

Projet Appli Web

AUCHERE Nathan, GAUTHIER Nino, RICARD Mattéo, TOLSAU Clément

Définition du site

Site de gestion d'agenda

Cas d'utilisation principal : la réservation des créneaux dans un emploi du temps pour des cours ou des événements. Un organisateur peut assigner un événement à un ou plusieurs utilisateurs / groupes (cours, partiel, oral...) ou proposer des créneaux disponibles pour cet événement parmi lesquels les utilisateurs peuvent choisir (PPP, Cambridge LinguaSkill...).

Cahier des charges

- Chaque utilisateur possède un compte.
- Une fois connecté, l'utilisateur est sur un menu qui lui permet d'accéder :
 - À son agenda
 - À son panel organisateur.
 - À son panel participant.
- Sur la page agenda, l'utilisateur a une vue de son emploi du temps constitué de ses agendas :
 - Il peut voir son agenda global (superposition de ses agendas).
 - Il peut voir ses agendas séparés (avec les créneaux et les tâches enregistrées).
 - Il peut exporter et importer un agenda sous format ".ics"
 - Il peut cliquer sur les créneaux et les tâches pour avoir plus d'informations.
- Sur le panel organisateur, l'utilisateur a le choix entre :
 - Créer un événement :
 - * Rentrer les créneaux disponibles (avec leur lieu, nombre de places, etc.). -Inviter des utilisateurs à participer à l'événement, soit à partir d'un lien d'invitation, soit en ajoutant des groupes d'utilisateurs.
 - Superviser ses événements :
 - * Pour chaque événement, pouvoir voir la liste des participants, les créneaux disponibles.
 - * Pouvoir exporter les données de chaque événement (en .csv par exemple) .
 - * Pouvoir ajouter/retirer un participant d'un événement.
 - * Pouvoir annuler un événement.
 - Gérer ses groupes.
 - * Créer/Modifier ses groupes en ajoutant/retirant des utilisateurs.
 - * Important/Exporter ses groupes pour les partager avec d'autres utilisateurs.
- Sur le panel participant, l'utilisateur pourra :
 - Trouver une liste des événements auxquels il est invité.
 - Trouver des événements à partir d'un lien/code
 - Regarder l'historique des événements auxquels il a participé.
 - Se désinscrire des événements auxquels il est inscrit.
- Sur un événement, l'utilisateur pourra :
 - Voir les détails de l'événement.
 - S'inscrire sur un créneau disponible.
 - * Ceux pour lesquels il est disponible en blanc (lien avec son agenda global).
 - * Ceux pour lesquels il n'est pas disponible de façon grisée (pour lui permettre d'adapter son agenda si ça l'arrange)

Routes des Pages

Proposition d'EJB

Le sens des flèches indique seulement le sens de lecture de la relation ("OTO" par exemple) et n'indique en aucun cas qui est le porteur de la relation (ni si elle est unidirectionnelle/bidirectionnelle).

Suivi 2

L'environnement de développement est fonctionnel chez tout le groupe.

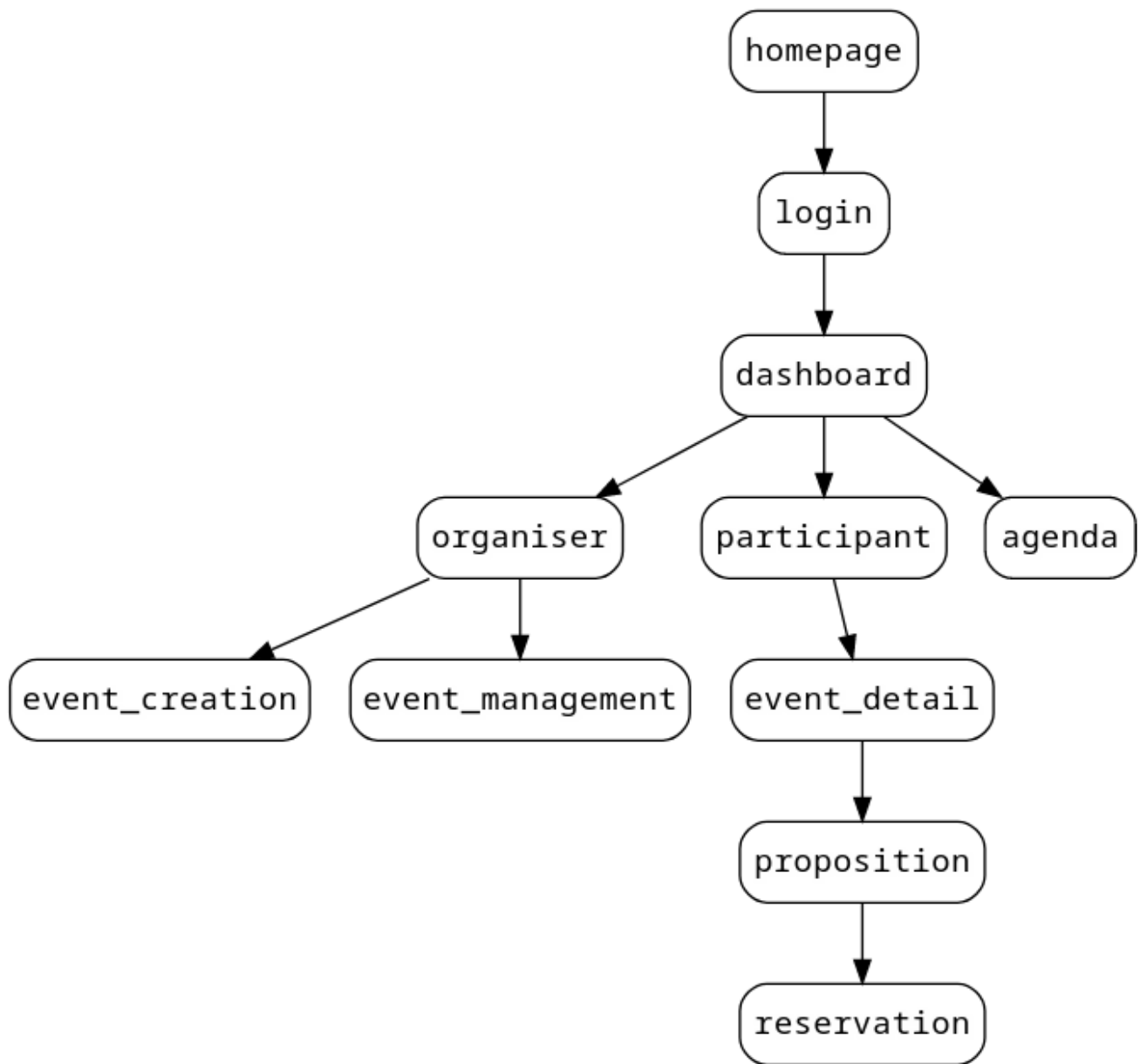


Figure 1: Route des pages prévues

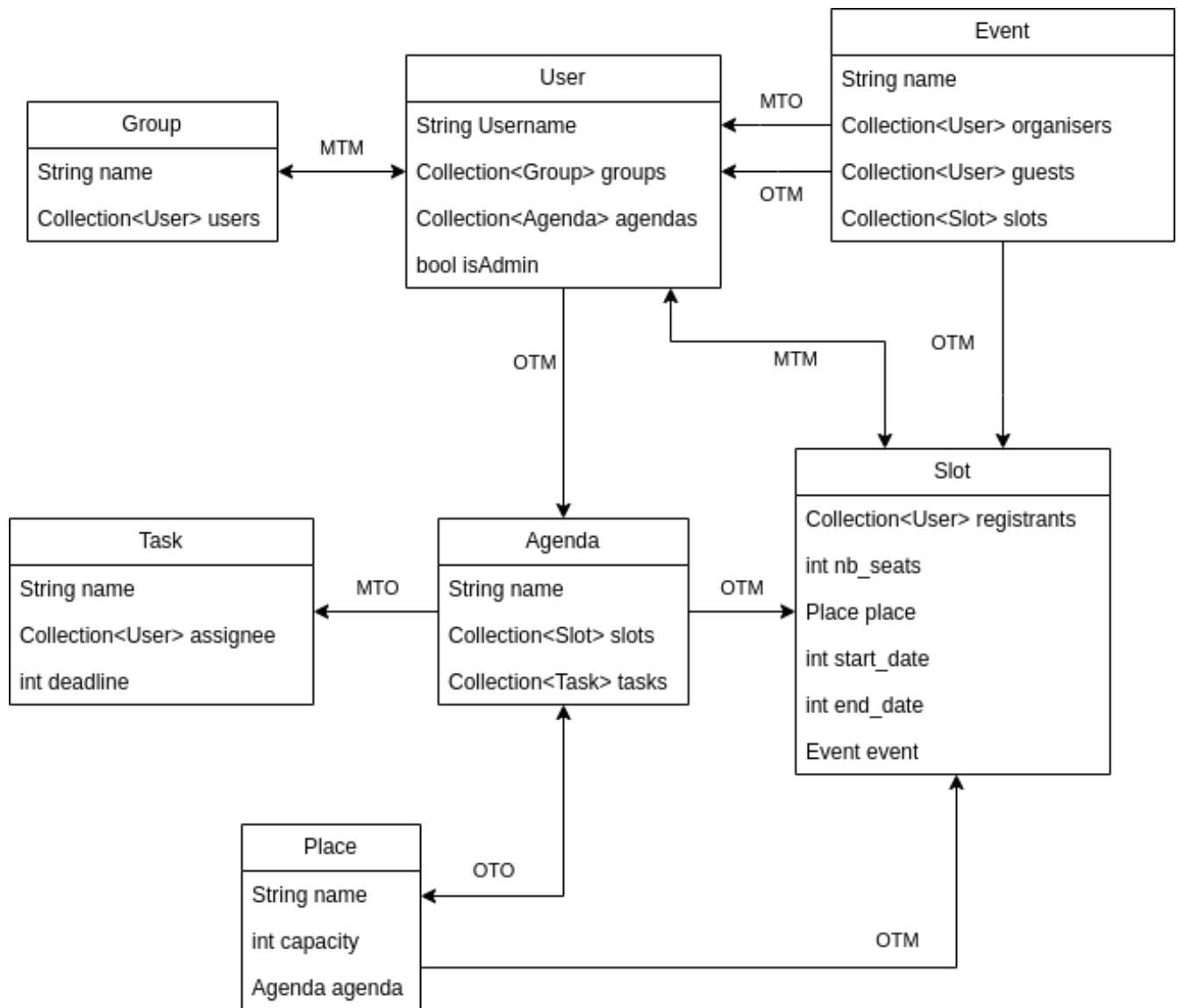


Figure 2: Porposition des EJB

Nous avons créé toutes les EJB que nous avons proposées, ainsi que les relations entre ces dernières.

Nous avons également commencé la façade en implémentant les principales fonctions dont nous pensons avoir besoin pour le moment.

Création d'une application React communiquant avec le serveur jboss via des requêtes POST afin d'éviter un changement de page trop fréquent. Pour le moment, la partie frontend affiche seulement l'équivalent du tp3 mais cette fois avec du JavaScript, c'était un exemple plus simple pour nous à prendre en main et nous pourrions plus facilement le reporter sur notre projet maintenant que nous en avons compris le fonctionnement.

Le but de la semaine prochaine serait de continuer à développer cette application React, cette fois-ci pour notre propre application, ainsi que d'utiliser tous les EJB que nous avons mis en place. Nous pourrions ainsi vérifier leur fonctionnement ainsi que leur bonne persistance dans la BD.

Suivi 3

Tous les EJB sont définis, ainsi que les relations entre ces derniers.

Nous avons avancé notre application React et créé une page d'inscription au service ainsi que de connexion au service. La structure de fichier de l'application a été repensée pour quelque chose de plus cohérent avec une meilleure séparation des pages. De plus, nous avons passé notre projet sous TypeScript pour un meilleur contrôle (il permet notamment de typer JavaScript).

Un nouvel EJB a été créé, l'EJB de token de connexion. Cet EJB est lié à l'EJB User par une relation One-ToMany/ManyToOne bidirectionnelle et permettra de garder en mémoire les cookies de session utilisateurs.

Les cookies de sessions sont eux fonctionnels, ainsi que les sessions. Lors de la connexion d'un utilisateur, ce dernier envoie son nom d'utilisateur ainsi que son mot de passe hashé. Si ces derniers coïncident, une session est lancée et un cookie est fourni à l'utilisateur, qu'il donnera au serveur à chaque requête pour s'identifier.

Concernant la communication avec la façade, nous sommes passés à une communication via requêtes POST par une API rest. Cette solution nous permet un accès simple et efficace à la façade via notre application React.

Le site a aussi commencé à être mis en page par du CSS.

Maintenant que la connexion et les sessions sont mises en place, nous pouvons nous pencher sur les fonctions façade spécifiques à chaque utilisateur et leur implémentation dans notre application, ce qui sera le but de ces prochains jours.

Séance 4

Nous avons poursuivi le développement de notre application en implémentant la vue de création des groupes ainsi qu'en corrigeant nombre de bugs pour permettre une communication correcte entre le front et le back-end.

Nous avons en parallèle commencé l'implémentation des événements.

Dernière semaine

À partir des connexions fonctionnelles, nous avons pu implémenter la majorité des fonctionnalités essentielles de notre application :

- Il est possible d'ajouter des groupes, des salles avec les paramètres correspondants sur le dashboard.
- Il est possible de créer des événements avec plusieurs créneaux possibles. Nous pouvons y assigner un nom, une salle ainsi que les utilisateurs invités.
- Une page récapitule tous les événements auxquels un utilisateur a été invité. En cliquant sur ces derniers, il est redirigé vers une page d'inscription où il peut choisir le créneau associé à cet événement qui lui convient. Il n'est pas possible de s'inscrire pour un créneau plein.
- Une page liste les événements auxquels un utilisateur est inscrit, ainsi qu'une vue sous forme d'emploi du temps qui permet de mieux visualiser les horaires.

Toutes les fonctions de la façade ne renvoient des informations que si l'utilisateur est vérifié par un token de connexion valide. De même, un utilisateur ne peut s'inscrire deux fois pour le même événement.

Recap technos utilisées

Nous avons utilisé les technologies ci-contre :

- Application React (<https://react.dev/>)
- Communication avec le serveur JBoss via une API REST (<https://restfulapi.net/>)
- Utilisation de TypeScript plutôt que JavaScript : (<https://www.typescriptlang.org/>)
 - Le typage statique permet une détection d'erreurs de typage, de conversion. – Garantir une meilleure lisibilité et compréhension du code.
- Utilisation de Sass (extension CSS) pour une personnalisation plus simple (<https://sass-lang.com/>)

Recap nouveaux EJB

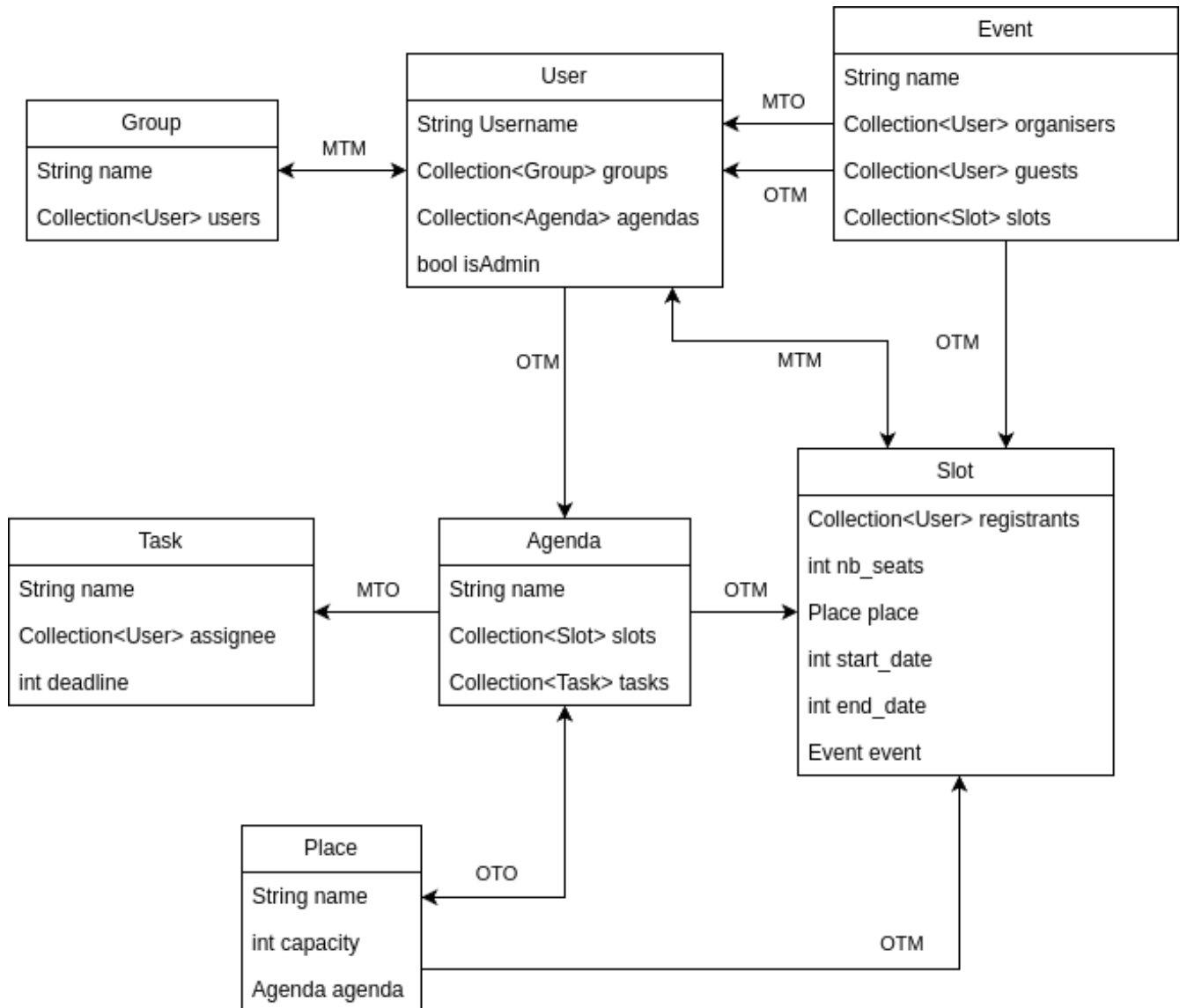


Figure 3: Diagramme des nouveaux EJB