

Java distribué

Gestionnaire d'agendas

Pauline Réquéna

Guillaume Dubuisson Duplessis

9 mars 2009

Table des matières

1	Introduction	3
2	Analyse détaillée du sujet	4
2.1	Description du sujet	4
2.2	Structure du site	4
2.3	Aspects techniques	5
3	Cas d'utilisation	5
3.1	Identification des acteurs	5
3.2	Diagramme de cas d'utilisation	6
3.3	Les scénarios détaillés	6
3.3.1	Scénario "s'authentifier"	6
4	Conclusion	8

1 Introduction

blabla

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 évènement. La modification ou la suppression d'un évènement 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 évènements 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

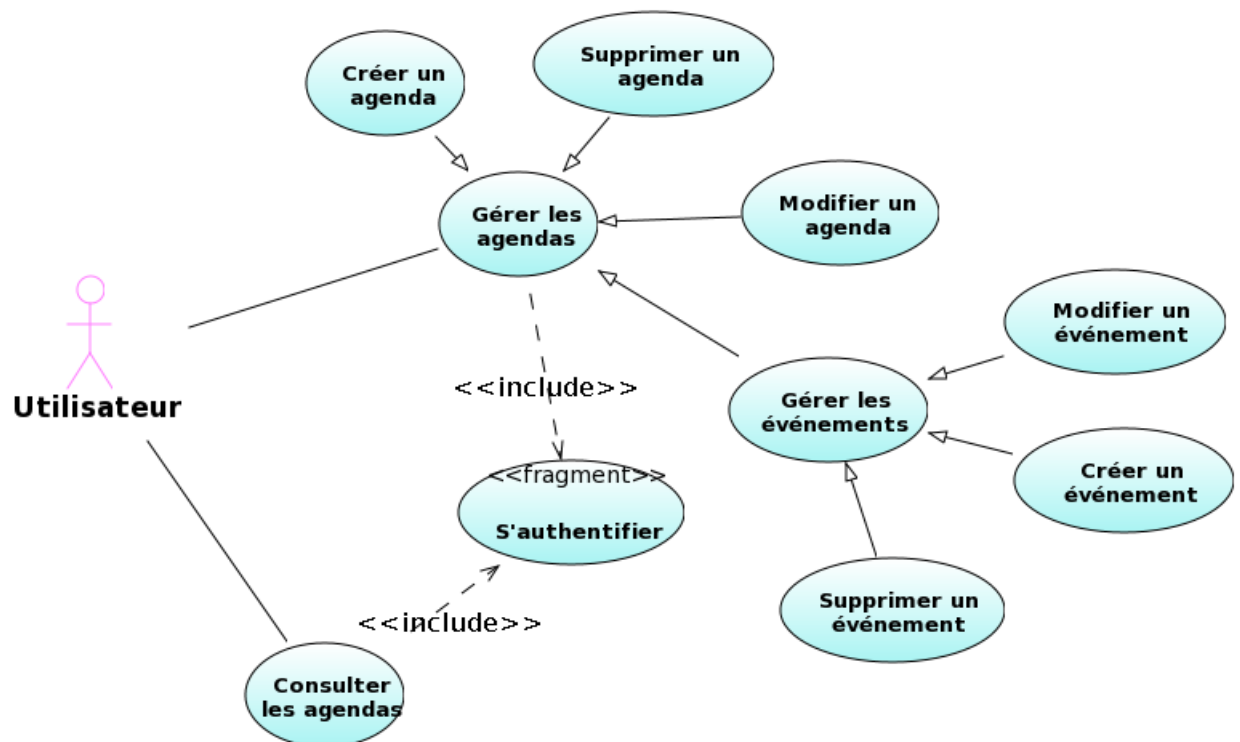
- Le planning de ce projet sera réalisé sur le logiciel GanttProject.
- La modélisation et le développement de ce projet seront effectués sur l'IDE Netbeans 6.5.
- Les interfaces des différentes pages web seront réalisées en xHTML/CSS.
- Des JSP permettront d'insérer des données dynamiques à ce contenu 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.

3 Cas d'utilisation

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)

4 Conclusion

blabla