land",
"0102030405");
INSERT INTO User (mail, name, firstname, login, password, address,
phoneNumber) VALUES ("emmanuel.macron@gmail.com", "Macron",
"Emmanuel", "EnMarche", SHA1("toor"), "1 root de
France", "0102030405");
INSERT INTO User (mail, name, firstname, login,
password, address, phoneNumber) VALUES ("angela.merkel@gmail.com",
"Kerrel", "Angela", "Deutsch92", SHA1("toor"), "1
root de Germany", "0102030405");
INSERT INTO User (mail, name, firstname,
login, password, address, phoneNumber) VALUES ("justin.trudeau@gmail.com",
"Trudeau", "Justin", "Canada93", SHA1("toor"), "1
root du Canada", "0102030405");

INSERT INTO Employee (mail,
employeeFunction, employeeService, deskNumber, profil) VALUES ("root@gmail.com",
"ceo", "administration", 1, "ADMIN");
INSERT INTO Employee
(mail, employeeFunction, employeeService, deskNumber, profil) VALUES
("angela.merkel@gmail.com", "secrétaire", "secrétariat", 2,
"LOAN_ADMIN");
INSERT INTO Employee (mail, employeeFunction, employeeService,
deskNumber, profil) VALUES ("justin.trudeau@gmail.com", "secrétaire",
"secrétariat", 2, "EQUIPMENT_ADMIN");

INSERT INTO Equipment (id, name,
available, imageUrl) VALUES (1, "Projecteur", 1,
"https://www.abtechnologies.net/1281-large_default/video-projecteur-sony-vlp-dx100-2300ans
i-lumens.jpg");
INSERT INTO Equipment (id, name, available, imageUrl) VALUES (2,
"Fiat 500 rouge", 1,
"https://static.latribune.fr/full_width/491371/nouvelle-fiat-500-2015.jpg");
INSERT
INTO Equipment (id, name, available, imageUrl) VALUES (3, "Peugeot Partner gris", 1,
"https://i.ytimg.com/vi/Qptt_y5S1Uo/maxresdefault.jpg");
INSERT INTO Equipment (id,
name, available, imageUrl) VALUES (4, "Harley Davidson Sportster 2 places", 1,
"https://media.motoservices.com/media/cache/slider_full/media/gallery/17550/Harley-Davidso
n_Sportster_S-08.jpg");
INSERT INTO Equipment (id, name, available, imageUrl) VALUES (5,
"Moto 2", 1,
"https://m.media-amazon.com/images/I/7140AaGHKSL._AC_SL1500_.jpg");
INSERT INTO
Equipment (id, name, available, imageUrl) VALUES (6, "Ordinateur 1", 1,
"https://www.apple.com/v/macbook-pro-14-and-16/a/images/meta/macbook-pro-14-and-16_overvie
w_fz0lron5xyuu_og.png");
INSERT INTO Equipment (id, name, available, imageUrl) VALUES
(7, "Ordinateur 2", 1,
"https://www.cdiscount.com/pdt2/b/i/s/1/700x700/r515jabq127tbis/rw/pc-portable-asus-vivobo
ok-r515ja-bq127t-15-6-fh.jpg");
INSERT INTO Equipment (id, name, available, imageUrl)
VALUES (8, "Ordinateur 3", 1,
"https://media.ldlc.com/r1600/ld/products/00/04/86/96/LD0004869623_2.jpg");
INSERT
INTO Equipment (id, name, available, imageUrl) VALUES (9, "Souris Logitech", 1,
"https://m.media-amazon.com/images/I/61UxfXTUyvL._AC_SX425_.jpg");
INSERT INTO
Equipment (id, name, available, imageUrl) VALUES (10, "Disque dur Toshiba", 1,
"https://image.darty.com/informatique/accessoire_portable/disque_dur_portable/toshiba_canv
iobasics_2_5_ltb_t1508204076290A_112059316.jpg");
INSERT INTO Equipment (id, name,
available, imageUrl) VALUES (11, "Clé USB Samsung", 1,
"https://image.darty.com/darty?type=image&source=/market/2018/08/16/mkp_mobvsASBnjv7dg

```

```

.png&width=400&height=300&quality=90");
INSERT INTO Equipment (id, name,
available, imageUrl) VALUES (12, "Remorque", 1,
"https://www.remorques-du-nord.fr/media/cache/product_image/trailers/R251M-Remorque-BW-R25
1-marron-1.jpeg");
INSERT INTO Equipment (id, name, available, imageUrl) VALUES (13,
"Porte bagage", 1,
"https://www.cdiscount.com/pdt2/9/9/0/1/700x700/auc0615200262990/rw/porte-bagages-en-metal
-barre-support-auto-voiture.jpg");
INSERT INTO Equipment (id, name, available, imageUrl)
VALUES (14, "Barres de toit", 1,
"https://accessoires.skoda.fr/pub/media/catalog/product/cache/6b50632d4c5750c2ac55e8e9234d
1d8c/b/a/barres-de-toit-scala-0.jpg");
INSERT INTO Equipment (id, name, available,
imageUrl) VALUES (15, "Tractopelle", 1,
"https://www.lemoniteurmateriels.fr/mediatheque/6/3/6/004714636_926x615_fb_fffff.jpg");
;

INSERT INTO Processor (id, name, brand, numberOfCores, frequency) VALUES (1,
"i5", "Intel", 8, 3.2);

INSERT INTO GraphicCard (id, name, brand,
frequency) VALUES (1, "RTX1080", "Nvidia", 3.5);

INSERT INTO
ComputerAccessory (id) VALUES (9);
INSERT INTO ComputerAccessory (id) VALUES (10);
INSERT
INTO ComputerAccessory (id) VALUES (11);

INSERT INTO VehicleAccessory (id) VALUES
(12);
INSERT INTO VehicleAccessory (id) VALUES (13);
INSERT INTO VehicleAccessory (id) VALUES
(14);

INSERT INTO Vehicle (id, kilometers, renewalKilometers, brand, model,
registrationNumber, state, maxSpeed, numberOfWorks, power) VALUES (2, 1000, 6000,
"Fiat", "500", "AA-000-AA", "GOOD", 180, 5,
80);
INSERT INTO Vehicle (id, kilometers, renewalKilometers, brand, model, registrationNumber,
state, maxSpeed, numberOfWorks, power) VALUES (3, 1000, 6000, "Peugeot",
"Partner", "BB-047-BC", "GOOD", 500, 5, 1200);
INSERT INTO
Vehicle (id, kilometers, renewalKilometers, brand, model, registrationNumber, state, maxSpeed,
numberOfWorks, power) VALUES (4, 1000, 6000, "Harley Davidson", "Sportser
S", "AA-BX0-AA", "GOOD", 500, 5, 1200);
INSERT INTO Vehicle (id,
kilometers, renewalKilometers, brand, model, registrationNumber, state, maxSpeed,
numberOfWorks, power) VALUES (5, 1000, 6000, "Yamaha", "MT07",
"AE-0X0-AA", "GOOD", 500, 5, 1200);
INSERT INTO Vehicle (id, kilometers,
renewalKilometers, brand, model, registrationNumber, state, maxSpeed, numberOfWorks, power)
VALUES (15, 1000, 6000, "Caterpillar", "MT07", "AE-0X0-AA",
"GOOD", 500, 5, 1200);

INSERT INTO Car (id, numberOfWorks) VALUES (2, 5);
INSERT
INTO Car (id, numberOfWorks) VALUES (3, 5);

INSERT INTO Bike (id, numberOfWorks) VALUES
(4, 8);
INSERT INTO Bike (id, numberOfWorks) VALUES (5, 16);

INSERT INTO
SuitableVehicle (accessoryId, vehicleId) VALUES (12, 2);
INSERT INTO SuitableVehicle
(accessoryId, vehicleId) VALUES (13, 3);
INSERT INTO SuitableVehicle (accessoryId, vehicleId)
VALUES (14, 2);

INSERT INTO Computer (id, brand, model, serialNumber, memorySize, isLaptop,
screenSize, purchaseDate, renewalDate, processorId, graphicCardId) VALUES (6,
"Apple", "Macbook", "ax8000xa", 500, true, 13, '2021-05-15',

```

```

'2025-05-15', 1, 1);
INSERT INTO Computer (id, brand, model, serialNumber, memorySize,
isLaptop, screenSize, purchaseDate, renewalDate, processorId, graphicCardId) VALUES (7,
"Asus", "Vivobook 8", "ax8000xa", 500, true, 13, '2021-05-15',
'2025-05-15', 1, 1);
INSERT INTO Computer (id, brand, model, serialNumber, memorySize,
isLaptop, screenSize, purchaseDate, renewalDate, processorId, graphicCardId) VALUES (8,
"HP", "EliteBook H5", "ax8000xa", 500, true, 13, '2021-05-15',
'2025-05-15', 1, 1);

INSERT INTO Loan (id, equipmentId, userMail, beginningDate, endDate,
isOver) VALUES (0, 1, "angela.merkel@gmail.com", "2021-11-16",
"2021-11-29", false);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (1, 2, "angela.merkel@gmail.com", "2021-12-14",
"2021-12-19", false);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (2, 3, "angela.merkel@gmail.com", "2021-10-12",
"2021-11-02", false);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (3, 4, "angela.merkel@gmail.com", "2022-10-12",
"2022-11-02", false);

INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (4, 1, "root@gmail.com", "2022-01-14",
"2022-01-16", false);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (5, 2, "root@gmail.com", "2021-11-30",
"2021-12-03", false);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (6, 3, "root@gmail.com", "2021-11-30",
"2021-12-15", false);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (7, 4, "root@gmail.com", "2022-11-30",
"2022-12-15", false);

INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (8, 1, "vladimir.putin@gmail.com", "2022-03-02",
"2022-03-13", false);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (9, 5, "vladimir.putin@gmail.com", "2022-06-03",
"2022-06-15", false);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (10, 6, "vladimir.putin@gmail.com", "2021-10-10",
"2021-10-30", true);

INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (11, 1, "emmanuel.macron@gmail.com", "2021-10-15",
"2021-10-17", true);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (12, 5, "emmanuel.macron@gmail.com", "2022-02-22",
"2022-03-09", false);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (13, 5, "emmanuel.macron@gmail.com", "2022-03-10",
"2022-04-25", false);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (14, 6, "emmanuel.macron@gmail.com", "2022-02-22",
"2022-03-09", false);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (15, 7, "emmanuel.macron@gmail.com", "2022-01-10",
"2022-01-25", false);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (16, 8, "emmanuel.macron@gmail.com", "2021-02-24",
"2021-04-08", true);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (17, 9, "emmanuel.macron@gmail.com", "2021-11-21",
"2021-11-28", false);
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (18, 10, "emmanuel.macron@gmail.com", "2021-09-05",
"2021-09-06", true);

INSERT INTO Loan (id, equipmentId, userMail, beginningDate,
endDate, isOver) VALUES (19, 1, "justin.trudeau@gmail.com", "2021-06-10",
"2021-06-25", true);

```

```
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,  
endDate, isOver) VALUES (20, 11, "justin.trudeau@gmail.com", "2021-06-01",  
"2021-06-09", true);  
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,  
endDate, isOver) VALUES (21, 12, "justin.trudeau@gmail.com", "2021-06-13",  
"2021-06-16", true);  
INSERT INTO Loan (id, equipmentId, userMail, beginningDate,  
endDate, isOver) VALUES (22, 13, "justin.trudeau@gmail.com", "2021-06-22",  
"2021-06-27", true);
```