

ECOLE NATIONALE DES SCIENCES GEOGRAPHIQUES
PROJET J2E

Rédigé par :

Amaryllis Vignaud et Claire-Marie Alla
Janvier 2022

Table des matières

1	Présentation générale du projet	2
2	Notice technique et d'installation	2
2.1	Notice technique	2
2.2	Notice d'installation	2
2.2.1	Initialisation de la base de données PostgreSQL	2
2.2.2	Démarrage de l'application	3
2.2.3	Couverture des tests	3
3	Schémas	3
3.1	Schéma de données	3
3.2	Enchaînement des pages	4
4	Fonctionnalités et limites du projet	5
4.1	Fonctionnalités opérationnelles	5
4.2	Limite du projet	5
4.3	Idées d'apports futur	5
5	Exemples	6
5.1	Les formulaires	6
5.2	Détails d'un évènement ou d'un participant	6
6	Organisation du travail en binôme	7

Table des figures

1	Structure de la base de données	3
2	Structure de l'enchaînement des pages	4
3	Formulaires d'un participant et d'un évènement	6
4	Détails d'un participants et d'un évènement	6

1 Présentation générale du projet

Le but du projet est de réaliser une application web de gestion des participants à des événements (réunion, meeting...). Ces événements sont organisés par des personnes autour d'un thème donné. Le site développé permet de gérer les événements et l'inscription de participants.

Le principe du site est le suivant : les administrateurs peuvent créer des événements avec un titre, un thème, une date de début, une durée en jours, le nombre maximum de participants, un texte de description, un organisateur et un type d'événement. Ils doivent également pouvoir ajouter des participants avec un nom, un prénom, une adresse email, une date de naissance, une organisation/entreprise et des observations éventuelles.

L'administrateur doit pouvoir ajouter, lister, modifier et supprimer les événements de l'application. Il doit pouvoir faire la même chose avec les participants. Pour ajouter un événement ou un participant, l'administrateur peut remplir un formulaire.

2 Notice technique et d'installation

2.1 Notice technique

L'application nécessite pour fonctionner :

- Un JDK de version 13 (azul...)
- Avoir installé maven (version 3.8.4)
- Un IDE (IntelliJ, VSC...)
- Avoir installé PostgreSQL (version 14.1)

2.2 Notice d'installation

2.2.1 Initialisation de la base de données PostgreSQL

Sous Linux :

```
sudo -i -u postgres
```

Sous Windows :

```
psql -U userName
```

Après s'être connecté, quel que soit l'environnement de l'ordinateur, on crée la base de données test :

```
CREATE USER test ;
ALTER ROLE test WITH CREATEDB;
CREATE DATABASE test OWNER test;
ALTER USER test WITH ENCRYPTED PASSWORD 'password';
```

2.2.2 Démarrage de l'application

Pour compiler le projet, cela peut-être compilé soit par IntelliJ, soit en ligne de commande :

```
mvn clean package
```

De même pour démarrer l'application :

```
mvn org.springframework.boot:spring-boot-maven-plugin:run
```

Une fois l'application lancée, se rendre dans le navigateur web à l'adresse localhost :8080/home.html. Pour se connecter, le nom d'utilisateur est "admin" et le mot de passe "password".

2.2.3 Couverture des tests

Pour faire tourner les tests et obtenir la couverture du code, en ligne de commande, on utilise :

```
mvn test jacoco:report
```

Puis se rendre sur la page index.html dans ProjetJ2E (ProjetJ2E/target/site/jacoco/index.html).

3 Schémas

3.1 Schéma de données

Pour expliquer le schéma de données, nous avons le diagramme suivant :

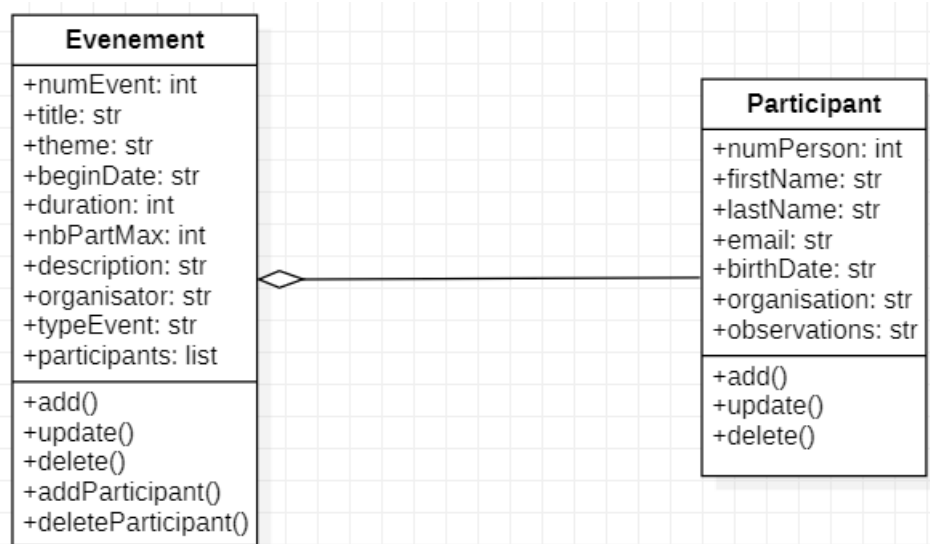


FIGURE 1 – Structure de la base de données

3.2 Enchaînement des pages

Pour visualiser l'enchaînement des pages, nous avons réalisé le diagramme ci-dessous :

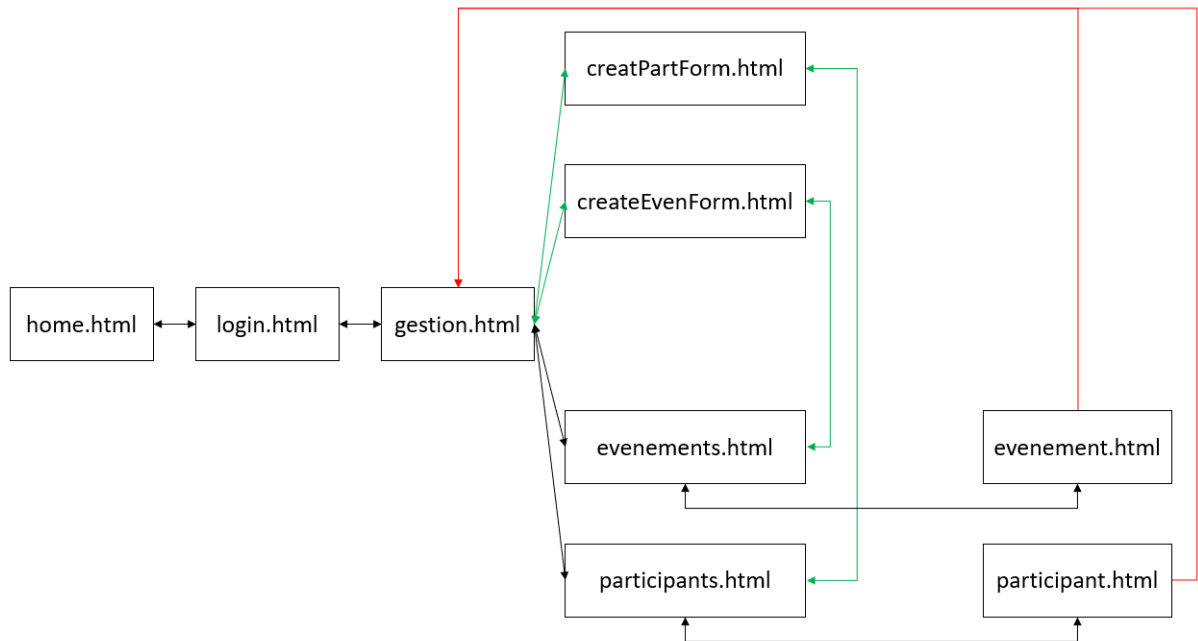


FIGURE 2 – Structure de l'enchaînement des pages

Lorsqu'on arrive sur la page d'accueil *home*, l'administrateur est directement redirigé vers la page de connexion *login*. Si les identifiants de l'administrateur entrés sont corrects, celui-ci est redirigé vers la page de *gestion*, page centrale de l'application. Dans le cas contraire, l'administrateur est renvoyé vers *login* pour tenter de se réauthentifier.

Une fois sur la page de *gestion*, l'administrateur a plusieurs choix : créer un évènement ou un participant, ou alors simplement accéder à la liste de ceux-ci. Créer un participant dans *createPartForm* renvoie après sur la liste de tous les participants. Cependant, depuis la liste des participants, l'administrateur peut être redirigé vers le formulaire de création de l'un d'eux. Le principe est également le même pour un évènement. Il faut noter que c'est depuis la page d'un *evenement* qu'un participant peut être ajouté ou supprimé de la liste s'il a été créé au préalable.

Enfin, depuis la liste des participants, on peut accéder à un participant spécifique (il en va de même pour les évènements). Depuis la liste des participants ou d'un participant spécifique, on peut revenir à la page de gestion.

Lorsque l'administrateur a fini, depuis la page *gestion*, il peut choisir de se déconnecter. Il est alors redirigé vers la page *login*.

4 Fonctionnalités et limites du projet

4.1 Fonctionnalités opérationnelles

Dans notre projet, la mise en place des différentes classes de contrôleurs et des fonctionnalités sont opérationnelles. Ainsi, on a facilement pu ajouter, supprimer ou encore modifier un événement ou un participant ainsi que des participants aux événements. La création des pages html et des différentes actions ont été bien implémentées. Il faut également noter qu'un certain nombre de champs sont requis pour la création d'un événement ou d'un participant dans un formulaire, empêchant ainsi la création d'objets vides. Le reste des champs est indiqué comme 'Non renseigné'.

Pour aller plus loin, l'administrateur doit s'authentifier pour accéder à l'application et effectuer ses tâches. Si le nombre maximum de participants est atteint, l'administrateur ne peut plus inscrire de participant à un événement. L'administrateur ne peut pas ajouter deux fois le même participant à un événement. Il peut supprimer d'un événement un participant. Enfin, la persistance des participants lors de la suppression d'un événement est maintenue. L'administrateur peut également se déconnecter.

Un travail sur l'ergonomie et l'architecture de l'application a également été réalisé. Une fois authentifié, l'administrateur peut ajouter un événement/participant, ou accéder à la liste de ceux-ci.

4.2 Limite du projet

Dans les limites du projet, on peut noter le fait qu'un participant ne peut être ajouté à un événement à la seule condition que celui-ci ait déjà été créé. De plus, il faut être sur la page d'un événement spécifique pour pouvoir ajouter ce participant en connaissant son nom.

4.3 Idées d'apports futur

Dans les améliorations immédiates, on peut revoir la façon dont un participant est ajouté à un événement. On pourrait rendre cela possible depuis la page des participants, ou encore rendre possible l'ajout d'un participant depuis la page d'un événement, en le créant directement depuis cette page. Depuis le formulaire de création d'un événement, on pourrait également pouvoir ajouter un participant.

Dans la continuité de ce projet, nous avons pensé à plusieurs idées d'apport. Un premier point serait de gérer la persistance des données, c'est-à-dire que ceux-ci soient sauvegardés dans la base de données en redémarrant l'application. On pourrait également améliorer le système d'authentification, et mettre en place l'authentification pour les utilisateurs : ils pourraient ainsi accéder à la liste des événements dans lesquels ils sont inscrits, ou encore s'inscrire à un événement. Un autre point pourrait être l'amélioration du design de l'application, pour la rendre plus agréable à regarder et ergonomique.

5 Exemples

5.1 Les formulaires

Les formulaires de l'application se présentent sous cette forme :

The image shows two side-by-side forms. The left form is titled 'Ajouter un participant :'. It contains input fields for 'Am' (with a note 'la taille doit être comprise entre 3 et 2147483647'), 'Vignaud', 'amaryllis.vignaud@ensg.eu', 'BirthDate', 'Organisation', and 'Observations'. At the bottom are 'Save' and 'Reset' buttons, and a note: 'Note : les champs '*' sont obligatoires.' The right form is titled 'Ajouter un évènement :'. It contains input fields for 'Undertale', 'Undertale AU', 'Date de début: JJ/MM/AAAA', '* Nombre de jour : 3', '* Nombre de participants max : 2', 'Gaming autour d\'Undertale', 'Toby Fox', and 'Live'. It also has 'Save' and 'Reset' buttons and the same note: 'Note : les champs '*' sont obligatoires.'

FIGURE 3 – Formulaires d'un participant et d'un évènement

A noter que si l'un des champs obligatoire est mal renseigné, la création de l'évènement n'est pas effectué et l'erreur est envoyée à l'utilisateur.

5.2 Détails d'un évènement ou d'un participant

Le détail d'un participant ou d'un évènement se fait sous la forme suivante :

The image shows two side-by-side detail views. The left view is titled 'Détails de Ama Vignaud'. It displays fields for 'Prénom : Ama', 'Nom : Vignaud', 'Email : amaryllis.vignaud@ensg.eu', 'Date de naissance : Non renseigné', 'Organisation : Non renseigné', and 'Observations : Non renseigné'. At the bottom are 'Modifier' and 'Supprimer' buttons. The right view is titled 'Détails de l'évènement Undertale'. It displays fields for 'L'évènement a pour thème : Undertale AU', 'Date de début : Non renseigné', 'Durée : 3 jours', 'Nombre de participants max : 2', 'Description : Gaming autour d\'Undertale', 'L'organisateur est Toby Fox.', 'Le type d'évènement est : Live', and 'Les participants sont : Ama Vignaud amaryllis.vignaud@ensg.eu'. Below this is a section 'La liste des personnes qui peuvent être inscrites :'. It lists 'Participant 1 Ama' with a 'Supprimer de cet évènement' button, and 'Participant 2 Claire-Marie' with an 'Ajouter' button. At the bottom are 'Modifier' and 'Supprimer' buttons.

FIGURE 4 – Détails d'un participants et d'un évènement

Ainsi, on peut manipuler un objet précis en le supprimant ou le modifiant directement sur la page dédiée. On peut également voir l'ajout ou la suppression possible d'un participant si celui-ci n'est pas déjà présent dans un évènement.

6 Organisation du travail en binôme

Le travail en binôme a été bien organisé et réparti. Les tâches étaient correctement divisées, afin que nous ne travaillions pas sur la même tâche. Ainsi, nous avons pu avancer efficacement tout au long du projet et nous ne dépendions pas de l'autre dans les tâches attribuées.

Au début du projet par exemple, une d'entre nous s'est occupée de mettre en oeuvre tout ce qui concernait la classe participant (controlleur, formulaire...) tandis que l'autre s'est occupée de tout ce qui concernait les évènements.

Quand cela a été fait, l'une de nous s'est occupée de tout ce qui concernait l'authentification et la connexion en tant qu'administrateur, tandis que l'autre s'est occupée d'ajouter des participants à un évènement et de poser des contraintes au formulaire (éviter des champs obligatoires vides...).