

# Java distribué

Gestionnaire d'agendas

Pauline Réquéna

Guillaume Dubuisson Duplessis

28 mars 2009

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Analyse détaillée du sujet</b>	<b>4</b>
2.1	Description du sujet . . . . .	4
2.2	Structure du site . . . . .	5
2.3	Aspects techniques . . . . .	5
2.4	Diagramme de Gantt du mini-projet . . . . .	6
<b>3</b>	<b>Modélisation</b>	<b>7</b>
3.1	Identification des acteurs . . . . .	7
3.2	Diagramme de cas d'utilisation . . . . .	7
3.3	Les scénarios détaillés . . . . .	8
3.3.1	Scénario "s'authentifier" . . . . .	8
3.3.2	Scénario "Création d'un nouvel agenda" . . . . .	9
3.3.3	Scénario "Modifier les paramètres d'un agenda" . . . . .	10
3.3.4	Scénario "Supprimer un agenda" . . . . .	12
3.3.5	Scénario "Création d'un nouvel évènement" . . . . .	13
3.3.6	Scénario "Modifier les détails d'un évènement" . . . . .	15
3.3.7	Scénario "Supprimer un évènement" . . . . .	17
3.4	Diagramme d'activité . . . . .	18
3.4.1	Authentification . . . . .	19
3.4.2	Création d'agenda . . . . .	19
3.4.3	Création d'évènement . . . . .	20
3.4.4	Suppression d'agenda . . . . .	21
3.4.5	Suppression d'évènement . . . . .	22
3.5	Diagramme de séquences . . . . .	22
3.5.1	Authentification . . . . .	23
3.5.2	Creer un agenda . . . . .	24
3.5.3	Modifier un agenda . . . . .	25
3.5.4	Supprimer un agenda . . . . .	26
3.5.5	Création d'un évènement . . . . .	27
3.5.6	Modifier un évènement . . . . .	28
3.5.7	Supprimer un évènement . . . . .	29
3.6	Diagramme de classes . . . . .	30
<b>4</b>	<b>Conclusion</b>	<b>31</b>

# 1 Introduction

A faire

## 2 Analyse détaillée du sujet

### 2.1 Description du sujet

Le sujet de notre mini-projet de Java Distribué consiste en la conception et l'implémentation d'un site web de gestion d'agendas. Ce site a pour fonction de gérer les différents types d'agendas d'un utilisateur. Par exemple, un utilisateur peut disposer d'un agenda « Travail » pour ses rendez-vous professionnels et d'un agenda « Perso » pour ses activités personnelles. Cette application permettra donc la gestion individuelle de ces 2 agendas.

Un utilisateur de ce gestionnaire devra donc pouvoir avoir accès aux fonctionnalités suivantes :

- **Consulter ses agendas sous une forme claire et fonctionnelle**

Nous avons opté pour l'affichage des agendas par semaine. Par ailleurs, afin de pouvoir bien distinguer les différents agendas lors de l'affichage des événements, chaque agenda est caractérisé par une couleur.

- **Créer un nouvel agenda**

Un agenda est caractérisé par un nom, un lieu d'application, une description, ainsi qu'une couleur.

- **Modifier les paramètres d'un agenda préexistant**

Une fois qu'un agenda a été créé, l'utilisateur doit pouvoir s'il le souhaite en modifier les détails ou la couleur d'affichage.

- **Supprimer un agenda préexistant**

Lorsqu'un agenda n'est plus utilisé par l'internaute, celui-ci peut le supprimer, afin de ne pas encombrer son compte de données inutiles. Ce n'est pas une action anodine, car toute suppression est irréversible et entraîne la suppression de tous événements de l'agenda.

- **Créer un nouvel événement dans un agenda préexistant**

Un événement appartient à un agenda. Il est caractérisé par un objet, une date, un lieu, des heures de début et de fin, ainsi qu'une description.

- **Modifier les caractéristiques d'un événement préexistant**

Une fois qu'un événement a été créé dans un agenda, l'utilisateur doit pouvoir s'il le souhaite en modifier les détails.

- **Supprimer un événement préexistant**

## 2.2 Structure du site

L'accès à ce site se fera par le biais d'une page d'authentification. En effet, les comptes de chaque utilisateur sont confidentiels. Pour se connecter au gestionnaire d'agendas, l'internaute devra donc en premier lieu saisir son login et son mot de passe.

Une fois l'authentification accomplie, l'internaute sera dirigé vers la page d'accueil, composée de plusieurs parties :

- **L'en-tête**

Elle est constituée du logo de l'application, d'un message d'accueil et du bouton de déconnexion.

- **Le menu de gauche**

Ce menu est composé tout d'abord de la liste des agendas de l'utilisateur connecté. Ce dernier peut sélectionner dans cette liste les agendas qu'il désire ou non afficher.

A travers ce menu, l'utilisateur peut également accéder aux différentes fonctionnalités de gestion des agendas, à savoir la création d'un nouvel agenda, la modification ou la suppression d'un agenda, ainsi que la création d'un nouvel événement. La modification ou la suppression d'un événement se fera par le biais du calendrier d'affichage.

- **Le corps de la page**

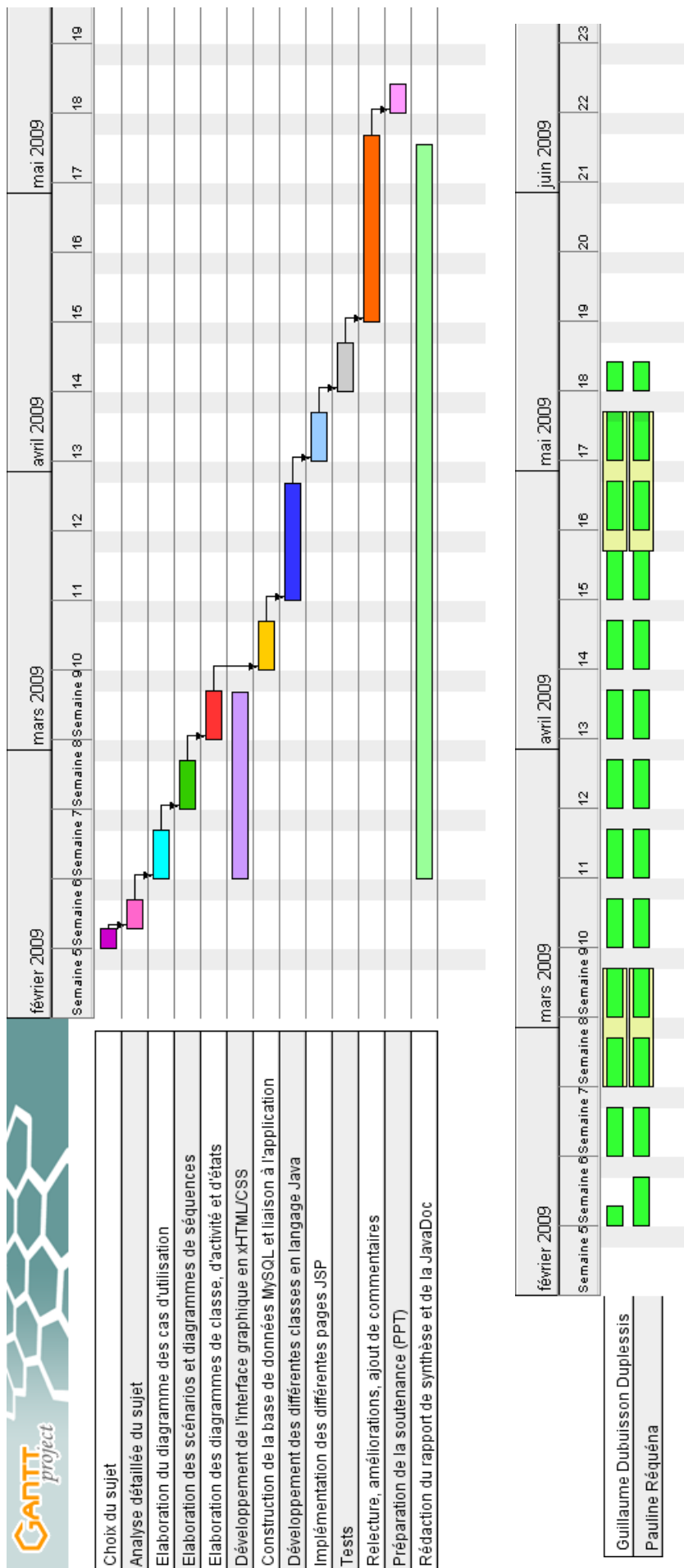
Il est composé du tableau d'affichage des agendas. En cliquant sur un des événements affiché sur ce calendrier, l'utilisateur pourra en modifier les détails, ou le supprimer.

L'affichage des agendas étant hebdomadaire, l'utilisateur pourra passer d'une semaine à l'autre à l'aide de flèches directionnelles.

## 2.3 Aspects techniques

- Le planning de ce projet sera réalisé sur le logiciel GanttProject.
- La modélisation et le développement de ce projet seront effectués sur l'IDE Netbeans 6.5.
- Les interfaces des différentes pages web seront réalisées en xHTML/CSS.
- Des JSP permettront d'insérer des données dynamiques à ce contenu statique.
- Les informations enregistrées dans l'application seront stockées dans une base de données MySQL et la liaison de l'application à la base se fera via une connexion JDBC.

## 2.4 Diagramme de Gantt du mini-projet

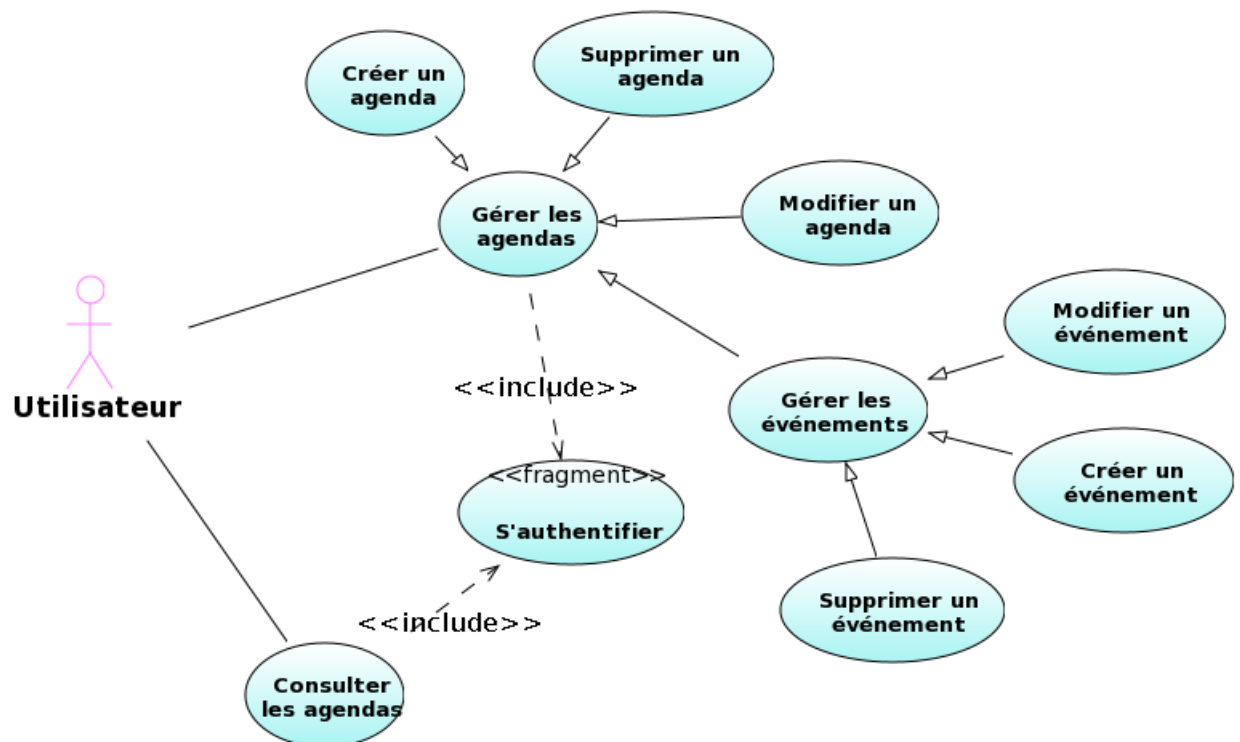


## 3 Modélisation

### 3.1 Identification des acteurs

Dans le cadre de ce gestionnaire d'agendas, nous pouvons déterminer deux principaux acteurs : l'utilisateur "lambda" qui gère ses agendas et l'administrateur du site. Dans un souci de simplicité, nous ne développerons pas l'administration du site. Nous ne considérerons donc pas l'administrateur.

### 3.2 Diagramme de cas d'utilisation



Nous pouvons remarquer qu'il n'apparaît pas de scénario d'inscription au site internet. En effet, nous préférons nous concentrer sur la gestion des agendas en elle-même que sur la gestion des membres.

### 3.3 Les scénarios détaillés

#### 3.3.1 Scénario "s'authentifier"

**Titre :** s'authentifier

**Résumé :** ce cas d'utilisation permet à l'utilisateur de s'authentifier

**Acteur :** utilisateur

**Préconditions :**

- L'utilisateur n'est pas déjà authentifié
- La base de données stockant la liste utilisateur/mot de passe est accessible

**Scénario nominal :**

1. Le système demande à l'utilisateur un nom d'utilisateur et un mot de passe
2. L'utilisateur indique son nom d'utilisateur et son mot de passe
3. Le système vérifie qu'un utilisateur correspond au nom d'utilisateur
4. Le système vérifie que le mot de passe indiqué correspond au mot de passe associé au nom de l'utilisateur
5. Le système crée une session pour l'utilisateur

**Enchaînements alternatifs :**

*A1 : nom d'utilisateur provisoirement erroné*

L'enchaînement A1 démarre au point 3 du scénario nominal.

4. Le système indique à l'utilisateur que le nom d'utilisateur est erroné pour la première ou la deuxième fois
5. Le système incrémente le compteur des tentatives d'authentification

Le scénario nominal reprend au point 1.

*A2 : mot de passe provisoirement erroné*

L'enchaînement A2 démarre au point 4 du scénario nominal.

5. Le système indique à l'utilisateur que le mot de passe est erroné pour la première ou la deuxième fois
6. Le système incrémente le compteur des tentatives d'authentification

Le scénario nominal reprend au point 1.

**Enchaînements d'erreur :**

*E1 : nom d'utilisateur définitivement erroné*

L'enchaînement E1 démarre au point 3 du scénario nominal.

4. Le système indique à l'utilisateur que le nom d'utilisateur est erroné pour la 3ème fois
5. Le système indique que toute tentative d'authentification sera refusée
6. Le système enregistre la machine comme étant bannie pour une durée déterminée (par exemple 1h)

*E2 : mot de passe définitivement erroné*

L'enchaînement E2 démarre au point 4 du scénario nominal.

5. Le système indique à l'utilisateur que le mot de passe est erroné pour la 3ème fois
6. Le système indique que toute tentative d'authentification sera refusée
7. Le système enregistre la machine comme étant bannie pour une durée déterminée (par exemple 1h)



### 3.3.2 Scénario "Création d'un nouvel agenda"

**Titre :** Création d'un nouvel agenda

**Résumé :** Ce scénario permet à l'utilisateur de créer un nouvel agenda dans son calendrier.

**Acteur :** utilisateur

**Préconditions :**

- L'utilisateur est authentifié
- La base de données stockant la table des agendas est accessible

**Scénario nominal :**

1. L'utilisateur indique qu'il souhaite créer un nouvel agenda
2. Le système affiche le formulaire de création d'agendas.
3. L'utilisateur renseigne le formulaire avec le nom de l'agenda, le lieu, une description, et une couleur représentative
4. Le système vérifie que le champ "Nom" est bien renseigné
5. Le système vérifie qu'aucun agenda du même nom n'existe déjà dans la base
6. Le système crée un nouvel agenda dans la base de données.

**Enchaînements alternatifs :**

*A1 : le champ nom est vide*

L'enchaînement A1 démarre au point 4 du scénario nominal.

5. Le système indique à l'utilisateur que le champ « Nom » n'est pas rempli et que la création est donc impossible.

Le scénario nominal reprend au point 2.

*A2 : il y a déjà un agenda du même nom dans la base*

L'enchaînement A2 démarre au point 5 du scénario nominal.

6. Le système indique à l'utilisateur que ce nom est déjà utilisé et qu'il doit en choisir un nouveau.

Le scénario nominal reprend au point 2.

### 3.3.3 Scénario "Modifier les paramètres d'un agenda"

**Titre :** Modification d'un agenda

**Résumé :** Ce scénario permet à l'utilisateur de modifier les caractéristiques d'un agenda existant, s'il veut par exemple changer sa couleur ou son nom.

**Acteur :** utilisateur

**Préconditions :**

- L'utilisateur est authentifié
- L'agenda doit exister
- La base de données stockant la table des agendas est accessible

**Scénario nominal :**

1. L'utilisateur indique qu'il souhaite modifier un de ses agendas
2. Le système affiche le formulaire de modification d'agendas
3. L'utilisateur sélectionne un agenda à modifier
4. Le système vérifie que l'agenda existe dans la base de données
5. Le système vérifie que l'agenda appartient bien à l'utilisateur
6. Le système va chercher dans la base les caractéristiques de l'agenda et les affiche dans le formulaire.
7. L'utilisateur renseigne les différents champs, suivant les modifications qu'il désire effectuer et clique sur « Modifier ».
8. Le système vérifie que le champ « Nom » est bien renseigné.
9. Le système vérifie qu'aucun agenda du même nom n'existe pas déjà dans la base.
10. Le système modifie les informations de l'agenda dans la base de données

**Enchaînements alternatifs :**

*A1 : l'agenda n'existe pas*

L'enchaînement A1 démarre au point 4 du scénario nominal.

5. Le système indique à l'utilisateur que l'agenda n'existe pas donc il n'y a pas de modification possible

Le scénario nominal reprend au point 2.

*A2 : l'agenda n'appartient pas à l'utilisateur*

L'enchaînement A2 démarre au point 5 du scénario nominal.

6. Le système indique à l'utilisateur qu'il n'a pas l'autorisation de modifier cet agenda

Le scénario nominal reprend au point 2.

*A3 : le champ nom est vide*

L'enchaînement A3 démarre au point 8 du scénario nominal.

9. Le système indique à l'utilisateur que le champ « Nom » n'est pas rempli et que la modification est donc impossible

Le scénario nominal reprend au point 6.

*A4 : il y a déjà un agenda du même nom dans la base*

L'enchaînement A4 démarre au point 9 du scénario nominal.

10. Le système indique à l'utilisateur que ce nom est déjà utilisé et qu'il doit en choisir un nouveau

Le scénario nominal reprend au point 6.

### 3.3.4 Scénario "Supprimer un agenda"

**Titre :** Suppression d'un agenda

**Résumé :** Ce scénario permet à l'utilisateur de supprimer un agenda existant, s'il n'en a plus l'utilité. Cette action a pour conséquence de supprimer tous les événements relatifs à cet agenda.

**Acteur :** utilisateur

**Préconditions :**

- L'utilisateur est authentifié
- L'agenda doit exister
- La base de données stockant la table des agendas et des événements est accessible

**Scénario nominal :**

1. L'utilisateur indique qu'il souhaite supprimer un agenda
2. Le système affiche le formulaire de modification d'agendas
3. L'utilisateur sélectionne un agenda à supprimer
4. Le système vérifie que l'agenda existe dans la base de données.
5. Le système vérifie que l'agenda appartient bien à l'utilisateur
6. Le système va chercher dans la base les caractéristiques de l'agenda et les affiche dans le formulaire.
7. L'utilisateur confirme la suppression
8. Le système supprime l'agenda dans la base de données, ainsi que tous les événements liés à cet agenda

**Enchaînements alternatifs :**

*A1 : l'agenda n'existe pas*

L'enchaînement A1 démarre au point 4 du scénario nominal.

5. Le système indique à l'utilisateur que l'agenda n'existe pas donc il n'y a pas de modification possible

Le scénario nominal reprend au point 2.

*A2 : l'agenda n'appartient pas à l'utilisateur*

L'enchaînement A2 démarre au point 5 du scénario nominal.

6. Le système indique à l'utilisateur qu'il n'a pas l'autorisation de modifier cet agenda

Le scénario nominal reprend au point 2.

### 3.3.5 Scénario "Création d'un nouvel évènement"

**Titre :** Création d'un nouvel évènement

**Résumé :** Ce scénario permet à l'utilisateur de créer un nouvel évènement dans un agenda de son calendrier

**Acteur :** utilisateur

**Préconditions :**

- L'utilisateur est authentifié
- L'agenda doit exister
- La base de données stockant la table des agendas et des événements est accessible

**Scénario nominal :**

1. L'utilisateur indique qu'il souhaite créer un nouvel événement
2. Le système affiche le formulaire de création d'évènement.
3. L'utilisateur renseigne le formulaire avec l'objet de l'évènement, la date, le lieu, les heures de début et de fin, l'agenda auquel il appartient, et une description
4. Le système vérifie que l'agenda sélectionné existe bien dans la base.
5. Le système vérifie que l'agenda sélectionné appartient bien à l'utilisateur.
6. Le système vérifie que les champs « Objet », « Date », « Heure de début » et « Heure de fin » sont bien renseignés.
7. Le système vérifie qu'aucun événement du même agenda n'existe déjà dans la base sur une plage horaire commune.
8. Le système crée un nouvel événement dans la base de données.

**Enchaînements alternatifs :**

*A1 : l'agenda n'existe pas*

L'enchaînement A1 démarre au point 4 du scénario nominal.

5. Le système indique à l'utilisateur que l'agenda n'existe pas donc il n'y a pas de modification possible

Le scénario nominal reprend au point 2.

*A2 : l'agenda n'appartient pas à l'utilisateur*

L'enchaînement A2 démarre au point 5 du scénario nominal.

6. Le système indique à l'utilisateur qu'il n'a pas l'autorisation de modifier cet agenda

Le scénario nominal reprend au point 2.

*A3 : un champ obligatoire n'est pas renseigné*

L'enchaînement A3 démarre au point 6 du scénario nominal.

7. Le système indique à l'utilisateur que certains des champs « Objet », « Date », « Heure de début » ou « Heure de fin » ne sont pas renseignés.

Le scénario nominal reprend au point 2.

*A4 : un évènement simultané existe déjà*

L'enchaînement A3 démarre au point 7 du scénario nominal.

8. Le système indique à l'utilisateur qu'il doit changer les horaires de son événement car cette plage horaire est déjà occupée.

Le scénario nominal reprend au point 2.

### 3.3.6 Scénario "Modifier les détails d'un évènement"

**Titre :** Modification d'un évènement

**Résumé :** Ce scénario permet à l'utilisateur de modifier les détails d'un évènement existant dans un agenda de son calendrier. Il peut par exemple vouloir changer la plage horaire d'un rendez-vous, ou le lieu d'une soirée etc.

**Acteur :** utilisateur

**Préconditions :**

- L'utilisateur est authentifié
- L'agenda doit exister
- La base de données stockant la table des agendas et des évènements est accessible

**Scénario nominal :**

1. L'utilisateur sélectionne l'évènement qu'il veut modifier.
2. Le système vérifie que l'évènement sélectionné existe bien dans la base.
3. Le système vérifie que l'évènement sélectionné appartient bien à l'utilisateur.
4. Le système va chercher dans la base les caractéristiques de l'évènement et affiche le formulaire de modification d'évènement.
5. L'utilisateur renseigne le formulaire en modifiant les champs voulus et clique sur le bouton « Modifier ».
6. Le système vérifie que les champs « Objet », « Date », « Heure de début » et « Heure de fin » sont bien renseignés.
7. Le système vérifie qu'aucun évènement du même agenda n'existe déjà dans la base sur une plage horaire commune.
8. Le système modifie les caractéristiques de l'évènement dans la base de données.

**Enchaînements alternatifs :**

*A1 : l'évènement n'existe pas*

L'enchaînement A1 démarre au point 4 du scénario nominal.

5. Le système indique à l'utilisateur que l'évènement n'existe pas donc il n'y a pas de modification possible

Le scénario nominal reprend au point 1.

*A2 : l'évènement n'appartient pas à l'utilisateur*

L'enchaînement A2 démarre au point 3 du scénario nominal.

6. Le système indique à l'utilisateur qu'il n'a pas l'autorisation de modifier cet évènement

Le scénario nominal reprend au point 1.

*A3 : un champ obligatoire n'est pas renseigné*

L'enchaînement A3 démarre au point 6 du scénario nominal.

7. Le système indique à l'utilisateur que certains des champs « Objet », « Date », « Heure de début » ou « Heure de fin » ne sont pas renseignés.

Le scénario nominal reprend au point 5.

*A4 : un évènement simultané existe déjà*

L'enchaînement A4 démarre au point 7 du scénario nominal.

8. Le système indique à l'utilisateur qu'il doit changer les horaires de son évènement car cette plage horaire est déjà occupée.

Le scénario nominal revient au point 5.



### 3.3.7 Scénario "Supprimer un évènement"

**Titre :** Suppression d'un évènement

**Résumé :** Ce scénario permet à l'utilisateur de supprimer un évènement existant

**Acteur :** utilisateur

**Préconditions :**

- L'utilisateur est authentifié
- L'agenda doit exister
- La base de données stockant la table des agendas et des événements est accessible

**Scénario nominal :**

1. L'utilisateur sélectionne l'évènement qu'il veut modifier
2. Le système vérifie que l'évènement existe dans la base de données
3. Le système vérifie que l'évènement appartient bien à l'utilisateur
4. Le système va chercher dans la base les caractéristiques de l'évènement sélectionné et les affiche dans le formulaire
5. L'utilisateur clique sur « Supprimer »
6. Le système supprime l'évènement de la base de données

**Enchaînements alternatifs :**

*A1 : l'évènement n'existe pas*

L'enchaînement A1 démarre au point 2 du scénario nominal.

5. Le système indique à l'utilisateur que l'évènement n'existe pas donc il n'y a pas de modification possible

Le scénario nominal reprend au point 1.

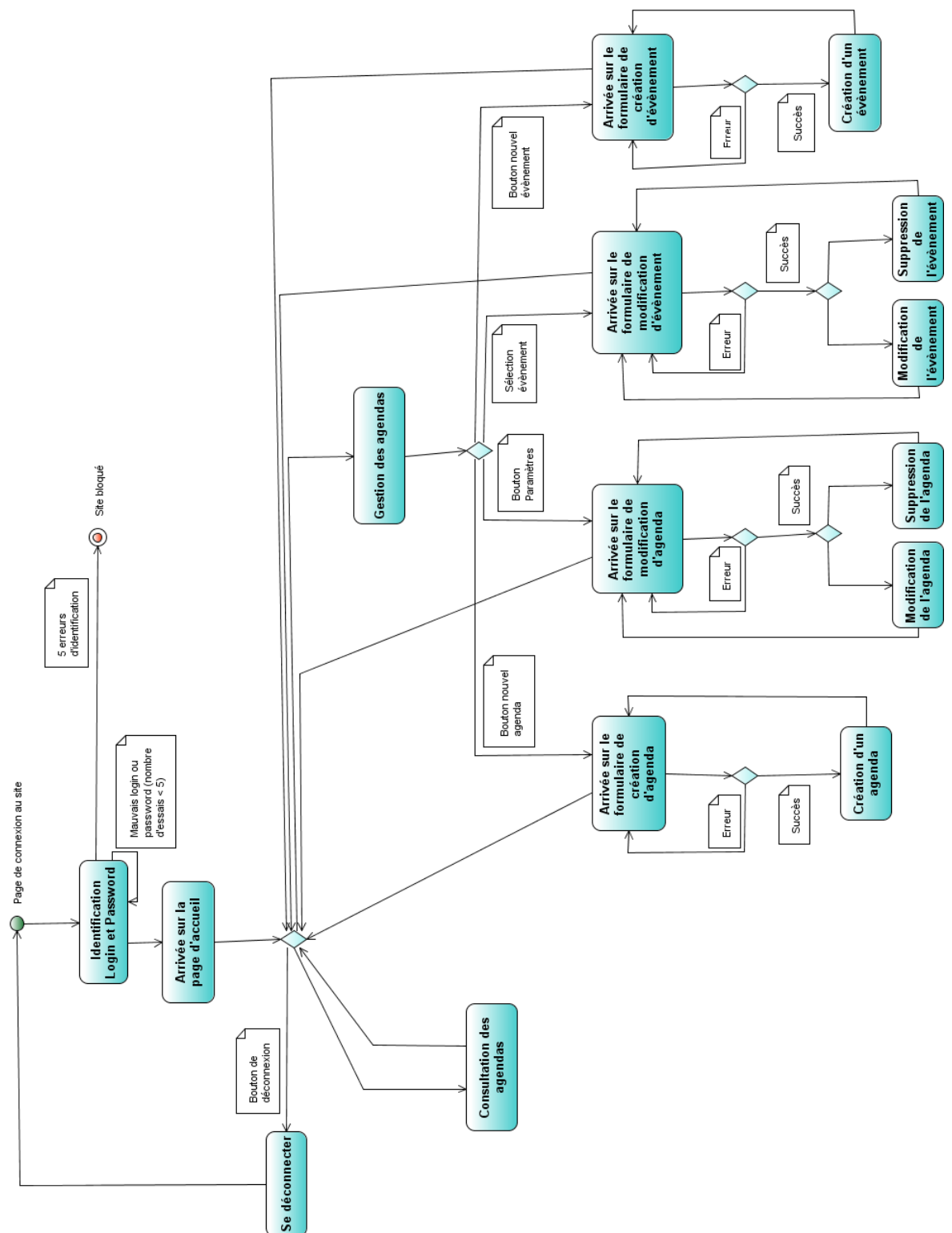
*A2 : l'évènement n'appartient pas à l'utilisateur*

L'enchaînement A2 démarre au point 3 du scénario nominal.

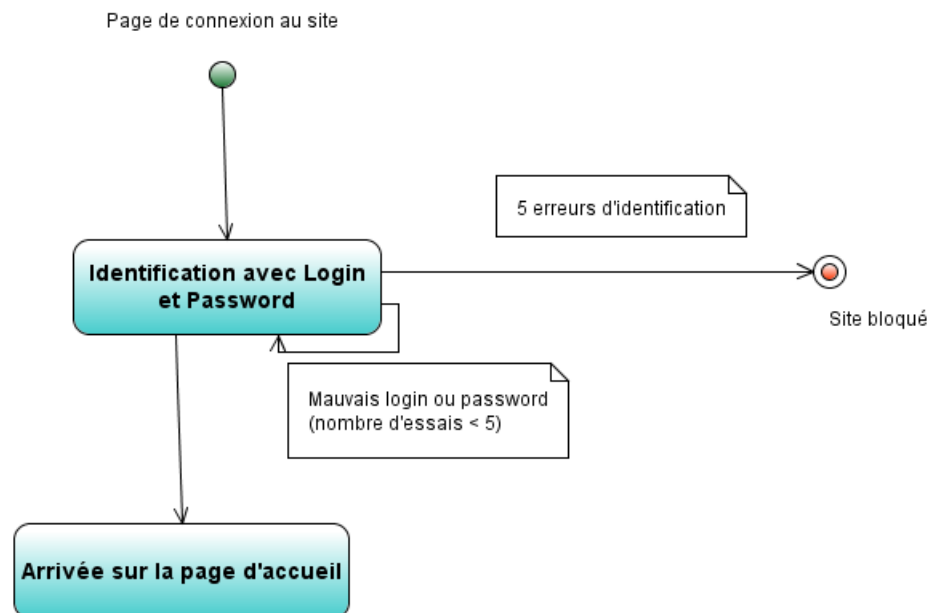
6. Le système indique à l'utilisateur qu'il n'a pas l'autorisation de modifier cet événement

Le scénario nominal reprend au point 1.

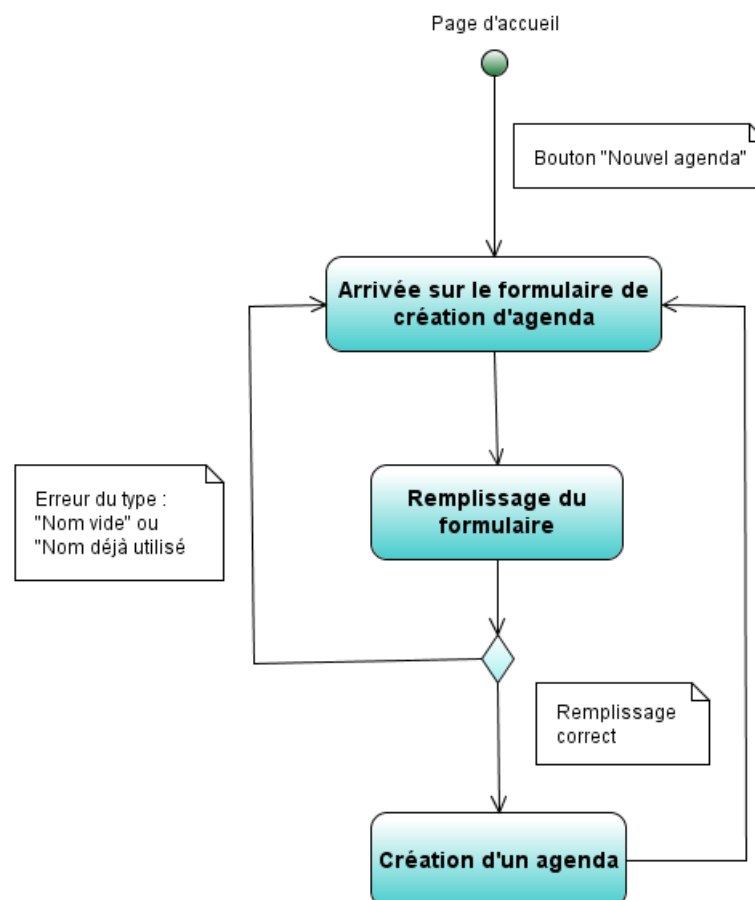
## 3.4 Diagramme d'activité



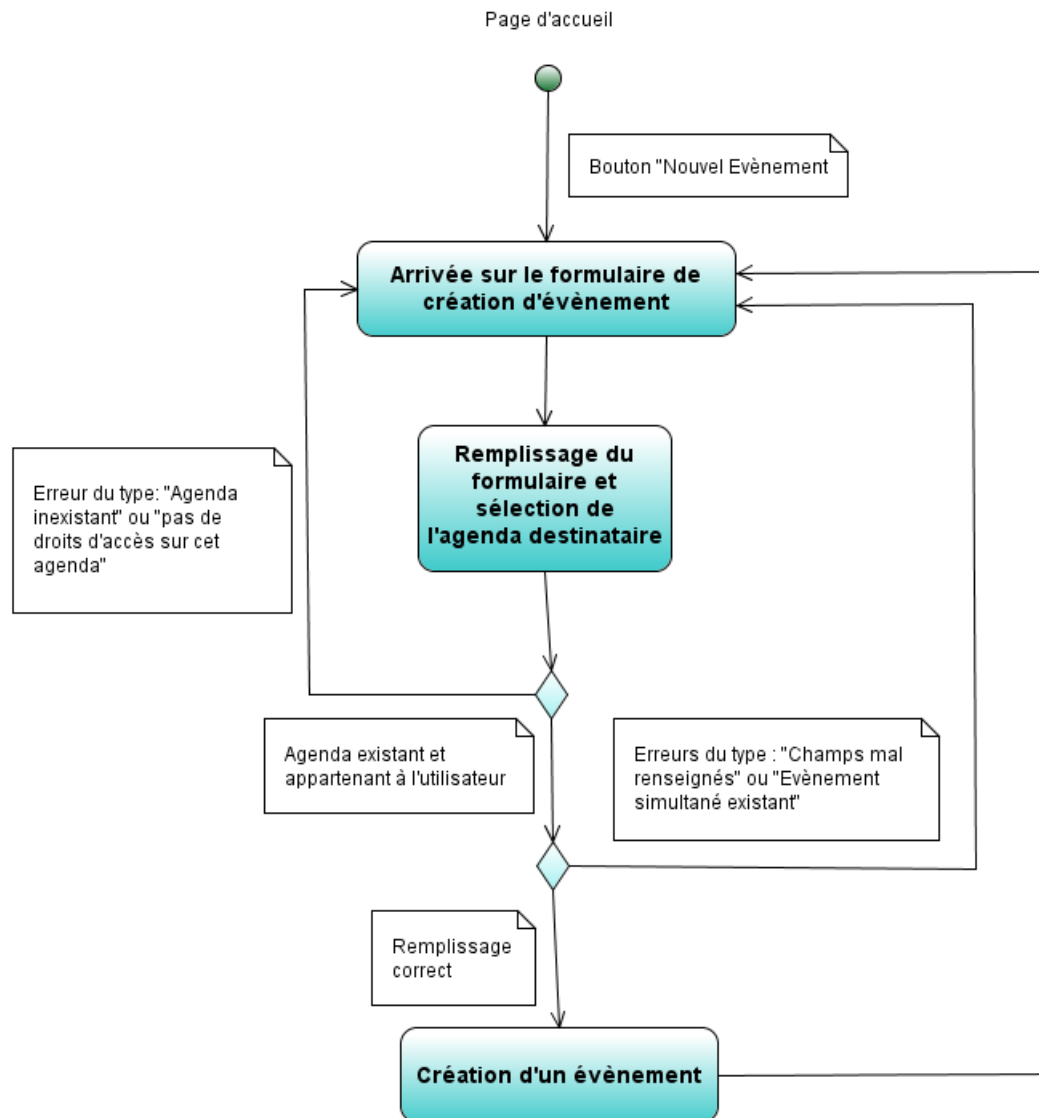
### 3.4.1 Authentification



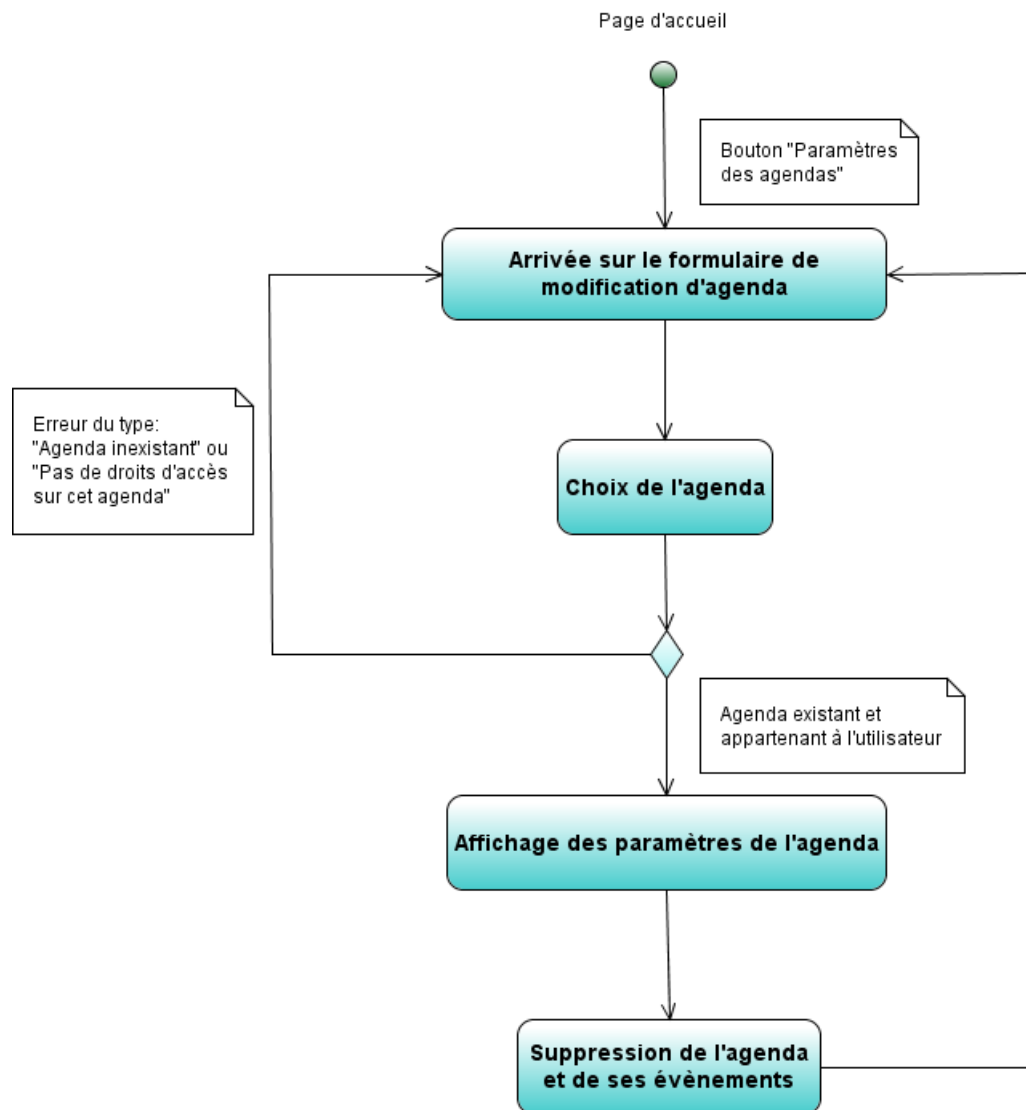
### 3.4.2 Création d'agenda



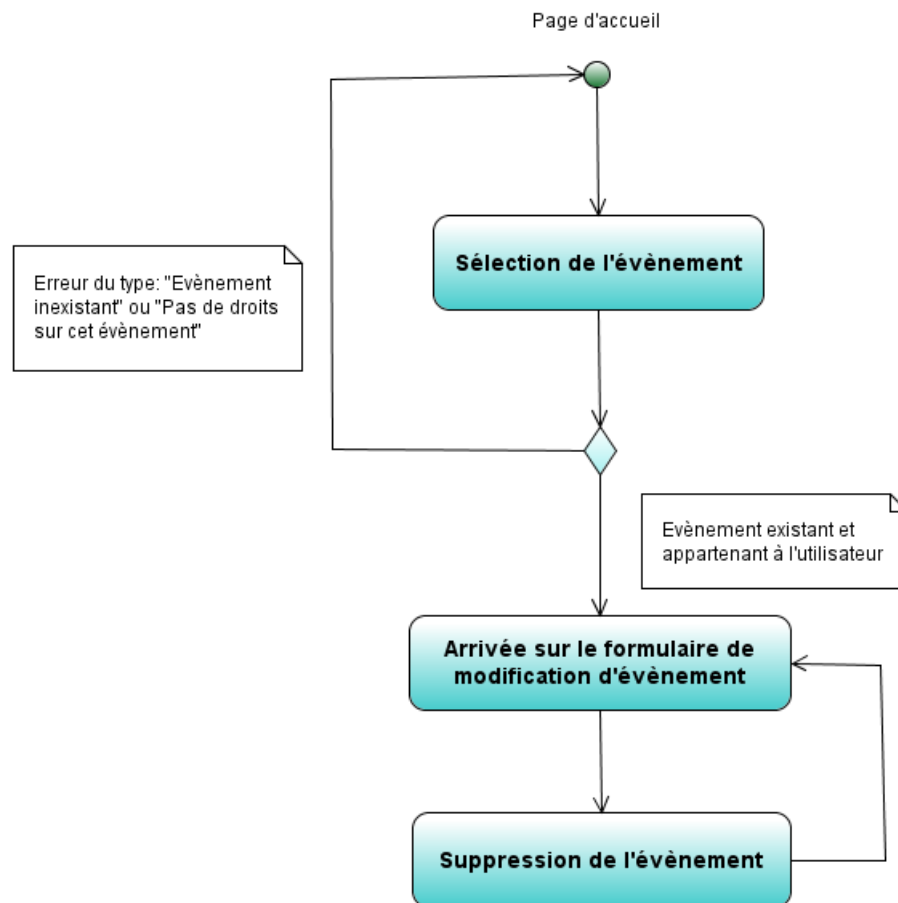
### 3.4.3 Création d'événement



### 3.4.4 Suppression d'agenda

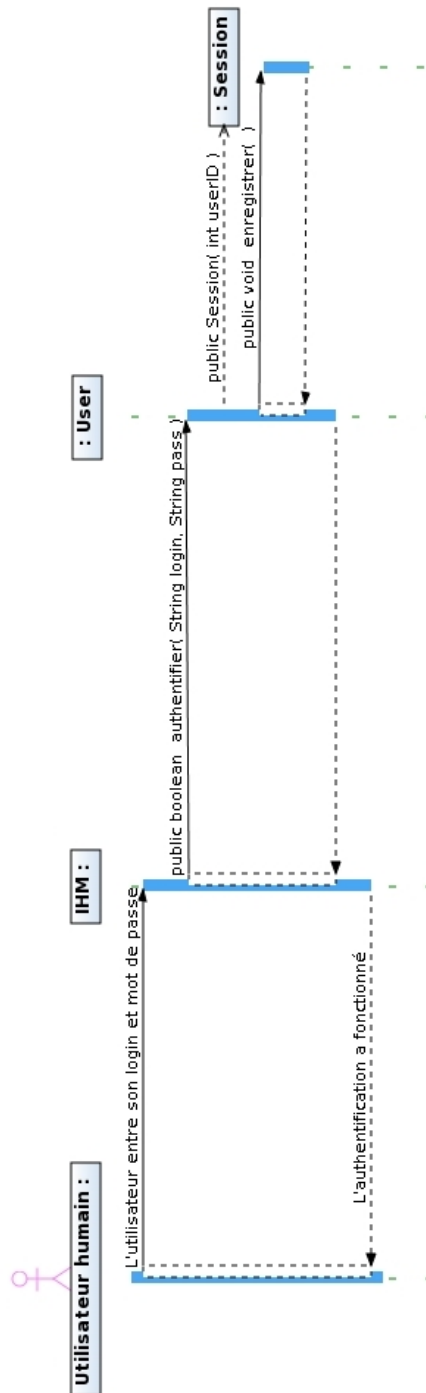


### 3.4.5 Suppression d'événement

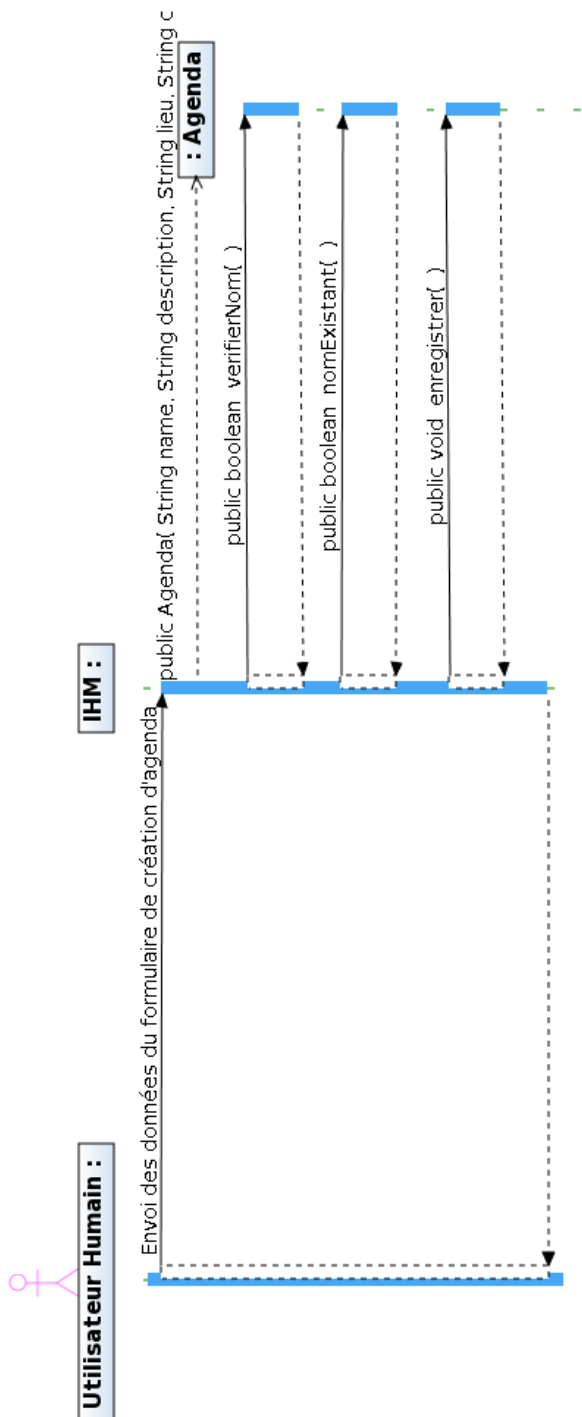


## 3.5 Diagramme de séquences

### 3.5.1 Authentification

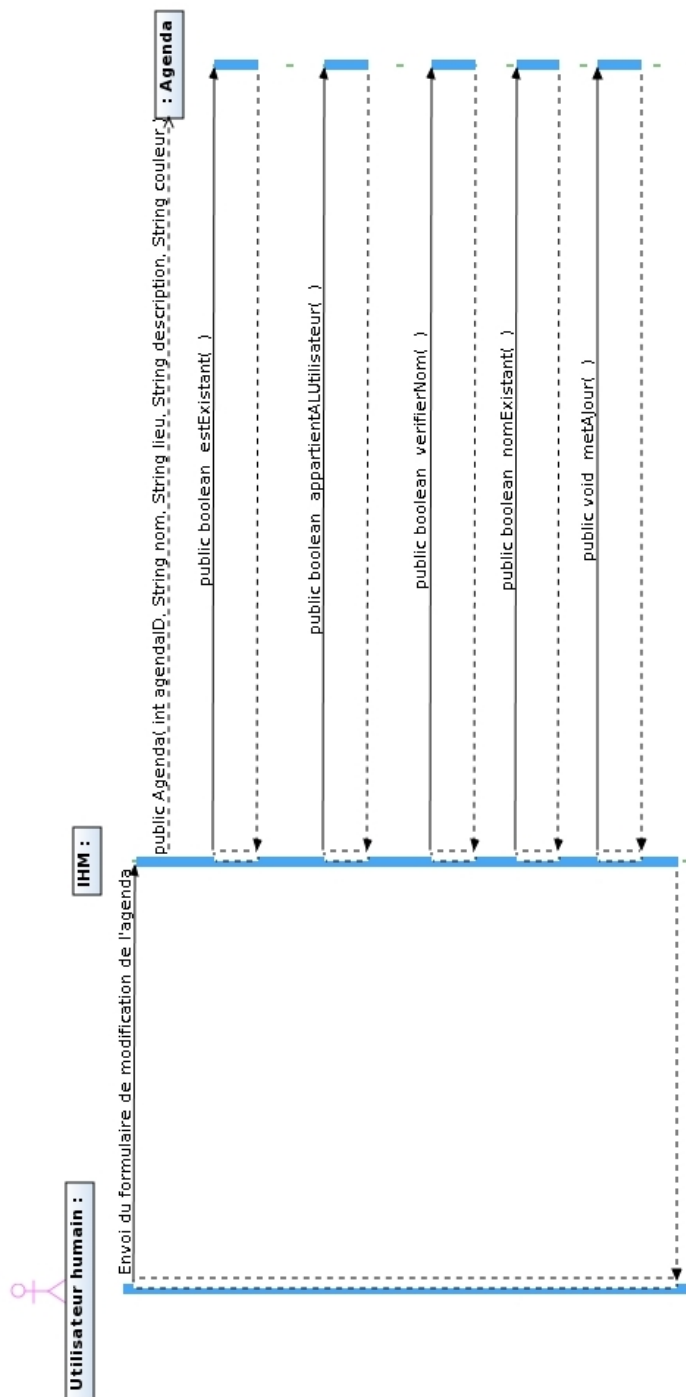


## 3.5.2 Créer un agenda

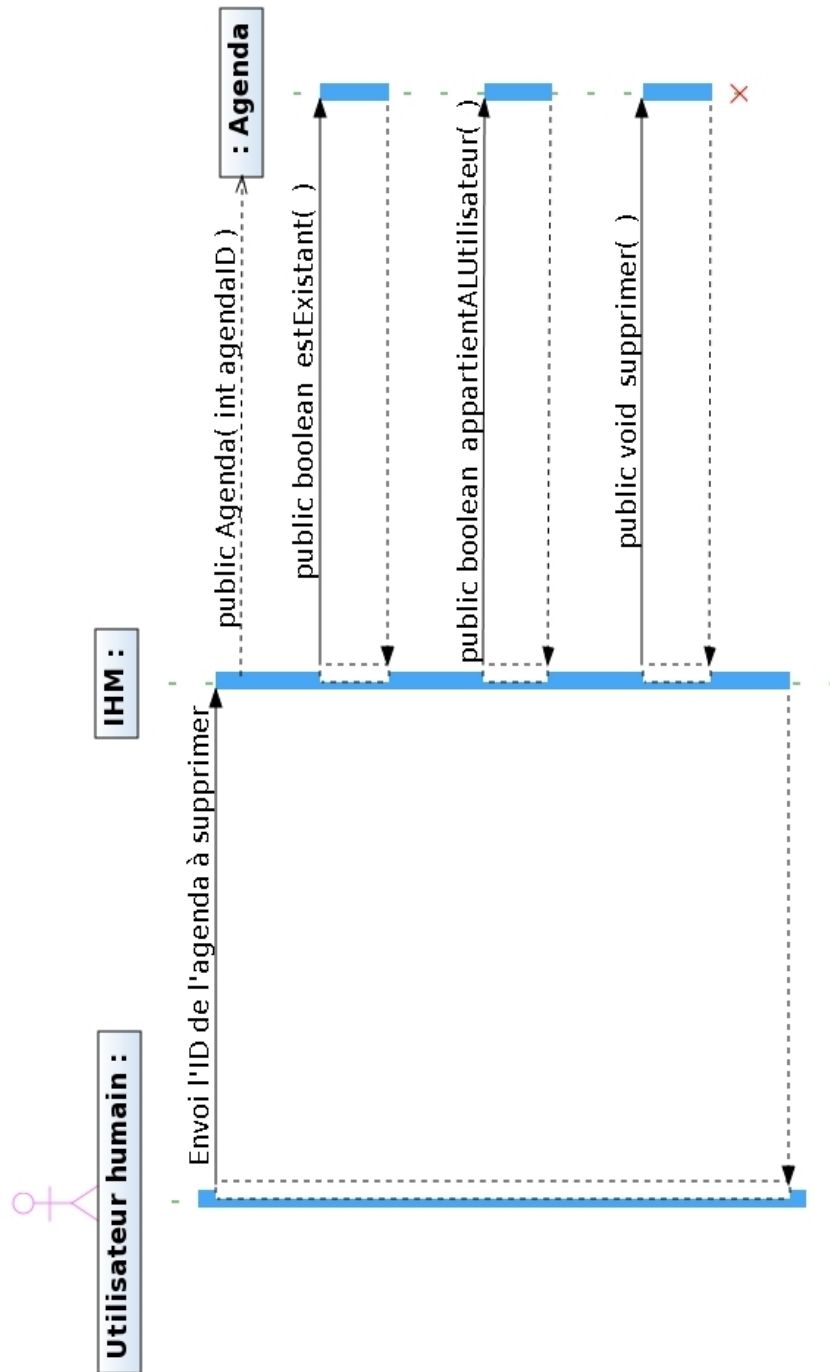




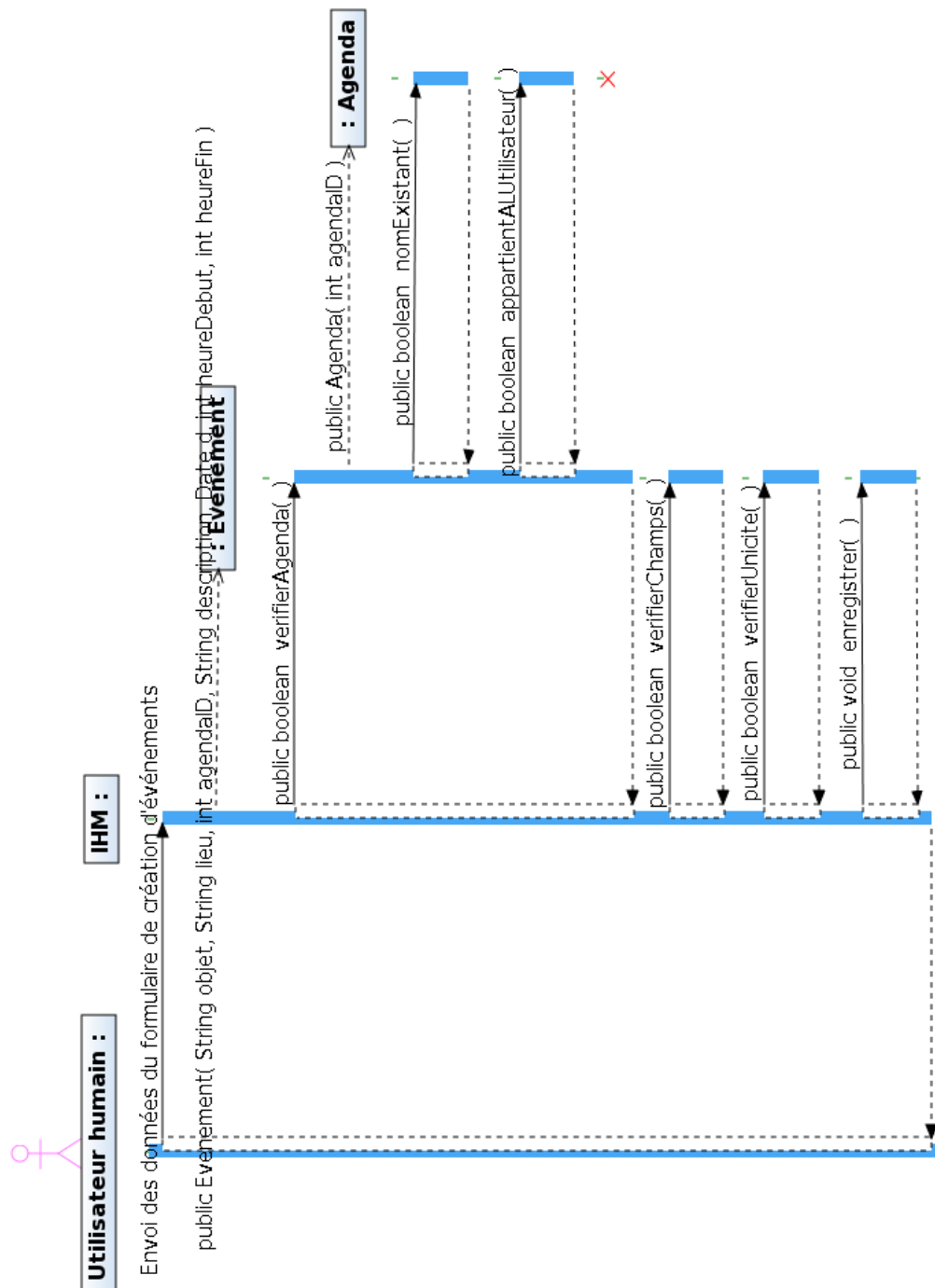
## 3.5.3 Modifier un agenda



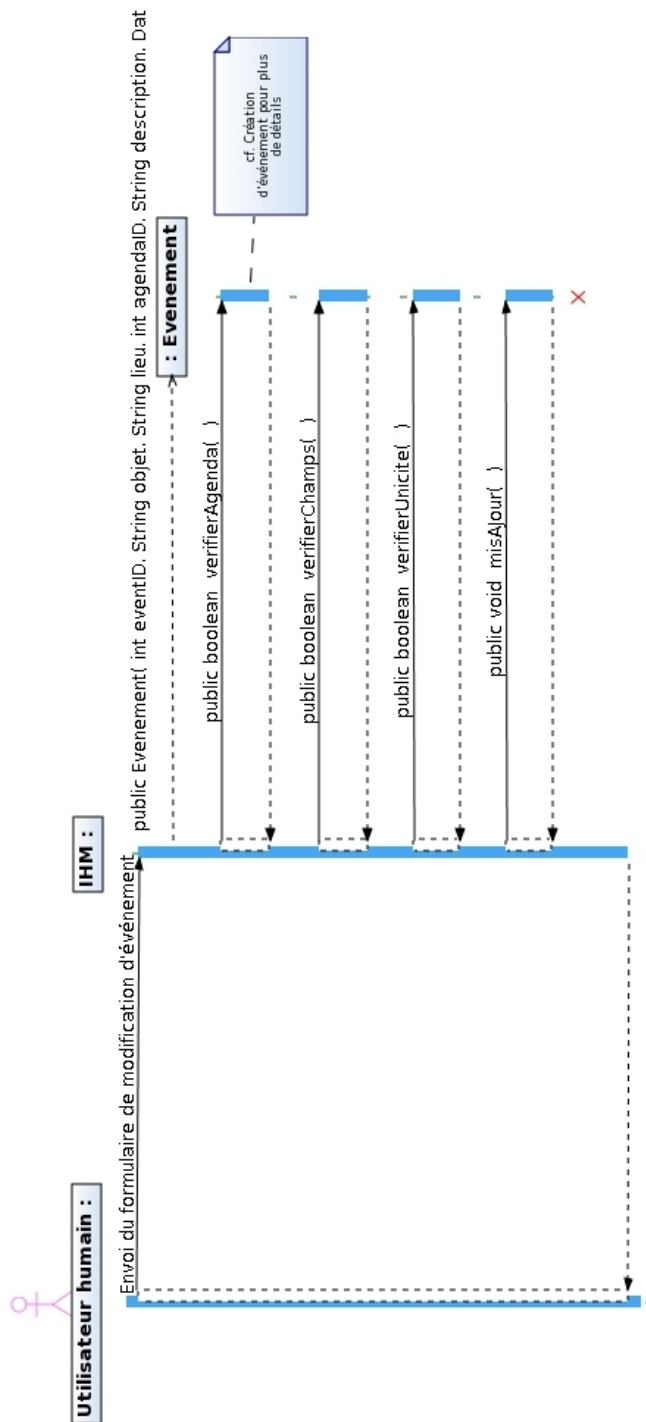
## 3.5.4 Supprimer un agenda



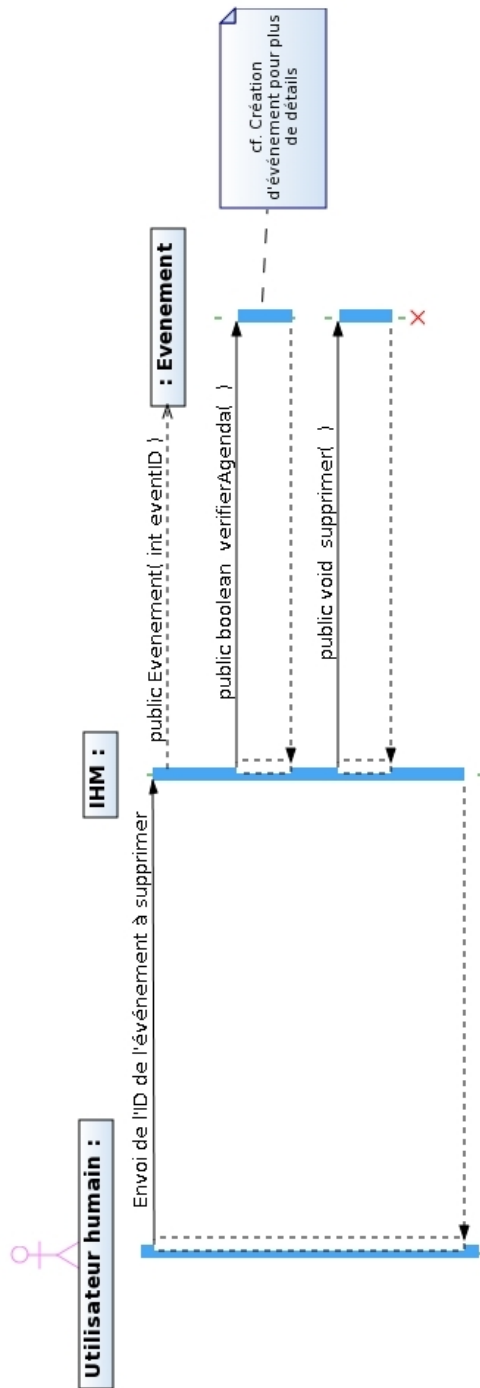
## 3.5.5 Création d'un événement



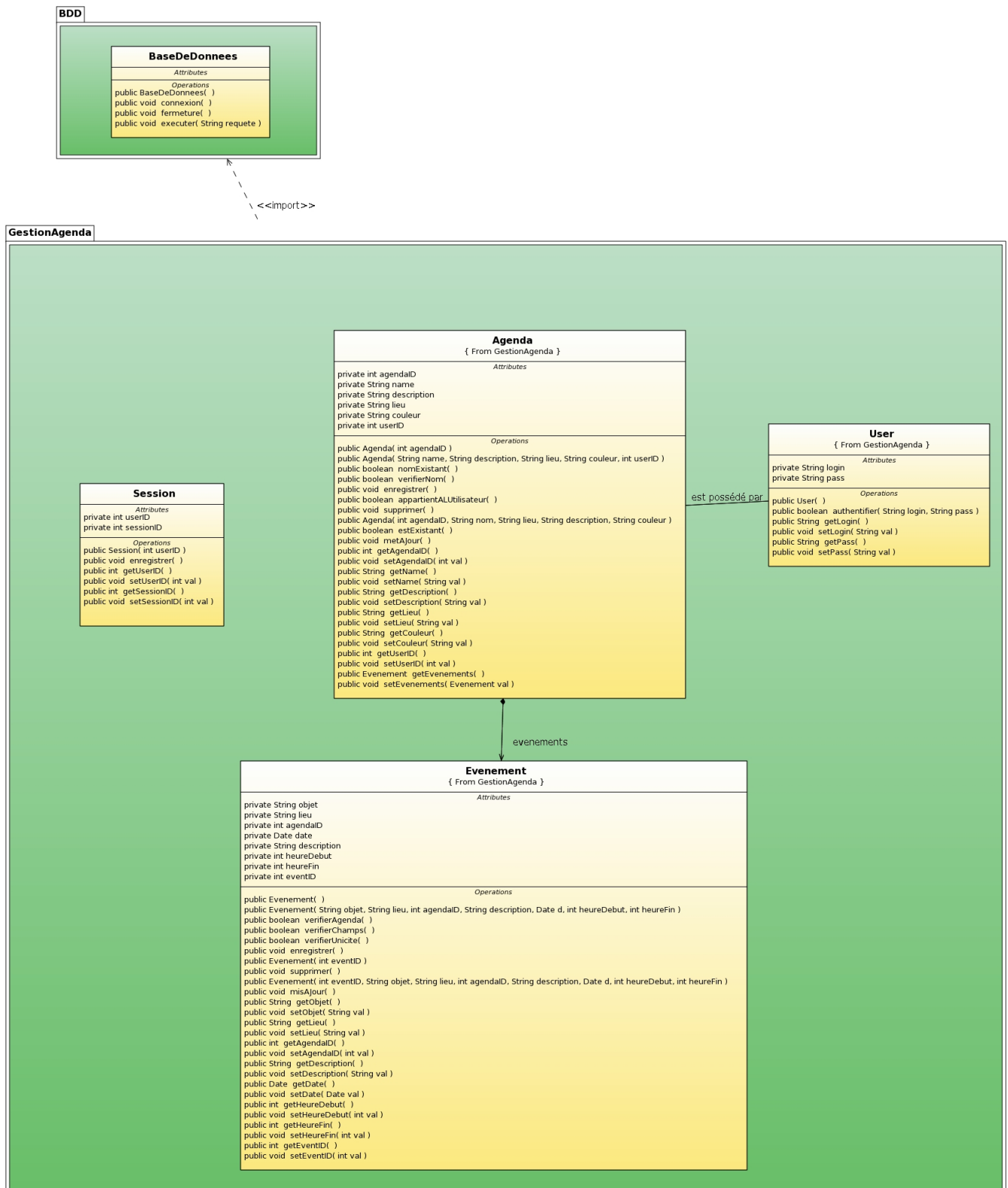
## 3.5.6 Modifier un événement



## 3.5.7 Supprimer un événement



### 3.6 Diagramme de classes



## 4 Conclusion

A faire