

58, bd Charles Livon -13284 Marseille Cedex 07

Standard téléphonique : 04 91 39 65 00

www.univ-amu.fr

Projet d'évaluation

M1 EN INFORMATIQUE

Jury :

Mr. Jean-luc-Massat

Binôme :

- **FELLOUSSI Idriss**

- **SMAIL Aghilas**

1. Les identifiants de connexion:

username : contact@email.fr
password : password

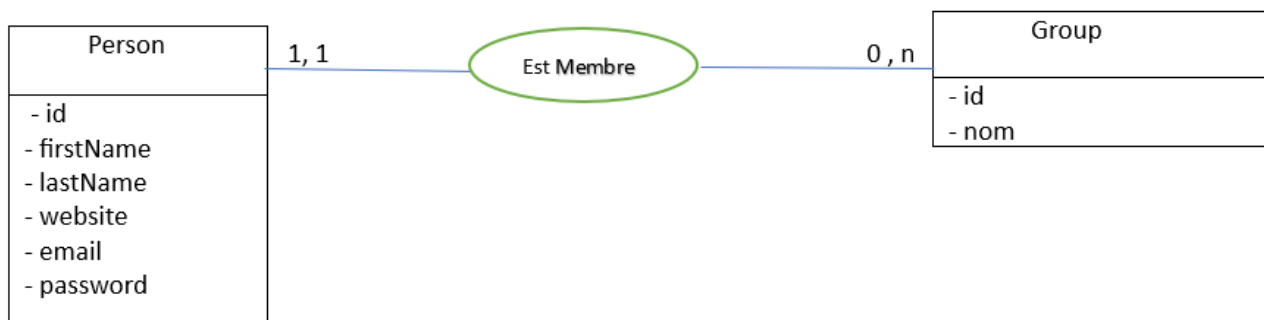
2. Les outils utilisés :

L'application web a été faite pour mettre en œuvre la majorité des concepts et technologies vu lors de cette UE notamment avec JEE , Hibernate, Spring-Boot, Spring-mvc,JUnit , JSTL ...

L'application est déployée sur un serveur Tomcat .

3. L' accès aux données :

3.1 Modèle Entités/Relations :



3.2 La couche REPO :

3.2.1. L'interface:

L'accès aux données doit implémenter les méthodes pour:

- . Récupérer les entités en fonction de leur identifiant,
- . Récupérer toutes les entités d'un certain type,
- . Récupérer toutes les personnes d'un certain groupe,

3.2.2. L'implémentation

La couche Repository accède à la base de données avec le JPA, et le L'ORM Hibernate a été choisi pour les requêtes.

3.3. Le service Directory Manager :

En fonction de l'utilisateur, les accès aux données ne sont pas les mêmes, ce service gère la logique des accès aux données et, par mesure de sécurité supplémentaire, nettoie les données qui ne devrait pas être accessible à l'utilisateur courant.

Pour fonctionner le service doit savoir qui est actuellement authentifié dans la session courante.

4. Les routes :

Toutes les routes sont éditées au même endroit, les autres composants de l'application doivent s'en servir pour l'affectation de toutes variables se rapportant à une route.

Ainsi les routes peuvent être modifiées très facilement sans affecter le fonctionnement du site.

5. Chargement de données automatique:

Pour tester le site web plus facilement des données automatiques sont générées lors de la configuration du repository.

5.1. La classe DirectoryManager

La classe DirectoryManager crée des données automatiques.(via la méthode init)

Elle a deux paramètres : le nombre de personnes à créer et le nombre de groupes à créer.

Ensuite les personnes créées ont des noms / prénoms aléatoires, un mail égal a 'mail@<id>' ou id est l'identifiant de la personne dans la base de données, un mot de passe égal a : 'pass<id>'. Elles sont affectées aléatoirement dans un groupe.

6 . Les contrôleurs :

6.1. Association des fonctions aux routes.

Pour être sûr d'associer leurs fonctions aux bonnes routes, ils utilisent le Mapper.

6.2. Les paramètres globaux

Certains paramètres peuvent potentiellement être traités par plusieurs routes comme :

- . 'id' : Identifiant de l'entité concernée par la route,

6.3. Revenue des contrôleurs

6.3.1 MyControler :

Il traite les routes qui font office de page d'accueil.

6.3.2 MyGroupeController :

Il traite les actions liées aux groupes, à savoir:

- . la visualisation d'un groupe,
- . Afficher la liste des groupes.

- . Ajouter un groupe.
- . Modifier un groupe.
- . Suppression d'un groupe
- . Le listing de tous les groupes de l'annuaire.

6.3.3 MyUserController :

Il traite les actions liées aux personnes , à savoir :

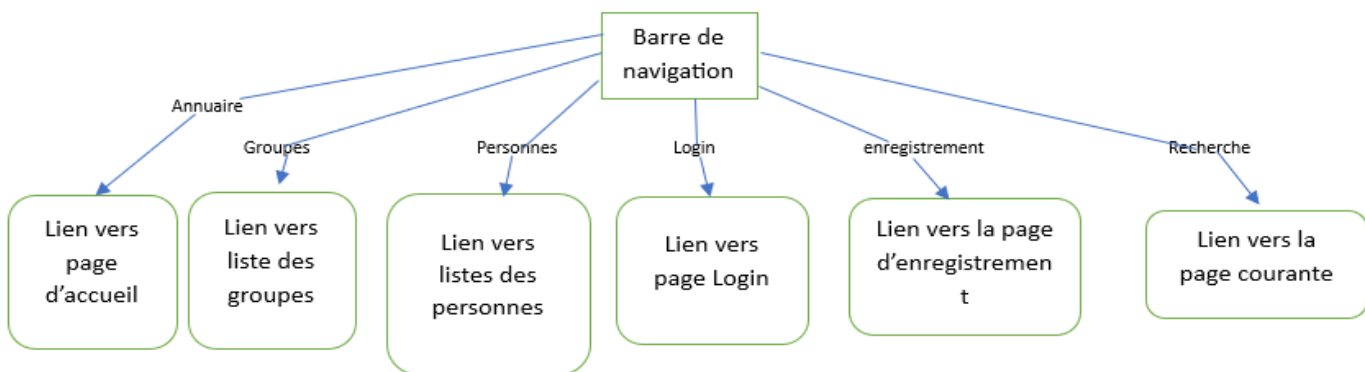
- . L'édition d'une personne,
- . La visualisation d'une personne.
- . l'enregistrement d'une personne.
- . Le listing de toutes les personnes de l'annuaire.

7 . Les vues :

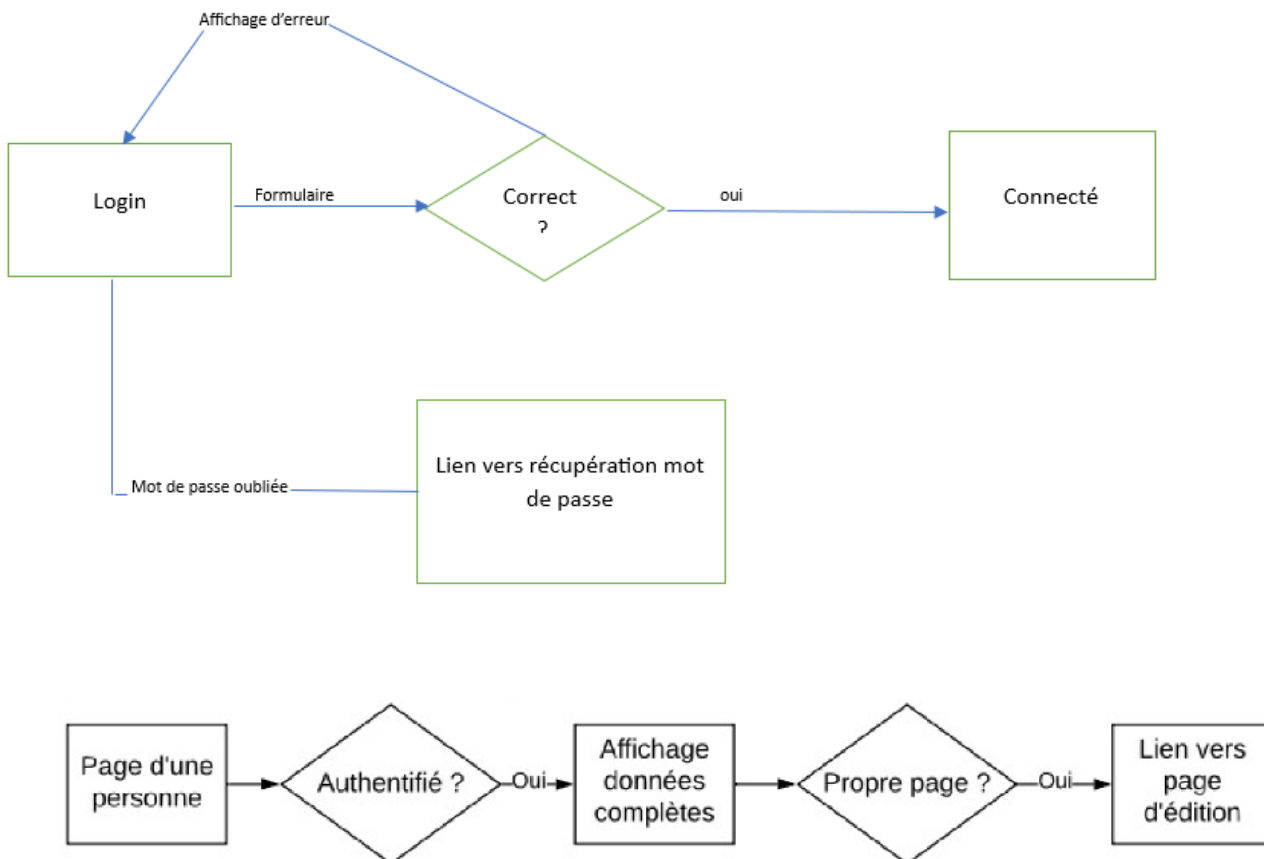
7.1. Framework Bootstrap :

Pour avoir un prototype présentable pour l'application, le framework Bootstrap accompagné avec du CSS.

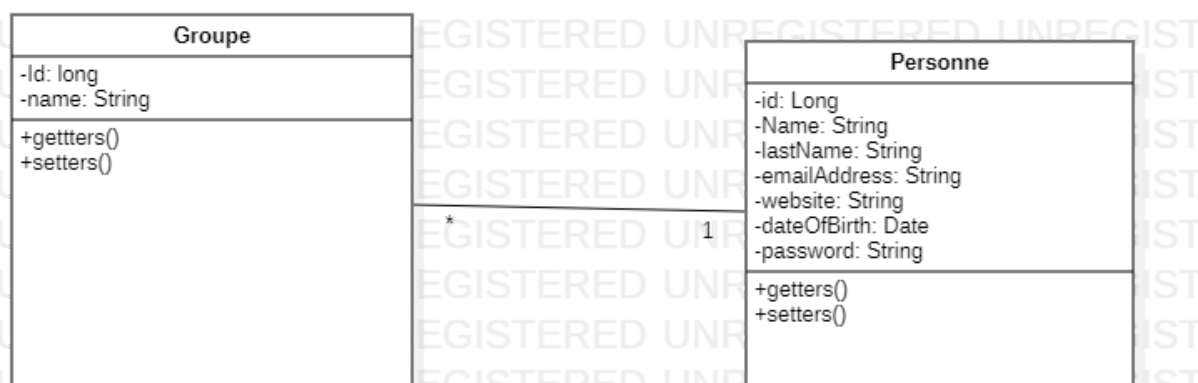
8. Diagramme de navigation :



- Procédure de connexion :



8.1 . Diagramme de classe personne et groupe :



9 . Tests Réalisés :

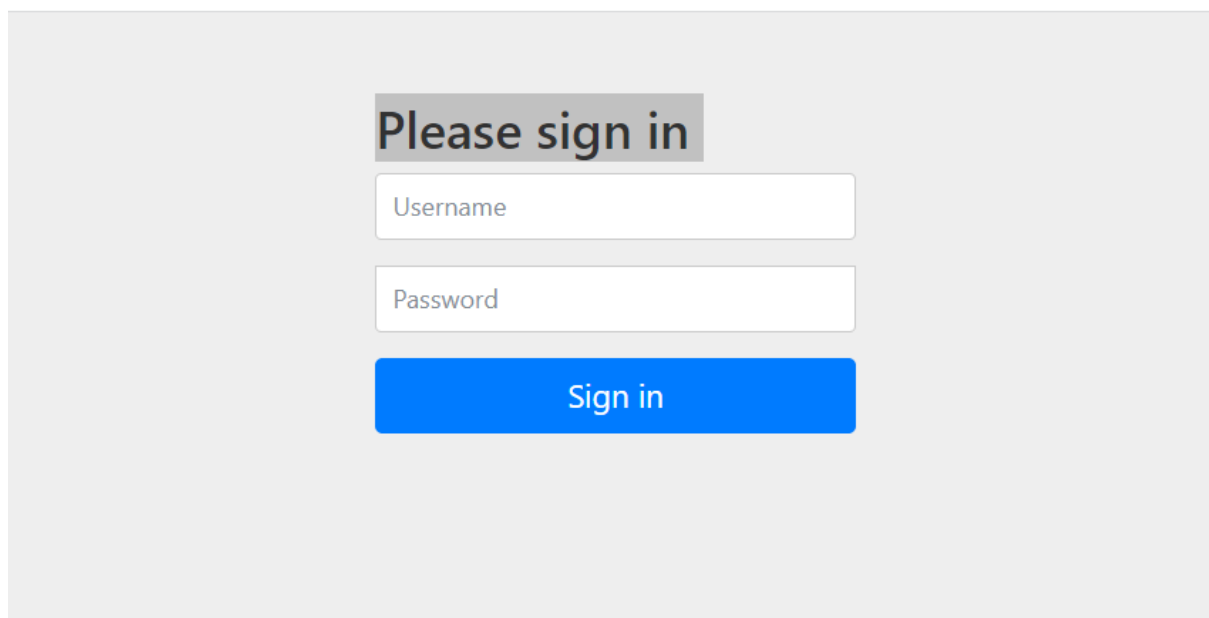
9.1 Tests Repository :

On a effectué 3 tests de base pour la génération des personnes et des groupes.

Tous les tests sauvegardent le jeu de données avec le repository et vérifient les résultats en appelant les fonctions de recherche de repository.

10 Annexes :

10.1. Page de connexion :



Please sign in

Username

Password

Sign in

10.2. Page d'enregistrement :


Enregistrement

Nom :

Prenom :

Adresse email :

Site web :

Date de naissance : 

Mot de passe :

Groupe : 

[Sing up](#)

10.3. Page liste des groupes :

Liste des groupes

ID	Nom	Operations
1	groupe 1	Modifier Supprimer
20	groupe 2	Modifier Supprimer
40	groupe 3	Modifier Supprimer
59	groupe 4	Modifier Supprimer
72	groupe 5	Modifier Supprimer
84	groupe 6	Modifier Supprimer
103	groupe 7	Modifier Supprimer
121	groupe 8	Modifier Supprimer
140	groupe 9	Modifier Supprimer

10.4 . Liste des personnes :

Liste des personnes

Nom	Prenom	Groupe	email
uzumaki1	Naruto1	groupe 1	contactNaruto1@email.com
uzumaki2	Naruto2	groupe 1	contactNaruto2@email.com
uzumaki3	Naruto3	groupe 1	contactNaruto3@email.com
uzumaki4	Naruto4	groupe 1	contactNaruto4@email.com
uzumaki5	Naruto5	groupe 1	contactNaruto5@email.com
uzumaki6	Naruto6	groupe 1	contactNaruto6@email.com
uzumaki7	Naruto7	groupe 1	contactNaruto7@email.com
uzumaki8	Naruto8	groupe 1	contactNaruto8@email.com
uzumaki9	Naruto9	groupe 1	contactNaruto9@email.com
uzumaki10	Naruto10	groupe 1	contactNaruto10@email.com

10.5. Page de modification :

Modifier mon Profil

Nom :

Prenom :

email :

Site web :

Date de naissance :
 

Nouveau mot de passe :

Groupe :

10.6 . Page de logout :

Please sign in

You have been signed out

Sign in

10.7 . Suite de tests

Nom de la fonction test	jeu de données	Résultat attendu
testPerson	1-Ajouter une personne a un groupe 2- Recherche une personne dans un groupe. 3- Rechercher à partir de prénom	1- L'ajout de la personne dans le groupe 2- Rechercher une personne dans un groupe 3- afficher une personne à partir de son prénom.
testGroupe	1- Ajout un groupe 2-chercher un groupe 3-Modifier un groupe	1- Création d'un group 2- Faire une recherche sur le groupe 3- effectuer une modification sur le groupe