

TABLE DES MATIERES

Rapport du Projet de site web	2
1 Spécifications	2
1.1 Objectif du site	2
1.2 Acteurs.....	2
1.3 User stories	2
1.4 Contraintes non fonctionnelles :	2
1.5 Mockup	3
2 Base de données	5
2.1 Diagramme entité relation	5
2.2 Schéma relationnel.....	5
2.3 Description des tables et des relations entre tables	5
2.4 Description précise des champs	5
3 Symfony	7
3.1 Créer un post	7
3.2 Lister et trier les posts par catégories, demande et offre d'emploi	8
3.3 Modifier mes posts.....	9
3.4 Supprimer mon post.....	10
3.5 Rechercher un post par son id.....	11
3.6 M'inscrire sur le site	12
3.7 Me connecter sur le site	13
3.8 Modifier mes informations.....	14
3.9 Supprimer mon compte.....	15
4 API Rest	17
4.1 Tableau récapitulatif.....	17
4.2 Détails des requêtes	17
5 Angular	21
5.1 Créer un post	21
5.2 Lister et trier les posts par catégories, demande et offre d'emploi	22
5.3 Modifier mes posts.....	23
5.4 Supprimer mon post.....	24
5.5 Rechercher un post par son id.....	25
5.6 M'inscrire sur le site	26
5.7 Me connecter sur le site	27
5.8 Modifier mes informations.....	28
5.9 Supprimer mon compte.....	29

6	Conclusion.....	31
6.1	Difficultés rencontrés	31
6.2	Objectifs atteints/non-atteints	31
6.3	Piste d'amélioration.....	32
7	Bibliographie	33

RAPPORT DU PROJET DE SITE WEB

1 SPÉCIFICATIONS

1.1 OBJECTIF DU SITE

Le site doit permettre à un utilisateur de proposer ou de rechercher des offres d'emploi sous forme de petites annonces et de trier celle-ci par catégories et offre/demande. Il faudra avoir un compte pour voir les informations de contact.

1.2 ACTEURS

- Utilisateur : personne qui utilise le site

2 autres sous-classes d'utilisateurs héritent de celui-ci et ont accès à tout le site

- Demandeur d'emploi
- Proposeur d'emploi

1 autre sous-classe qui n'a pas accès à tout le site

- Guest

1.3 User stories

Priorité En tant qu'utilisateur, je veux pouvoir créer des catégories afin de classer les notes

En tant qu'utilisateur, je veux pouvoir créer une offre ou demande d'emploi via un formulaire.	1
En tant qu'utilisateur, je veux pouvoir lister les différentes propositions par catégorie et offre ou demande.	1
En tant qu'utilisateur, je veux pouvoir modifier la fiche que j'ai posté.	1
En tant qu'utilisateur, je veux pouvoir supprimer sa fiche.	1
En tant qu'utilisateur, je veux pouvoir rechercher une fiche.	1
En tant qu'utilisateur, je veux pouvoir me connecter sur le site.	1
En tant qu'utilisateur, je veux pouvoir m'inscrire sur le site via un formulaire.	2
En tant qu'utilisateur, je veux pouvoir modifier les informations de mon compte.	2
En tant qu'utilisateur, je veux pouvoir supprimer mon compte.	2

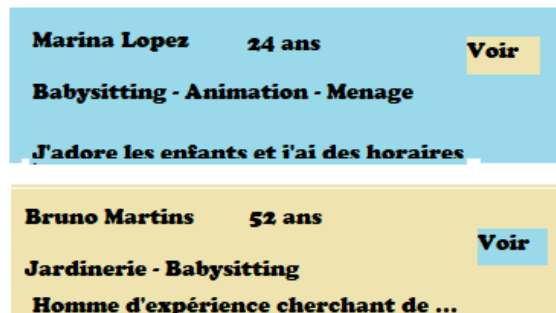
1.4 CONTRAINTES NON FONCTIONNELLES :

Le site a le devoir d'être responsif et ergonomique.

1.5 MOCKUP

1.5.1 LISTE DE DEMANDE D'EMPLOI

Demande de travail : Babysitting



1.5.2 PAGE COMPTE

1.5.3 EDITION DUNE NOTE

OFFRE PROPOSITION RECHERCHER BIENVENU THIBAUT

EDITION

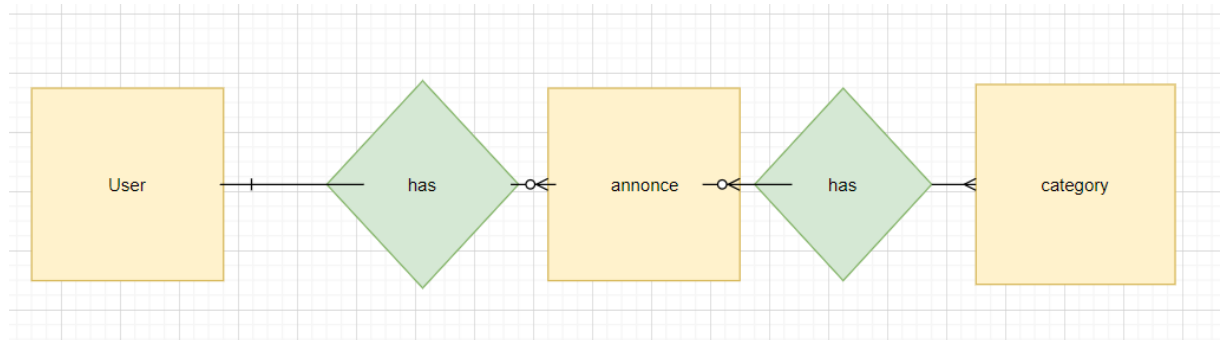
CATEGORIE :

**PRECISION
PERSONNELLE :**

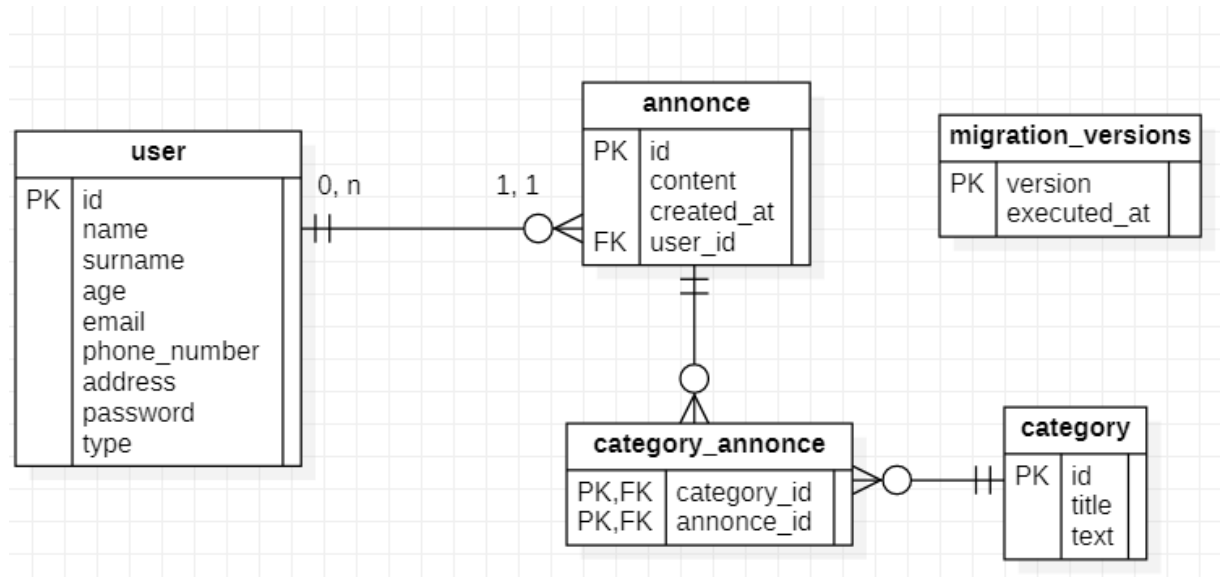
ENREGISTRER

2 BASE DE DONNÉES

2.1 DIAGRAMME ENTITÉ RELATION



2.2 SCHÉMA RELATIONNEL



2.3 DESCRIPTION DES TABLES ET DES RELATIONS ENTRE TABLES

Table	Table	Type	Description
User	Annonce	One to many	Chaque annonce a 1 utilisateur et un utilisateur peut avoir 0 ou n annonce.
Annonce	Category	Many to many	Une catégorie peut avoir 0 ou n annonces et une annonce peut avoir 1 ou n catégories.

2.4 DESCRIPTION PRÉCISE DES CHAMPS

2.4.1 TABLE : USER

Nom	Key	Non null	Type	Référence	Intitulé	Commentaire
id	PK	O	int		Identifiant	
name		O	varchar(255)		Prénom	
surname		O	varchar(255)		Nom de famille	

age		O	int		Age	
email		O	varchar(255)		Email	
phone_number		O	varchar(255)		Numéro de téléphone	
address		O	varchar(255)		Adresse	
password		O	varchar(255)		Mot de passe	Crypter grâce à bcrypt
type		O	varchar(255)		Type de compte	Employeur ou chercheur d'emploi

2.4.2 TABLE : ANNONCE

Nom	Key	Non null	Type	Référence	Intitulé	Commentaire
id	PK	O	int		Identifiant	
content		O	text		Contenu	
created_at		O	varchar(255)		Heure de création de l'annonce	
user_id	FK	O	int	id/user	Lien avec user	

2.4.3 TABLE : CATEGORY

Nom	Key	Non null	Type	Référence	Intitulé	Commentaire
id	PK	O	int		Identifiant	
title		O	varchar(255)		Nom de la catégorie	
text		O	text		Description de la catégorie	Non-utilisé dans le site, juste là pour une future implémentation

2.4.4 TABLE : CATEGORY_ANNONCE

Nom	Key	Non null	Type	Référence	Intitulé	Commentaire
category_id	PFK	O	int	id/category	Catégorie	Les deux peuvent servir de clé primaire
annonce_id	PFK	O	int	id/annonce	Annonce	

3 SYMFONY

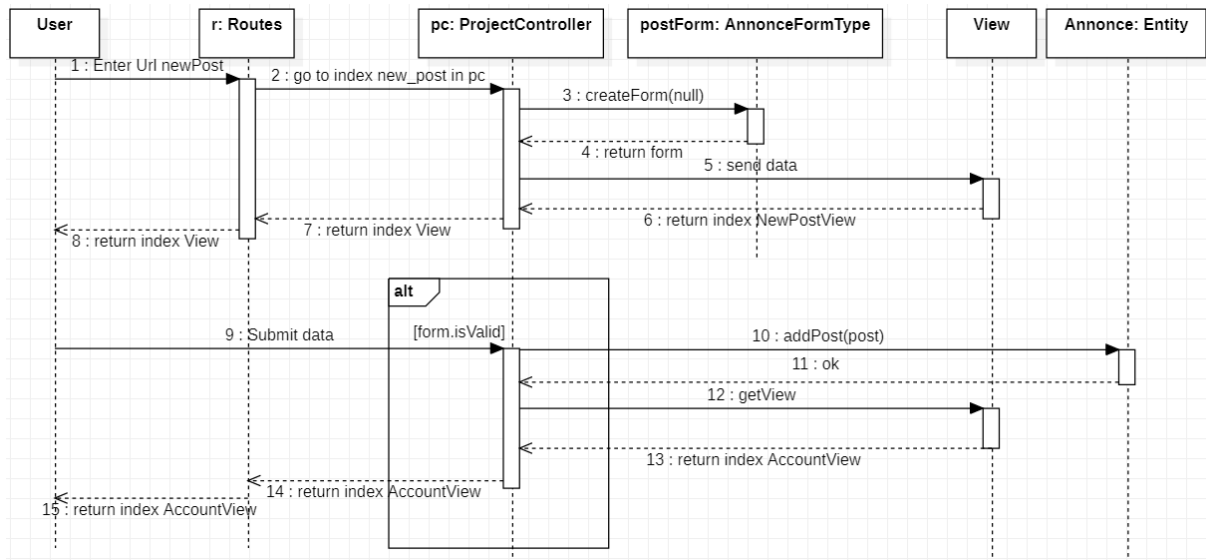
Le rapport étant déjà assez lourd comme cela, certains diagrammes de séquence seront simplifiés.

3.1 CRÉER UN POST

3.1.1 EXPLICATION DE LA FONCTIONNALITÉ

1. Cliquez sur « Add a post » dans « My account »
2. Le formulaire apparaît
3. Complétez-le
4. Cliquez sur « Add my post ! »
5. Vous êtes redirigé vers « My account »

3.1.2 DIAGRAMME DE SÉQUENCE



3.1.3 SÉCURITÉ ET GESTION D'ERREUR

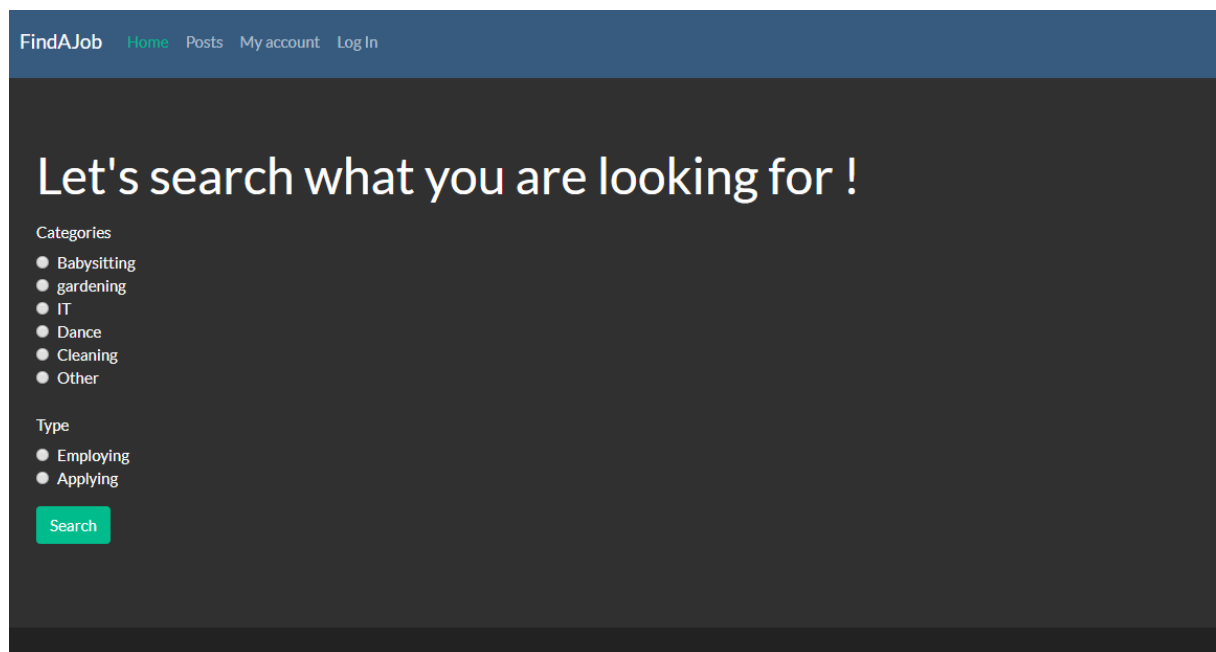
- On doit cocher au moins une catégorie
- Le contenu doit contenir au moins 20 caractères
- Il faut être connecté

3.1.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Controller/ProjectController.php
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Entity/Annonce.php
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Form/AnnonceFormType.php
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/templates/project/newPost.html.twig

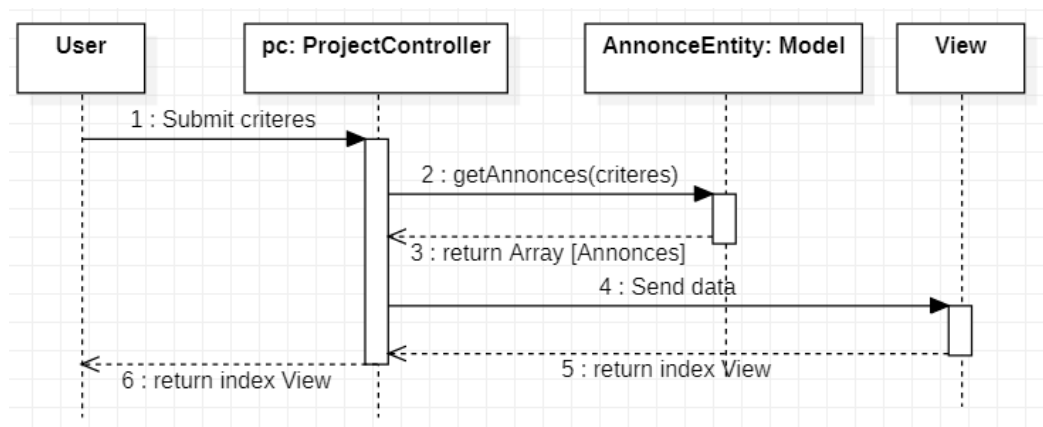
3.2 LISTER ET TRIER LES POSTS PAR CATÉGORIES, DEMANDE ET OFFRE D'EMPLOI

3.2.1 EXPLICATION DE LA FONCTIONNalité



1. Cochez vos critères de recherches dans le formulaire
2. Cliquez sur « Search ».
3. Vos recherches s'affichent après un rechargement de la page

3.2.2 DIAGRAMME DE SÉQUENCE



3.2.3 SÉCURITÉ ET GESTION D'ERREUR

- On ne pourra cliquer sur les détails d'une annonce qu'une fois connecter

3.2.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

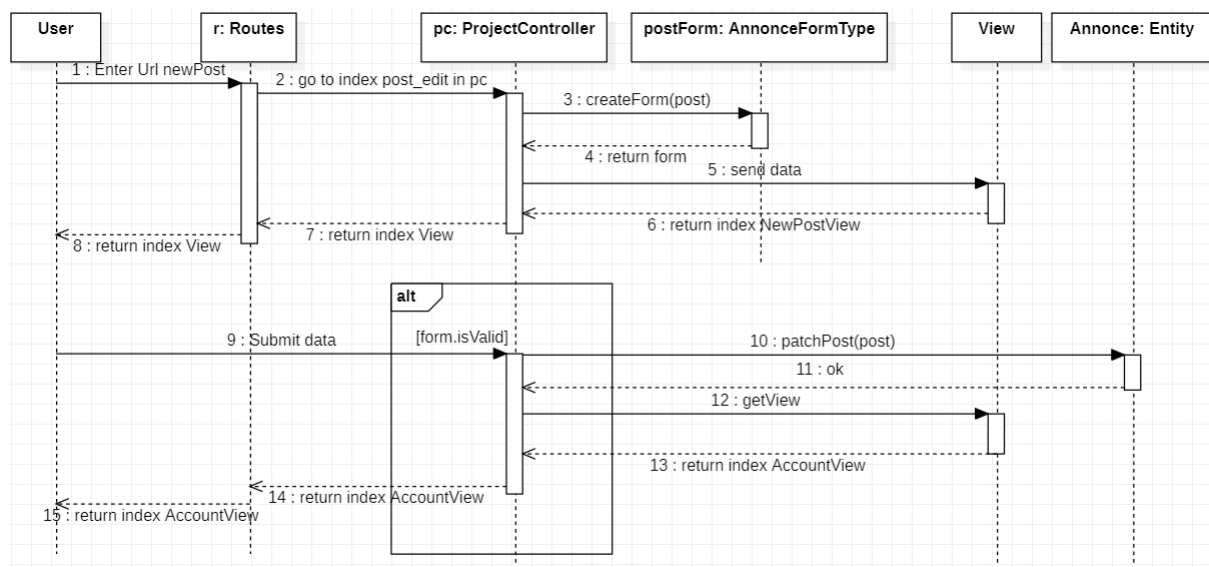
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Controller/ProjectController.php
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Form/SearchType.php
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/templates/project/search.html.twig

3.3 MODIFIER MES POSTS

3.3.1 EXPLICATION DE LA FONCTIONNALITÉ

- L'utilisateur clique sur edit en bas d'une de ses annonces
- Le formulaires pré-encodé apparait
- Il modifie ce qu'il doit modifier
- Il clique sur « Save »
- Le formulaire disparaît et il est redirigé vers

3.3.2 DIAGRAMME DE SÉQUENCE



3.3.3 SÉCURITÉ ET GESTION D'ERREUR

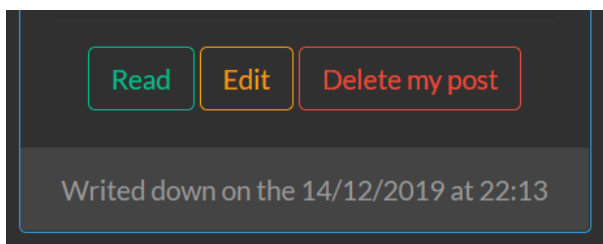
- On doit cocher au moins une catégorie
- Le contenu doit contenir au moins 20 caractères
- On ne peut modifier que ses annonces

3.3.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/templates/project/newPost.html.twig
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Controller/ProjectController.php
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Entity/Annonce.php
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Form/AnnonceFormType.php

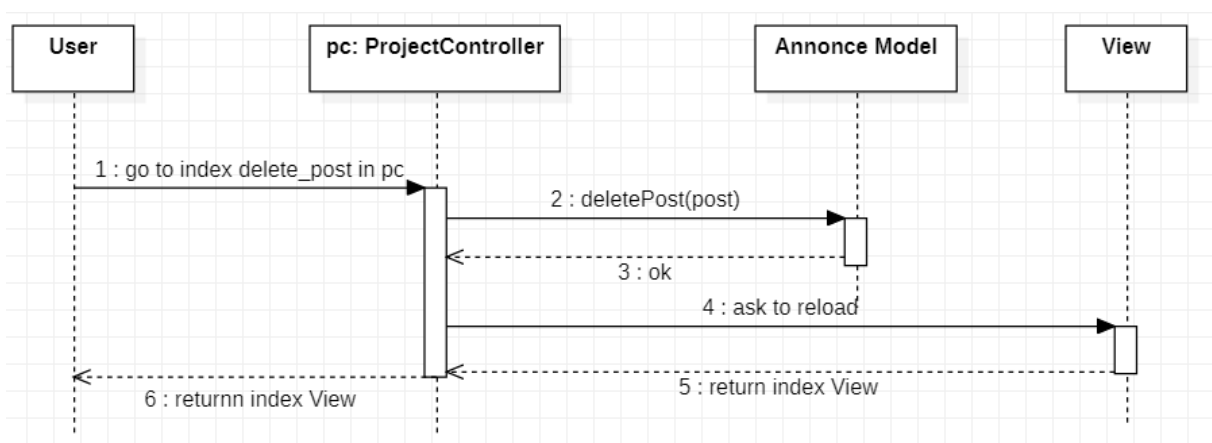
3.4 SUPPRIMER MON POST

3.4.1 EXPLICATION DE LA FONCTIONNALITÉ



1. L'utilisateur clique sur « Delete my post » en bas d'une annonce
2. L'annonce est supprimée de la base de données
3. L'annonce disparaît de la vue

3.4.2 DIAGRAMME DE SÉQUENCE



3.4.3 SÉCURITÉ ET GESTION D'ERREUR

- On ne peut supprimer que ses annonces

3.4.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Controller/ProjectController.php

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Entity/Annonce.php

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/templates/project/Annonce.html.twig

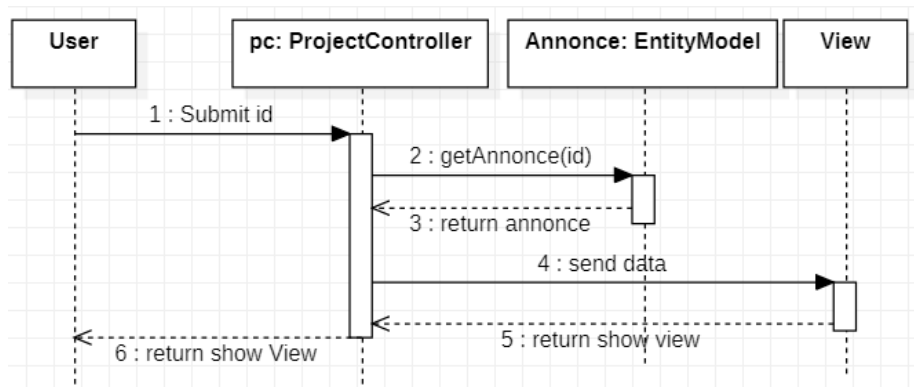
3.5 RECHERCHER UN POST PAR SON ID

3.5.1 EXPLICATION DE LA FONCTIONNalité



1. Entrez l'id d'une annonce dans la barre de recherche
2. Cliquez sur « search by Id »
3. Votre annonce s'affiche dans une vue (vous devez être connecté)

3.5.2 DIAGRAMME DE SÉQUENCE



3.5.3 SÉCURITÉ ET GESTION D'ERREUR

- On ne peut entrer qu'un int
- Il faut être connecté

3.5.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Controller/ProjectController.php
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/templates/base.html.twig
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/templates/project/show.html.twig

3.6 M'INSCRIRE SUR LE SITE

3.6.1 EXPLICATION DE LA FONCTIONNALITÉ

Register on FINDJOB

Name

Surname

Type

- ☐ Employing
- ☐ Applying

Age

Phone number

Address

Email

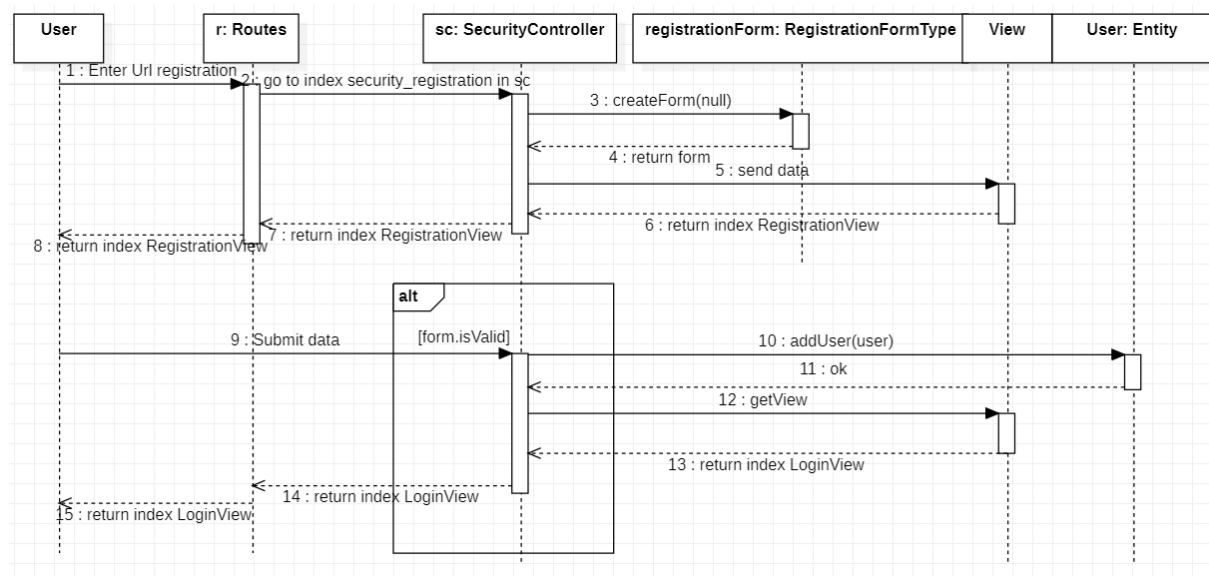
Password

Confirm password

[Register now !](#)

1. L'utilisateur encode ses informations
2. Il clique sur Save
3. Si les informations sont valides il est redirigé vers login

3.6.2 DIAGRAMME DE SÉQUENCE



3.6.3 SÉCURITÉ ET GESTION D'ERREUR

- Le mot de passe doit contenir au moins 6 caractères
- Il y a une confirmation du mot de passe
- L'email doit être un email valide et pas déjà présent dans la base de données
- Chaque champ doit être remplis

3.6.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

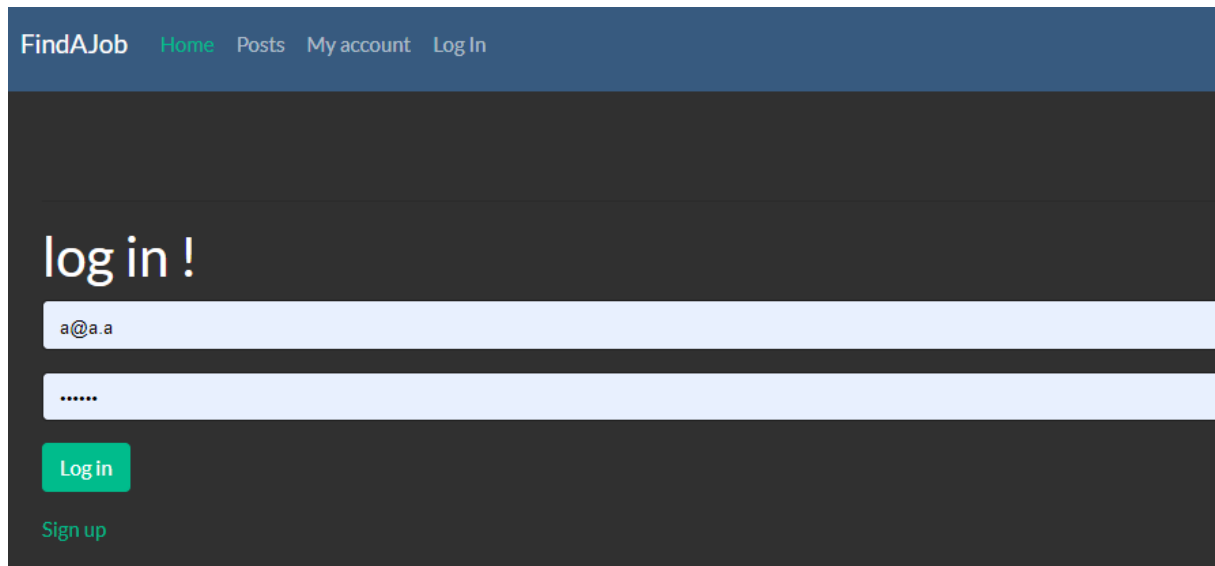
https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Controller/SecurityController.php

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Form/RegistrationFormType.php

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Entity/User.php

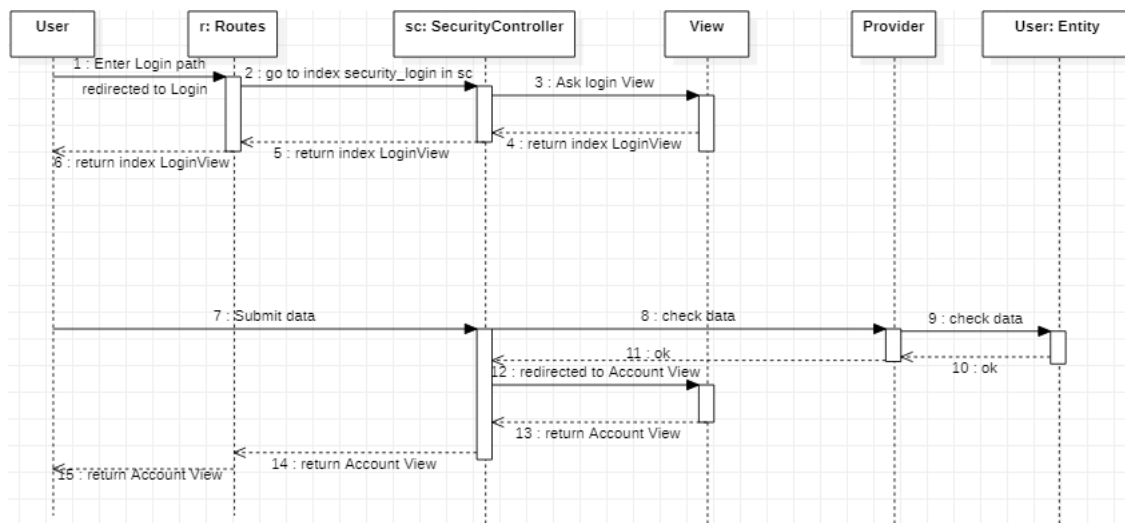
3.7 ME CONNECTER SUR LE SITE

3.7.1 EXPLICATION DE LA FONCTIONNALITÉ



1. J'entre mon identifiant et mot de passe
2. Je clique sur login (ou « Sign up » si je n'ai pas de compte)
3. Je suis redirigé vers mon compte

3.7.2 DIAGRAMME DE SÉQUENCE



3.7.3 SÉCURITÉ ET GESTION D'ERREUR

- Le mot de passe est flouté
- Les données sont vérifiées dans la base de données

3.7.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Controller/SecurityController.php

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/templates/security/login.html.twig

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/config/packages/security.yaml

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Entity/User.php

3.8 MODIFIER MES INFORMATIONS

3.8.1 EXPLICATION DE LA FONCTIONNALITÉ

Register on FINDJOB

Name: Delvaux Pierre

Surname: Pierre

Type: ☐ Employing ☒ Applying

Age: 1

Phone number: 0473561825

Address: 14 avenue des vieux amis

Email: x@x

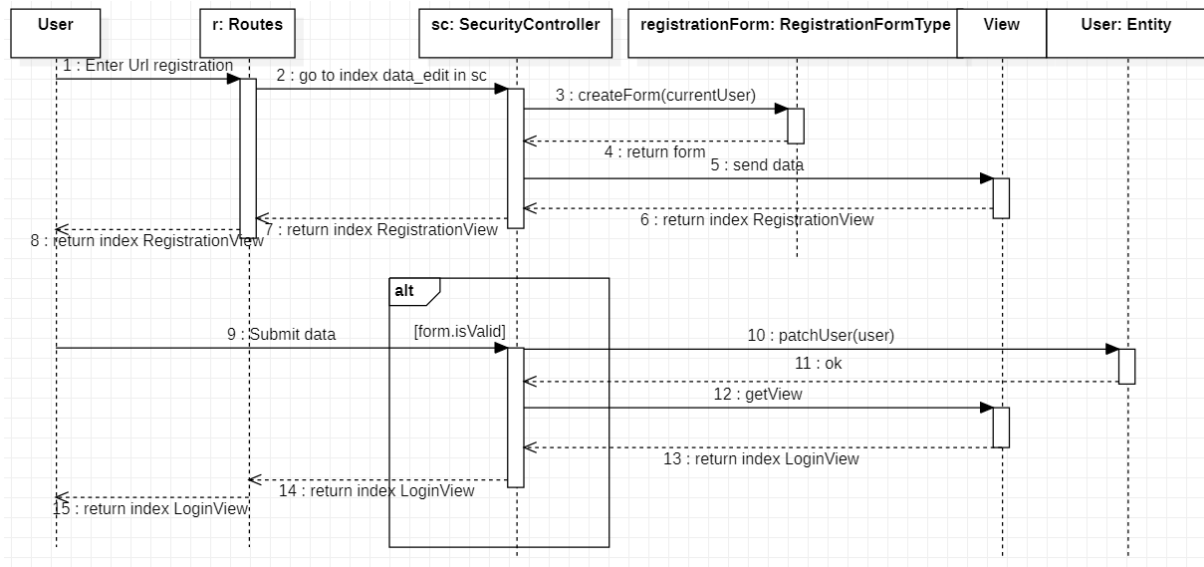
Password: [masked]

Confirm password: [masked]

Save

- 1 L'utilisateur modifie ses informations
- 2 Il clique sur Save
- 3 Si les informations sont valides il est redirigé vers login

3.8.2 DIAGRAMME DE SÉQUENCE



3.8.3 SÉCURITÉ ET GESTION D'ERREUR

- Le mot de passe doit contenir au moins 6 caractères
- Il y a une confirmation du mot de passe
- L'email doit être un email valide et pas déjà présent dans la base de données
- Chaque champ doit être remplis

3.8.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

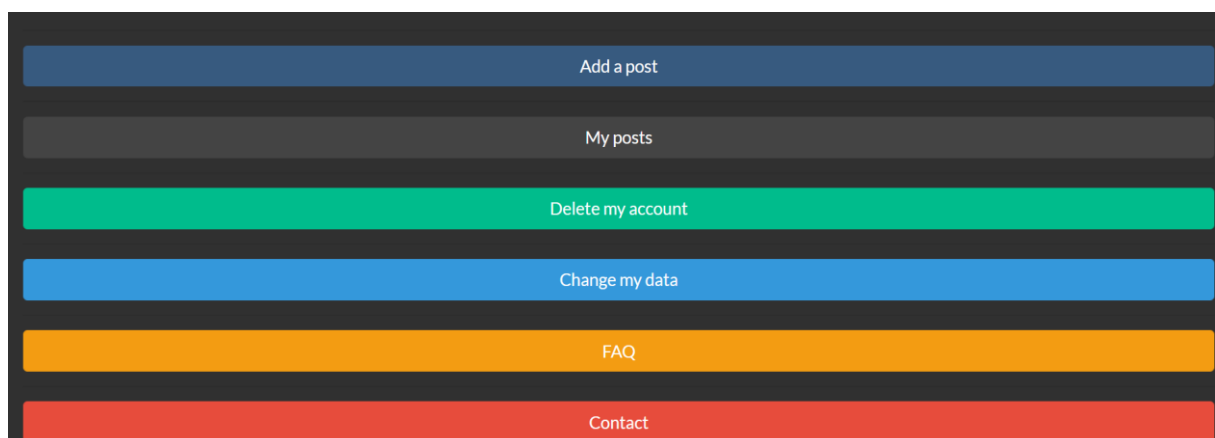
https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Controller/SecurityController.php

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Form/RegistrationFormType.php

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Entity/User.php

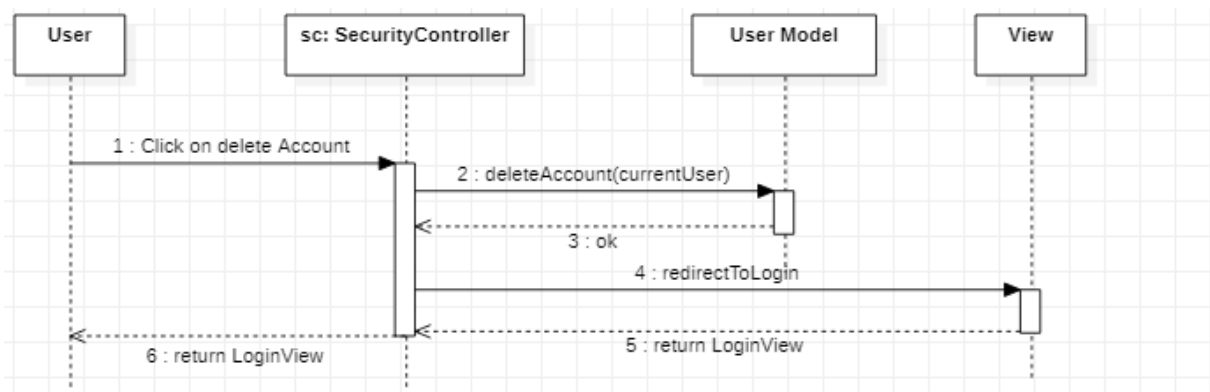
3.9 SUPPRIMER MON COMPTE

3.9.1 EXPLICATION DE LA FONCTIONNALITÉ



- 1 L'utilisateur clique sur « Delete my account »
- 2 Le compte est supprimé de la base de données
- 3 L'utilisateurs est dirigé vers login

3.9.2 DIAGRAMME DE SÉQUENCE



3.9.3 SÉCURITÉ ET GESTION D'ERREUR

- On ne peut évidemment que supprimer son propre compte

3.9.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Controller/SecurityController.php

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Entity/User.php

4 API REST

Pour créer mon api j'ai utilisé l'extension Symfony (fait par d'autres développeurs) API-platform qui permet de créer des requêtes avec un système d'annotations dans les entités de Symfony. Les données envoyées sont sous format json.

C'est un framework puissant mais facile à utiliser, dédié aux projets pilotés par API. Il est très utilisé dans l'industrie web. Il contient une bibliothèque PHP pour créer des API complètes prenant en charge les normes leaders de l'industrie (JSON-LD et Hydra, Graph, OpenAPI ...), fournit des outils JavaScript ambitieux pour consommer ces API en un clin d'œil (admin, PWA et générateurs d'applications mobiles, client hypermédia ...) et est livré avec une belle intégration Docker et Kubernetes pour développer et déployer instantanément sur le cloud.

4.1 TABLEAU RÉCAPITULATIF

L'url commence toujours avec le chemin suivant : <http://localhost:8000/api>

https://github.com/Horsefactor/Architecture_Web/blob/master/demo/config/routes/api_platform.yaml

Méthode http	Url (requêtes)	Action
GET	../annonces	Donne toutes les annonces
DELETE	../annonces/{id}	Supprime une annonce grâce à son id
PATCH	../annonces/{id}, body	Met à jour une annonce grâce à son id, body représente les données
POST	../annonces/{id}, body	Publie une annonce dans la base de données grâce à son id, body représente les données
GET	../annonces/{id}	Obtient une annonce grâce à son id
GET	../users/{id}/annonces	Obtient les annonces d'un utilisateur par son id
GET	../categories	Obtient toutes les catégories
GET	../categories/{id}	Obtient une catégorie grâce à son id
POST	../categories/{id}, body	Publie une catégorie grâce à son id, body représente les données
PATCH	../categories/{id}, body	Met à jour une catégorie grâce à son id, body représente les données
DELETE	../categories/{id}	Supprime une catégorie grâce son id
GET	../users	Renvoi tous les utilisateurs
GET	../users/{id}	Obtient un utilisateur grâce à son id
POST	../users/{id}, body	Publie un utilisateurs grâce à son id, body représente les données
PATCH	../users/{id}, body	Met à jour un utilisateurs grâce à son id, body représente les données
DELETE	../users/{id}	Supprime un utilisateur grâce à son id
POST	../authentication_token/{email, password}	Renvoi un JWT et les informations de l'utilisateurs si l'email et le mot de passe correspond à l'utilisateur (sauf l'adresse et le mot de passe par sécurité)

4.2 DÉTAILS DES REQUÊTES

- Chaque requête envoie ses données sous format json.
- Quand la requête nécessite un id celui-ci est évidemment un nombre.
- Quand on veut faire une référence vers un objet dans les données json transmises dans la requête api-platform procède comme l'exemple qui suit :
Exemple : « categories » : [« /api/categories/1 », « /api/categories/2 »]
Ça permet de référencer directement les catégories par exemple sans devoir passer tout l'objet.
- Lorsqu'on veut récupérer une collection d'objet les données en réponses prendront le format suivant :

```
{
  "@context": "/api/contexts/Annonce",
  "@id": "/api/annonces",
  "@type": "hydra:Collection",
  "hydra:member": [ *ici se situe la liste des objets renvoyés dont la structure est expliqué plus bas* ]
  "hydra:totalItems": 6
}
```

4.2.1 REQUÊTES D'ANNONCES

Pour les méthodes POST, PATCH des annonces, le body doit être comme suit :

```
{
  "content": "string",
  "createdAT": "2020-01-04T22:21:11.348Z",
  "categories": ["string"],
  "user": "string"
}
```

La réponse d'une requête GET ressemblera à ça :

```
{
  "@context": "string",
  "@id": "string",
  "@type": "string",
  "id": 0,
  "content": "string",
  "createdAT": "2020-01-04T22:37:51.296Z",
  "categories": [
    {
      "@context": "string",
      "@id": "string",
      "@type": "string",
      "title": "string"
    }
  ],
  "user": {
    "@context": "string",
    "@id": "string",
    "@type": "string",
    "name": "string",
    "surname": "string",
    "age": 0,
    "email": "string",
    "phoneNumber": "string",
    "type": "string"
  }
}
```

Cette fois-ci j'ai choisi de personnaliser la réponse pour ne pas devoir faire pleins de requêtes lors de l'affichage d'une annonce.

Le service dans angular 8 qui gère les requêtes sur les annonces :

https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/posts/post.service.ts

4.2.2 REQUÊTES D'USERS

Pour les méthodes POST et PATCH des users le body doit être comme suit :

```
{
  "name": "string",
  "surname": "string",
  "age": 0,
  "email": "string",
  "phoneNumber": "string",
  "Annonces": [
    "string"
  ],
  "address": "string",
  "password": "string",
  "type": "string"
}
```

Annonces ne doit pas obligatoirement être présent car un utilisateur peut avoir posté zéro annonce.

La réponse d'une requête GET ressemblera à ça :

```
{
  "@context": "string",
  "@id": "string",
  "@type": "string",
  "id": 0,
  "name": "string",
  "surname": "string",
  "age": 0,
  "email": "string",
  "phoneNumber": "string",
  "Annonces": [
    "string"
  ],
  "address": "string",
  "password": "string",
  "username": "string",
  "type": "string"
}
```

Le service qui gère les requêtes d'utilisateur dans angular 8 :

https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/users/user.service.ts

4.2.3 REQUÊTES DE CATÉGORIES

Pour les méthodes POST et PATCH des catégories le body doit être comme suit :

```
{
  "title": "string",
  "text": "string",
  "Annonces": [
    "string"
  ]
}
```

Annonces ne doit pas obligatoirement être présent car une catégorie peut avoir zéro annonce.

La réponse d'une requête GET ressemblera à ça :

```
{
  "@context": "string",
  "@id": "string",
  "@type": "string",
  "id": 0,
  "title": "string",
  "text": "string",
  "Annonces": [
    "string"
  ]
}
```

Le service qui gère les requêtes de catégorie dans angular 8 :

https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/categories/category.service.ts

4.2.4 REQUÊTE D'AUTHENTIFICATION

Le body de la requête contient un dictionnaire json :

{« email » : « exemple@gmail.be », « password » : « 123456 »}

Et renverra si les informations sont bien dans la base de données :

{« token » : « la token crypté »,

« data » : { **information sur l'utilisateurs**}}

Le service qui gère l'authentification dans angular 8 :

- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/security/auth.service.ts

Fichier.yaml gérant la requête d'authentification dans Symfony 4 :

- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/config/packages/security.yaml
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/config/packages/api_platform.yaml
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/config/packages/lexik_jwt_authentication.yaml
- https://github.com/Horsefactor/Architecture_Web/blob/master/demo/config/routes/annotations.yaml

Chemin des clés privée et public pour générer le JWT :

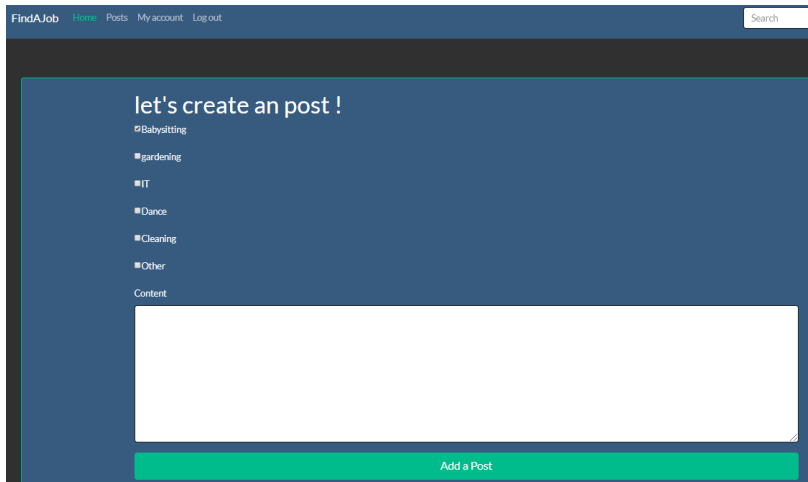
- https://github.com/Horsefactor/Architecture_Web/tree/master/demo/config/jwt

5 ANGULAR

Le rapport étant déjà assez lourd comme cela, certains diagrammes de séquence seront simplifiés.

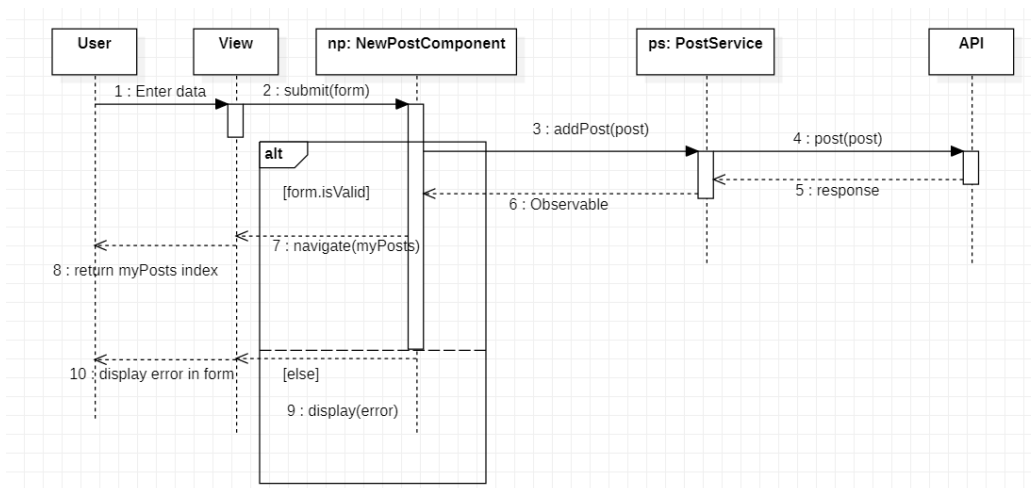
5.1 CRÉER UN POST

5.1.1 EXPLICATION DE LA FONCTIONNALITÉ



1. L'utilisateur choisit ajouter un post dans le menu « My account ».
2. Le formulaire apparaît.
3. L'utilisateur encode le contenu de son annonce et les catégories dans lesquelles il veut que son annonce apparaisse.
4. Il clique sur sauver.

5.1.2 DIAGRAMME DE SÉQUENCE



5.1.3 SÉCURITÉ ET GESTION D'ERREUR

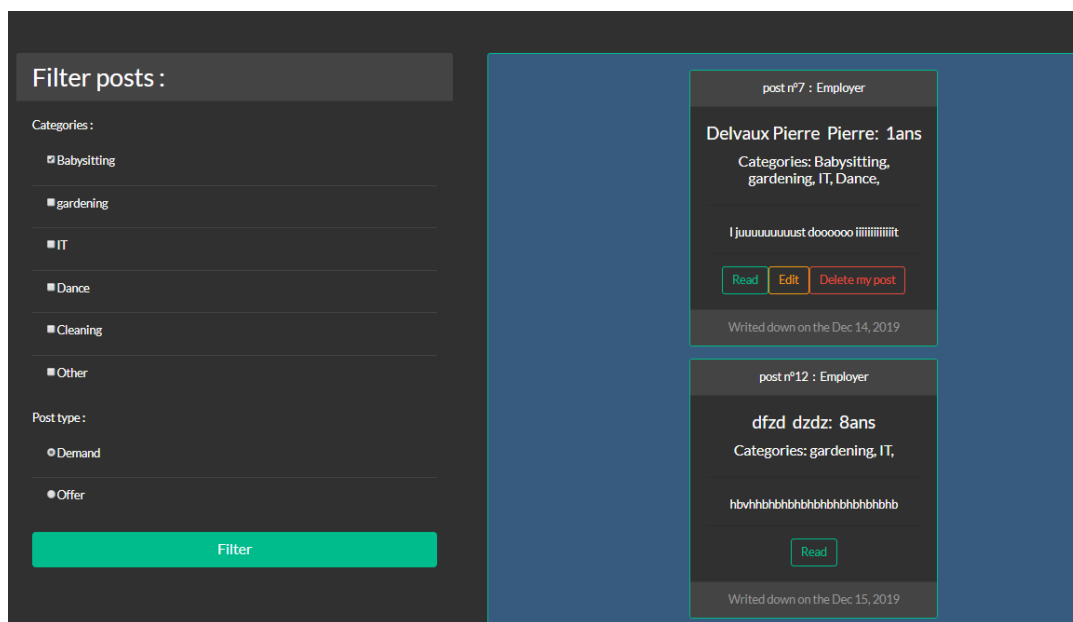
- Accessible que si on est connecté sur un compte
- On doit cocher au moins une catégorie
- Le contenu doit contenir au moins 20 caractères

5.1.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/new-post/new-post.component.html
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/new-post/new-post.component.ts
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/posts/post.service.ts

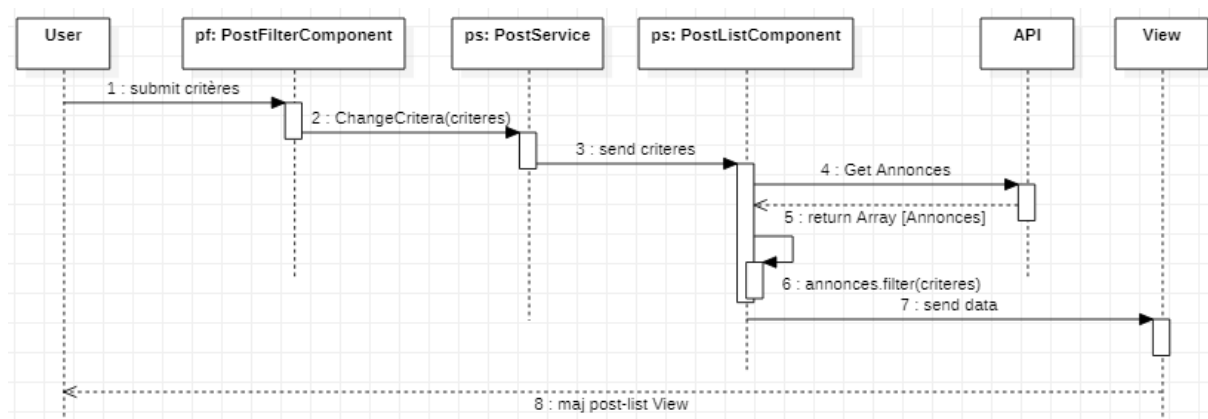
5.2 LISTER ET TRIER LES POSTS PAR CATÉGORIES, DEMANDE ET OFFRE D'EMPLOI

5.2.1 EXPLICATION DE LA FONCTIONNALITÉ



1. Le site montre de base toute les annonces, vous ne pouvez éditer et supprimer que les votre.
2. Cochez vos critères de recherches dans le formulaire de filtre à gauche
3. Cliquez sur « Filter ».
4. Vos recherches s'affichent à droite

5.2.2 DIAGRAMME DE SÉQUENCE



5.2.3 SÉCURITÉ ET GESTION D'ERREUR

- On ne peut pas cliquer sur « Read » si on n'est pas connecté sur un compte car on afficherait les données pour joindre l'auteur de l'annonce
- On ne peut éditer et supprimer que ses annonces
- Par défaut on affiche toutes les annonces

5.2.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/posts/post-service.ts
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/post-filter/post-filter.component.html
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/post-filter/post-filter.component.ts
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/post-list/post-list.component.ts
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/post-list/post-list.component.html

5.3 MODIFIER MES POSTS

5.3.1 EXPLICATION DE LA FONCTIONNALITÉ

let's edit your post !

- ☒ Babysitting
- ☐ gardening
- ☐ IT
- ☐ Dance
- ☐ Cleaning
- ☐ Other

Content

I juuuuuuuust doooooo it

Save

Cancel

1. On clique sur « edit » sur son annonce
2. Le formulaire apparait
3. Les données réapparaissent dedans
4. On change ce que l'on veut changer
5. On clique sur « Save »
6. On est redirigé vers la page de ses posts

5.3.2 DIAGRAMME DE SÉQUENCE

Même diagramme de séquence que pour créer un post excepté que la méthode de la requête vers l'api est patch au lieu de post. En effet le même formulaire est utilisé dans lequel on réinjecte les données de l'annonce sélectionnée.

5.3.3 SÉCURITÉ ET GESTION D'ERREUR

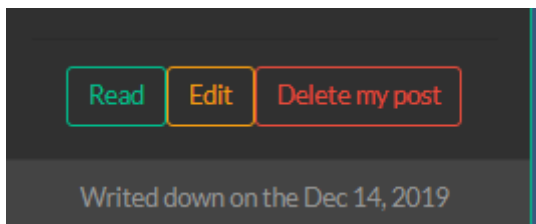
- Accessible que si on est connecté sur un compte et que c'est son post
- On doit cocher au moins une catégorie
- Le contenu doit contenir au moins 20 caractères

5.3.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/new-post/new-post.component.html
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/new-post/new-post.component.ts
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/posts/post.service.ts

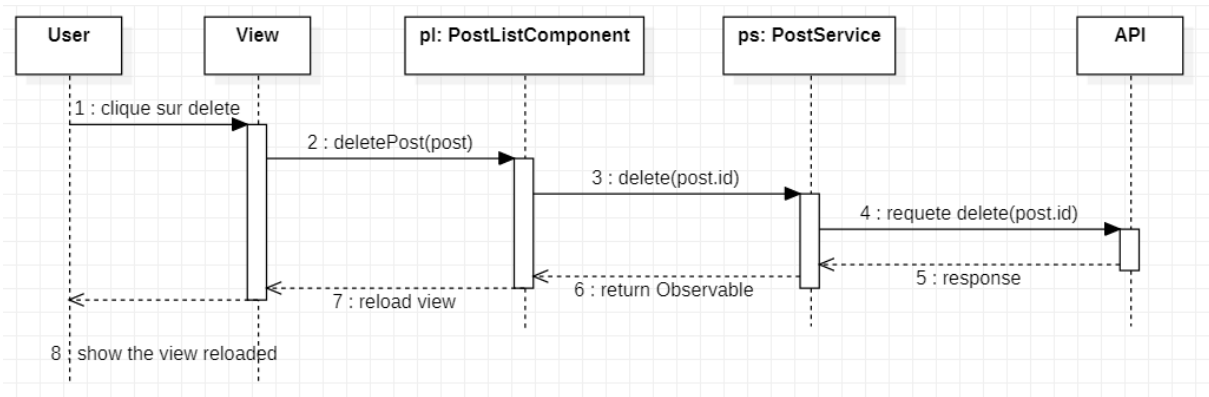
5.4 SUPPRIMER MON POST

5.4.1 EXPLICATION DE LA FONCTIONNALITÉ



1. L'utilisateur clique sur « Delete » en bas de son annonce
2. Le post est supprimé de la base de données
3. La page est rechargée

5.4.2 DIAGRAMME DE SÉQUENCE



5.4.3 SÉCURITÉ ET GESTION D'ERREUR

- On ne peut supprimer que ses posts

5.4.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

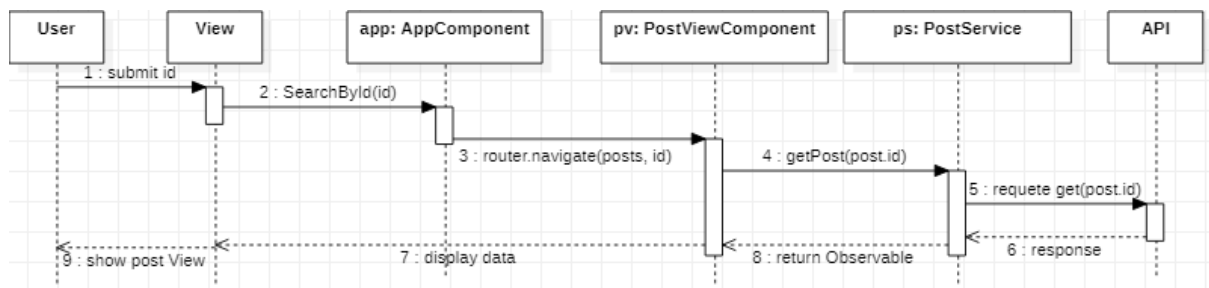
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/post-list/post-list.component.html
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/posts/post.service.ts
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/post-list/post-list.component.ts

5.5 RECHERCHER UN POST PAR SON ID

5.5.1 EXPLICATION DE LA FONCTIONNalité



5.5.2 DIAGRAMME DE SÉQUENCE



4. Entrez l'id d'une annonce dans la barre de recherche
5. Cliquez sur « search by Id »
6. Votre annonce s'affiche dans une vue (vous devez être connecté)

5.5.3 SÉCURITÉ ET GESTION D'ERREUR

- On doit être connecté, car on affiche la vue d'une annonce avec les coordonnées de l'auteur
- On doit taper un entier

5.5.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/post-view/post-view.component.ts
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/post-view/post-view.component.html
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/app.component.ts
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/app.component.html

5.6 M'INSCRIRE SUR LE SITE

5.6.1 EXPLICATION DE LA FONCTIONNALITÉ

FindAJob Home Posts My account Log in

let's create an account !

First name

Second name

Age

Email

Phone number

Address

Account type

☐ Appuyer

☒ Supplier

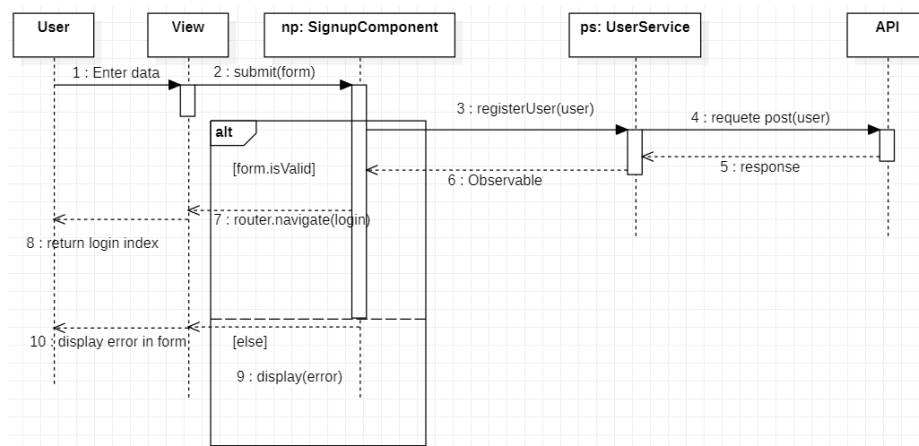
Password

Sign up

Log in

1. L'utilisateur encode ses informations
2. Il clique sur « Sign up »
3. Si les informations sont valides il est redirigé vers login

5.6.2 DIAGRAMME DE SÉQUENCE



5.6.3 SÉCURITÉ ET GESTION D'ERREUR

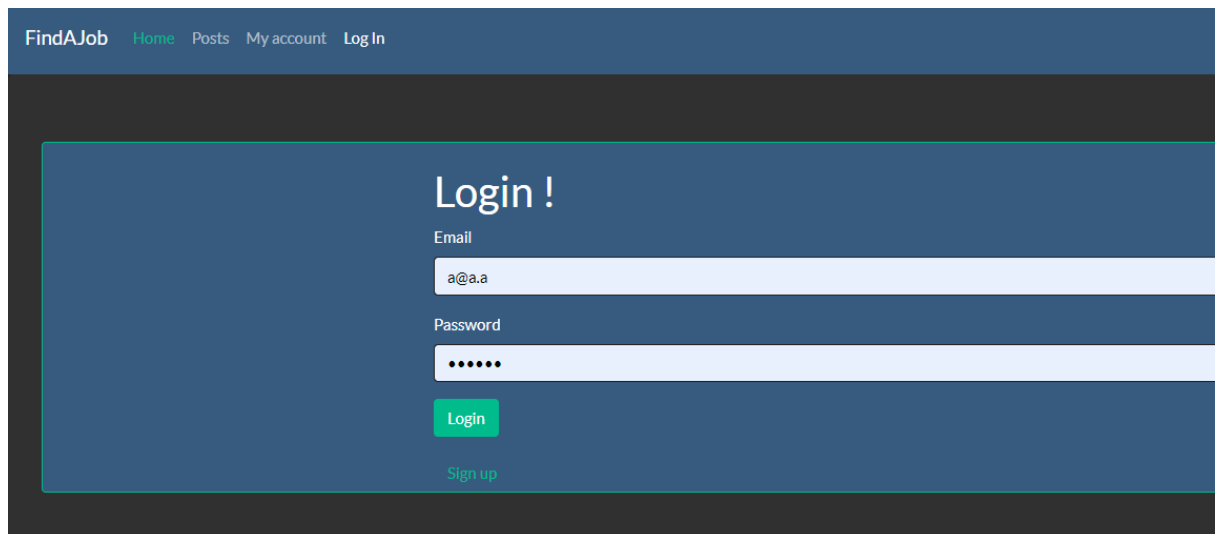
- Le mot de passe est flouté et doit faire au moins 6 caractères
- Tous les champs doivent être remplis
- L'email doit être un email valide

5.6.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/security/sign-up/sign-up.component.ts
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/security/sign-up/sign-up.component.html
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/users/user.service.ts

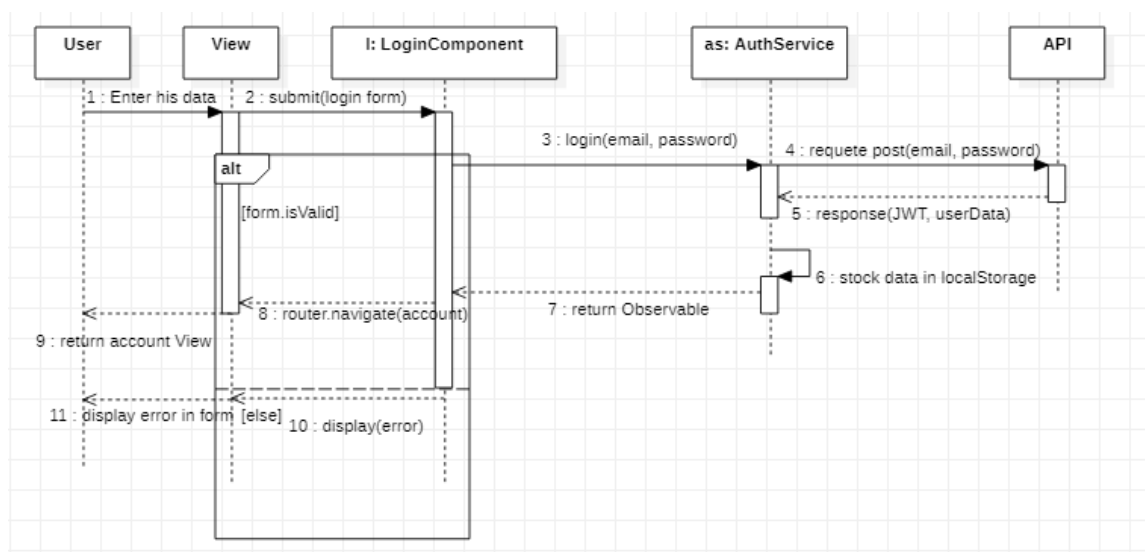
5.7 ME CONNECTER SUR LE SITE

5.7.1 EXPLICATION DE LA FONCTIONNALITÉ



- 1 J'entre mon identifiant et mot de passe
- 2 Je clique sur login (ou « Sign up » si je n'ai pas de compte)
- 3 Je suis redirigé vers mon compte

5.7.2 DIAGRAMME DE SÉQUENCE



5.7.3 SÉCURITÉ ET GESTION D'ERREUR

- On doit avoir un compte pour se connecter sinon on doit s'inscrire
- On est redirigé vers cette page quand on tente d'accéder à des parties du site ou il faut être connecté

5.7.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/security/auth.service.ts
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/security/login/login.component.html
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/security/login/login.component.ts

5.8 MODIFIER MES INFORMATIONS

5.8.1 EXPLICATION DE LA FONCTIONNALITÉ

The screenshot shows a web form with a dark blue header and a light blue body. The title is "let's edit your account information !". The form contains several input fields and a "Save" button. The fields are labeled as follows:

- First name: Delvaux Pierre
- Second name: Pierre
- Age: 1
- Email: a@a.a
- Phone number: 0473561825
- Address: a@a.a
- Account type: ☒ Applyer, ☐ Supplier
- Password: *****

At the bottom, there is a green "Save" button and a blue "Cancel" link.

- 1 L'utilisateur encode ses informations
- 2 Il clique sur Save
- 3 Si les informations sont valides il est redirigé vers login

5.8.2 DIAGRAMME DE SÉQUENCE

Même diagramme de séquence que pour s'inscrire sur le site excepté que la méthode de la requête vers l'api est patch au lieu de post. En effet le même formulaire est utilisé dans lequel on réinjecte les données de l'utilisateur en cours.

5.8.3 SÉCURITÉ ET GESTION D'ERREUR

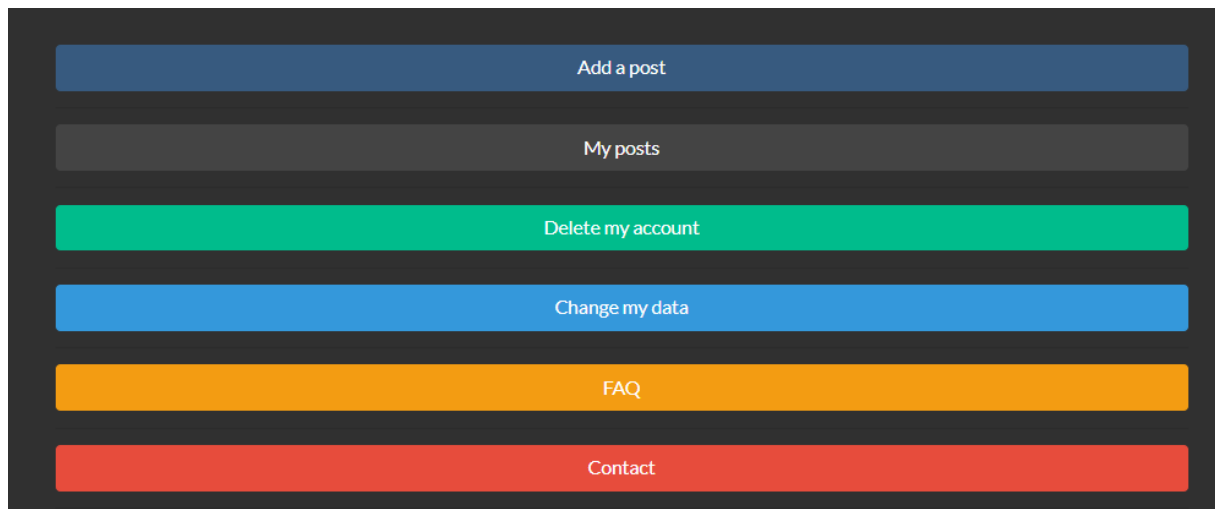
- Le mot de passe est flouté et doit faire au moins 6 caractères
- Tous les champs doivent être remplis
- L'email doit être un email valide

5.8.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/security/sign-up/sign-up.component.ts
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/security/sign-up/sign-up.component.html
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/users/user.service.ts

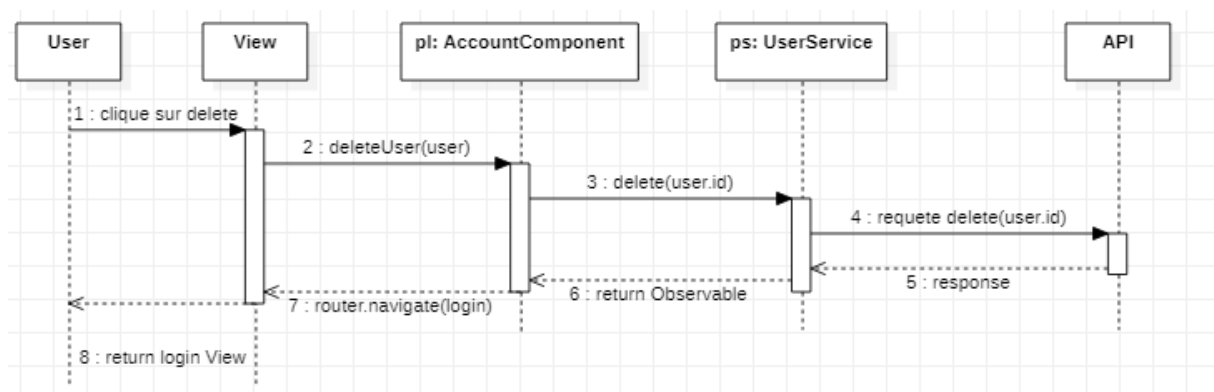
5.9 SUPPRIMER MON COMPTE

5.9.1 EXPLICATION DE LA FONCTIONNALITÉ



- 1 L'utilisateur clique sur « Delete my account »
- 2 Le compte est supprimé de la base de données
- 3 L'utilisateurs est dirigé vers login

5.9.2 DIAGRAMME DE SÉQUENCE



5.9.3 SÉCURITÉ ET GESTION D'ERREUR

- On ne peut évidemment que supprimer son propre compte

5.9.4 LIENS VERS GIT HUB ET/OU PARTIE DU CODE SIGNIFICATIVE

- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/account/account.component.html
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/account/account.component.ts
- https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/users/user.service.ts

6 CONCLUSION

6.1 DIFFICULTÉS RENCONTRÉS

Lors de la réalisation de ce projet j'ai eu énormément de difficulté à de nombreux niveau mais j'ai su surmonter la plus part.

Tout citer serait impossible surtout que ce rapport est déjà bien long. Toutefois je vais me concentrer sur deux problèmes que j'ai eu avec angular.

Premièrement, pour mettre en place la connexion avec le JWT token, j'ai dû comprendre comment modifier le format de donnée renvoyé par Symfony mais aussi tout les paramètres de sécurité offert par le framework. J'ai dû notamment créer la clé privée et publique. Comme la sécurité n'est pas la même pour le site en Symfony et celui en angular, cela s'est avéré compliqué, tant les bugs se sont cumulés pendant plusieurs jours.

Deuxièmement dans angular on ne sait pas facilement mettre à jour les checkboxes dans les formulaires réactifs quand la base de données évolue contrairement à Symfony. J'ai dû trouver quelques petites astuces.

(code => https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/new-post/new-post.component.ts)

Cela m'a permis de comprendre en profondeur les formulaires réactifs.

6.2 OBJECTIFS ATTEINTS/NON-ATTEINTS

Symfony				Angular	
User stories	Priorité	Réalisation	Reste à faire	Réalisation	Reste à faire
En tant qu'utilisateur, je veux pouvoir créer une offre ou demande d'emploi via un formulaire.	1	100		100	
En tant qu'utilisateur, je veux pouvoir lister les différentes propositions par catégorie et offre ou demande.	1	100		100	
En tant qu'utilisateur, je veux pouvoir modifier la fiche que j'ai posté.	1	100		100	
En tant qu'utilisateur, je veux pouvoir supprimer sa fiche.	1	100		100	
En tant qu'utilisateur, je veux pouvoir rechercher une fiche.	1	99	(Bug sera résolu d'ici l'oral)	100	
En tant qu'utilisateur, je veux pouvoir me connecter sur le site.	1	80	La connexion marche bien mais on peut accéder à certaine url en	70	Problème de connexion des nouveaux utilisateurs

			tapant en path a la main		Et on doit recharger la page pour être complètement connecté
En tant qu'utilisateur, je veux pouvoir m'inscrire sur le site via un formulaire.	2	100		90	Le mot de passe n'est pas crypté dans angular (surement la source du bug de connection)
En tant qu'utilisateur, je veux pouvoir modifier les informations de mon compte.	2	100		100	
En tant qu'utilisateur, je veux pouvoir supprimer mon compte.	2	90	Manque un message de confirmation pour cette opération importante	90	Manque un message de confirmation pour cette opération importante

6.3 PISTE D'AMÉLIORATION

Je ferai mon possible pour améliorer le site d'ici sa présentation mais voici déjà des pistes d'amélioration en plus de ce qui reste à faire :

- Faire des alertes en cas de succès et des confirmations avant d'effectuer une opération
- Améliorer la vérification des données
- Fixer les problèmes de login
- Créer un type de compte admin qui puisse ajouter des catégories (déjà possible dans le menu compte via des routes non faites pour ça)
- La précision des diagrammes de séquences
- Etc...

7 BIBLIOGRAPHIE

<https://api-platform.com/>

<https://angular.io/>

<https://symfony.com/doc/current/index.html#gsc.tab=0>