Table des matières

[Rapport du Projet de site web 2](#_Toc29157169)

[1 Spécifications 2](#_Toc29157170)

[1.1 Objectif du site 2](#_Toc29157171)

[1.2 Acteurs 2](#_Toc29157172)

[1.3 User stories 2](#_Toc29157173)

[1.4 Contraintes non fonctionnelles : 2](#_Toc29157174)

[1.5 Mockup 3](#_Toc29157175)

[2 Base de données 5](#_Toc29157176)

[2.1 Diagramme entité relation 5](#_Toc29157177)

[2.2 Schéma relationnel 5](#_Toc29157178)

[2.3 Description des tables et des relations entre tables 5](#_Toc29157179)

[2.4 Description précise des champs 5](#_Toc29157180)

[3 Symfony 7](#_Toc29157181)

[3.1 Créer un post 7](#_Toc29157182)

[3.2 Lister et trier les posts par catégories, demande et offre d’emploi 8](#_Toc29157183)

[3.3 Modifier mes posts 9](#_Toc29157184)

[3.4 Supprimer mon post 10](#_Toc29157185)

[3.5 Rechercher un post par son id 11](#_Toc29157186)

[3.6 M’inscrire sur le site 12](#_Toc29157187)

[3.7 Me connecter sur le site 13](#_Toc29157188)

[3.8 Modifier mes informations 14](#_Toc29157189)

[3.9 Supprimer mon compte 15](#_Toc29157190)

[4 API Rest 17](#_Toc29157191)

[4.1 Tableau récapitulatif 17](#_Toc29157192)

[4.2 Détails des requêtes 17](#_Toc29157193)

[5 Angular 21](#_Toc29157194)

[5.1 Créer un post 21](#_Toc29157195)

[5.2 Lister et trier les posts par catégories, demande et offre d’emploi 22](#_Toc29157196)

[5.3 Modifier mes posts 23](#_Toc29157197)

[5.4 Supprimer mon post 24](#_Toc29157198)

[5.5 Rechercher un post par son id 25](#_Toc29157199)

[5.6 M’inscrire sur le site 26](#_Toc29157200)

[5.7 Me connecter sur le site 27](#_Toc29157201)

[5.8 Modifier mes informations 28](#_Toc29157202)

[5.9 Supprimer mon compte 29](#_Toc29157203)

[6 Conclusion 31](#_Toc29157204)

[6.1 Difficultés rencontrés 31](#_Toc29157205)

[6.2 Objectifs atteints/non-atteints 31](#_Toc29157206)

[6.3 Piste d’amélioration 32](#_Toc29157207)

[7 Bibliographie 33](#_Toc29157208)

# Rapport du Projet de site web

## 1 Spécifications

### 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.

### 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

### 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 |

### Contraintes non fonctionnelles :

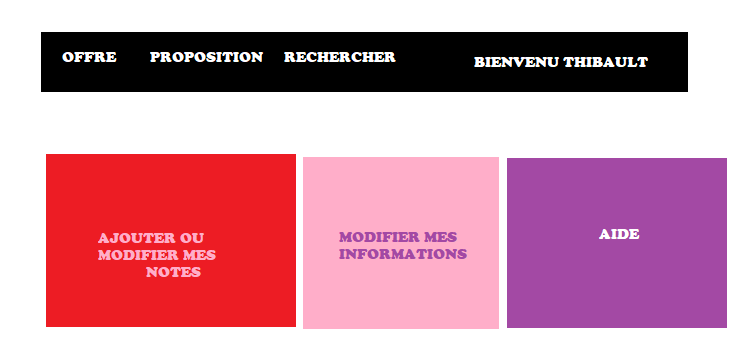
Le site a le devoir d’être responsif et ergonomique.

### Mockup

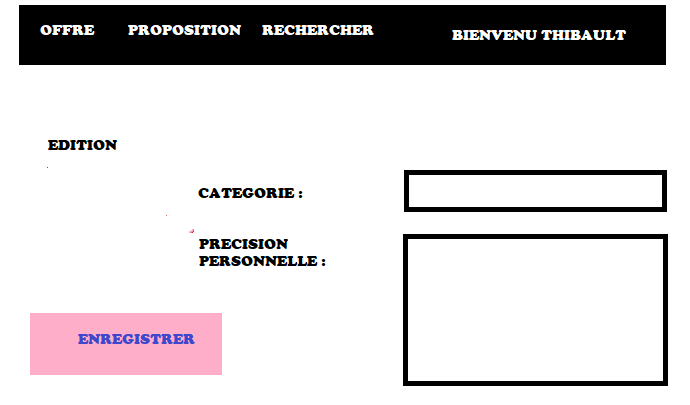
#### liste de demande d’emploi



#### Page compte

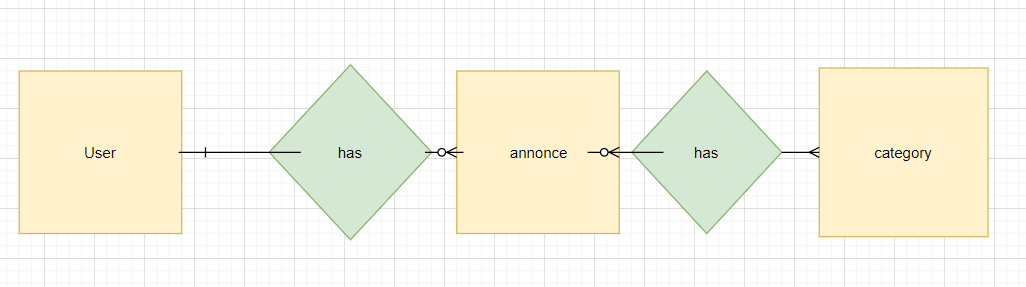


#### Edition dune note

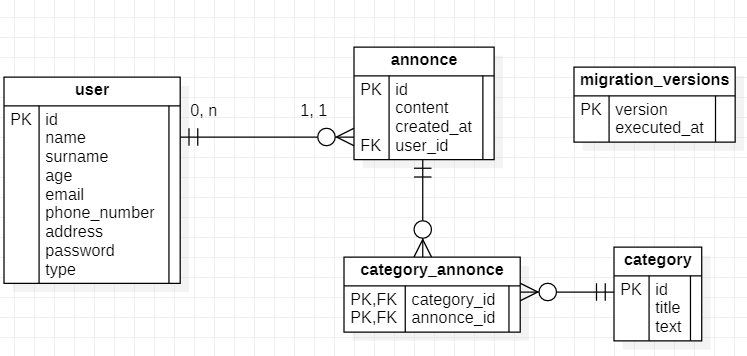


## Base de données

### Diagramme entité relation



### Schéma relationnel



### 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. |

### Description précise des champs

#### 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) |  | Addresse |  |
| password |  | O | varchar(255) |  | Mot de passe | Crypter grâce à bcrypt |
| type |  | O | varchar(255) |  | Type de compte | Employeur ou chercheur d’emploi |

#### 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 |  |

#### 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 |

#### Table : category\_annonce

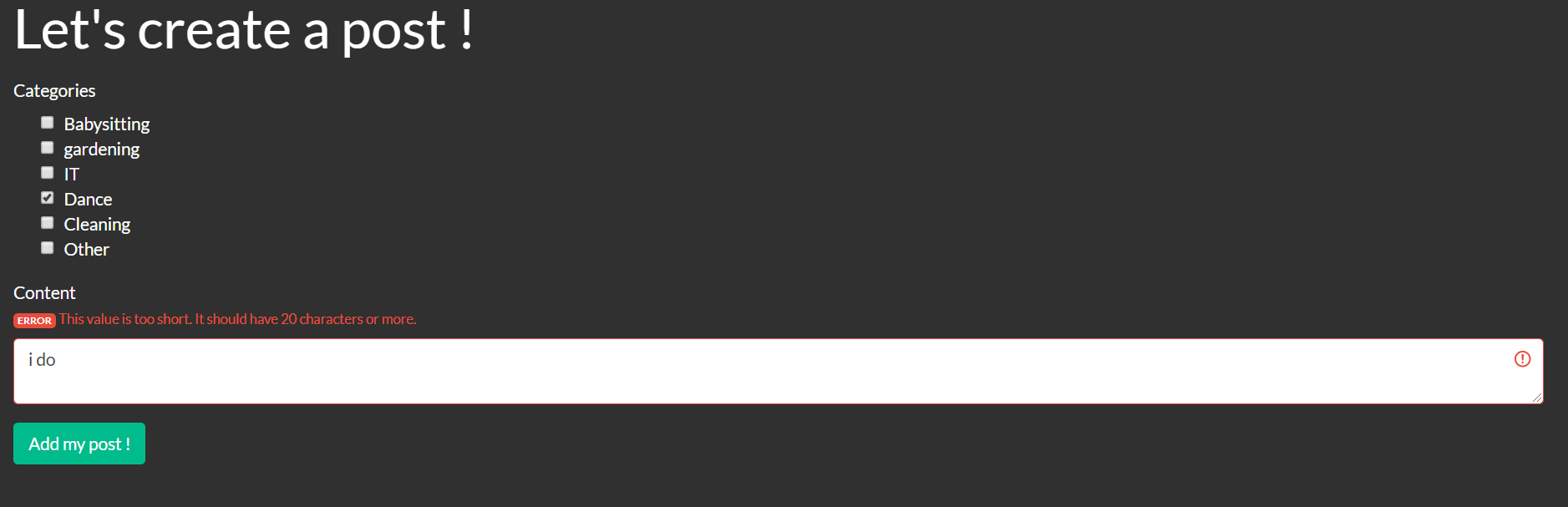
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nom** | **Key** | **Non null** | **Type** | **Référence** | **Intitulé** | **Commentaire** |
| category\_id | PFK | O | int | id/category | Catégorie | Les deux peuvent servir de clé primaire |
| annonce\_id | PFK | O | int | id/annonce | Annonce |  |

## Symfony

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

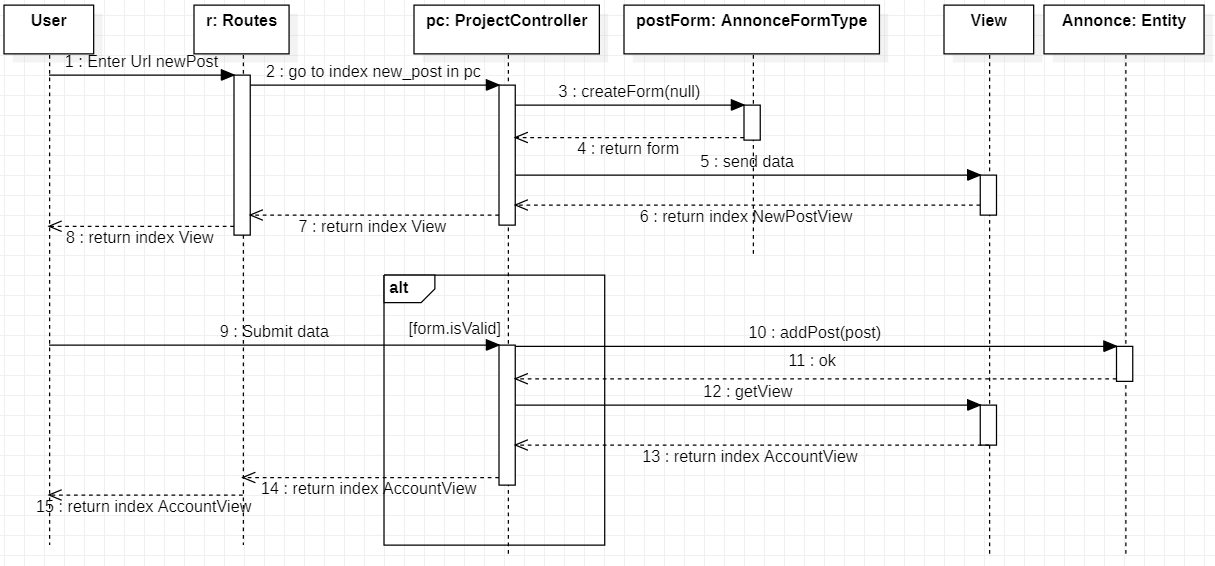
### Créer un post

#### explication de la fonctionnalité



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

#### Diagramme de séquence



#### Sécurité et gestion d’erreur

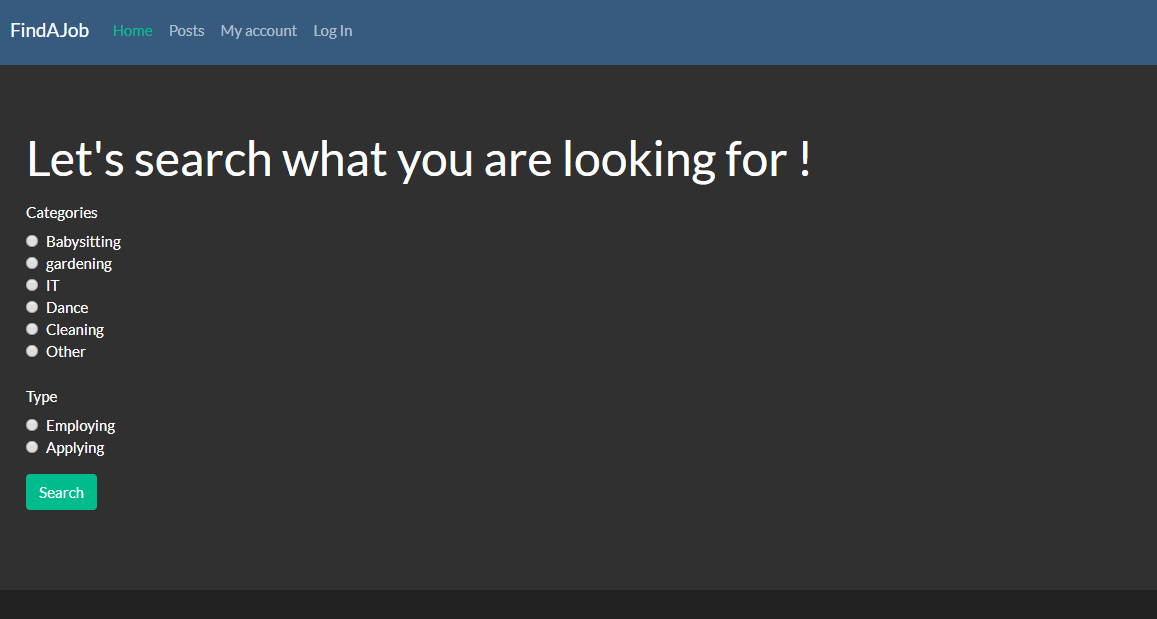
* On doit cocher au moins une catégorie
* Le contenu doit contenir au moins 20 caractères
* Il faut être connecté

#### 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>

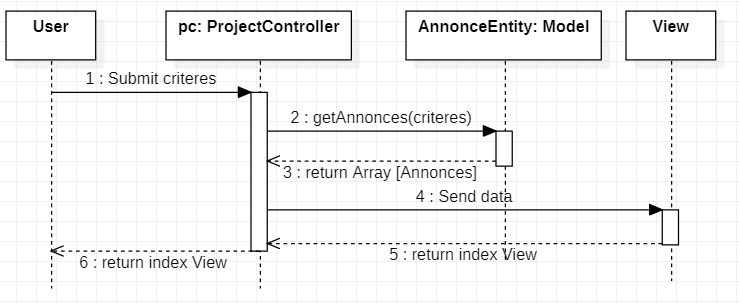
### Lister et trier les posts par catégories, demande et offre d’emploi

#### 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

#### Diagramme de séquence



#### Sécurité et gestion d’erreur

* On ne pourra cliquer sur les détails d’une annonce qu’une fois connecter

#### 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>

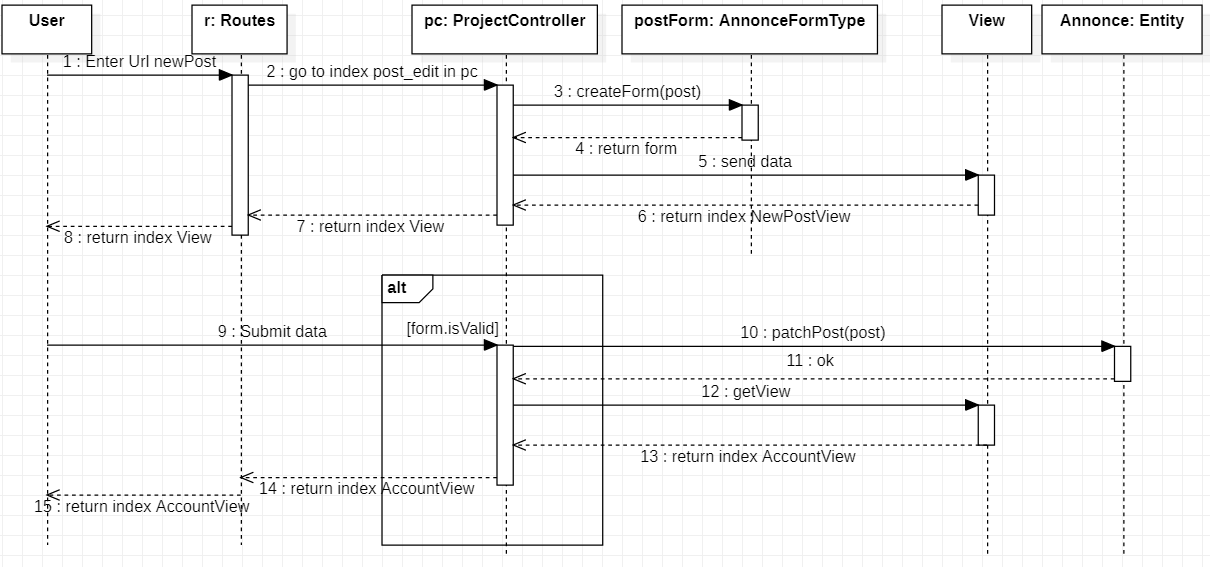
### Modifier mes posts

#### explication de la fonctionnalité



1. L’utilisateur clique sur edit en bas d’une de ses annonces
2. Le formulaires pré-encodé apparait
3. Il modifie ce qu’il doit modifier
4. Il clique sur « Save »
5. Le formulaire disparait et il est redirigé vers

#### Diagramme de séquence



#### Sécurité et gestion d’erreur

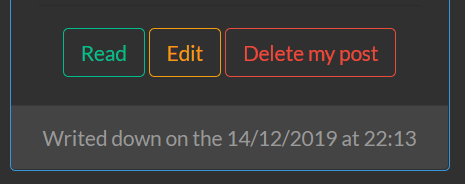
* On doit cocher au moins une catégorie
* Le contenu doit contenir au moins 20 caractères
* On ne peut modifier que ses annonces

#### 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>

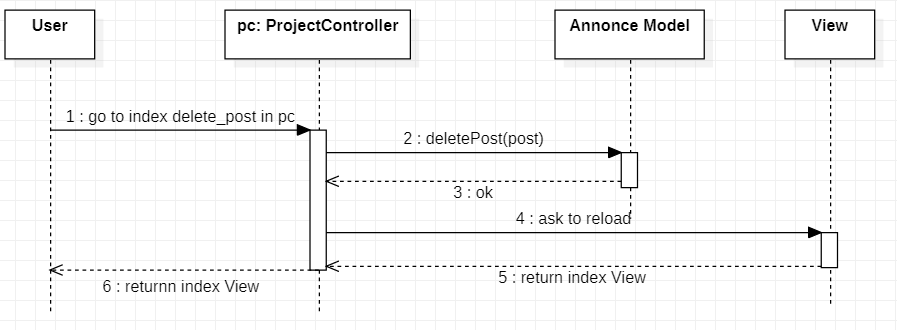
### Supprimer mon post

#### 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 disparait de la vue

#### Diagramme de séquence



#### Sécurité et gestion d’erreur

* On ne peut supprimer que ses annonces

#### 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>

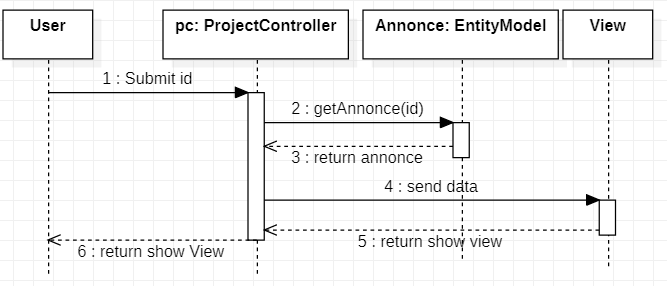
### Rechercher un post par son id

#### 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)

#### Diagramme de séquence



#### Sécurité et gestion d’erreur

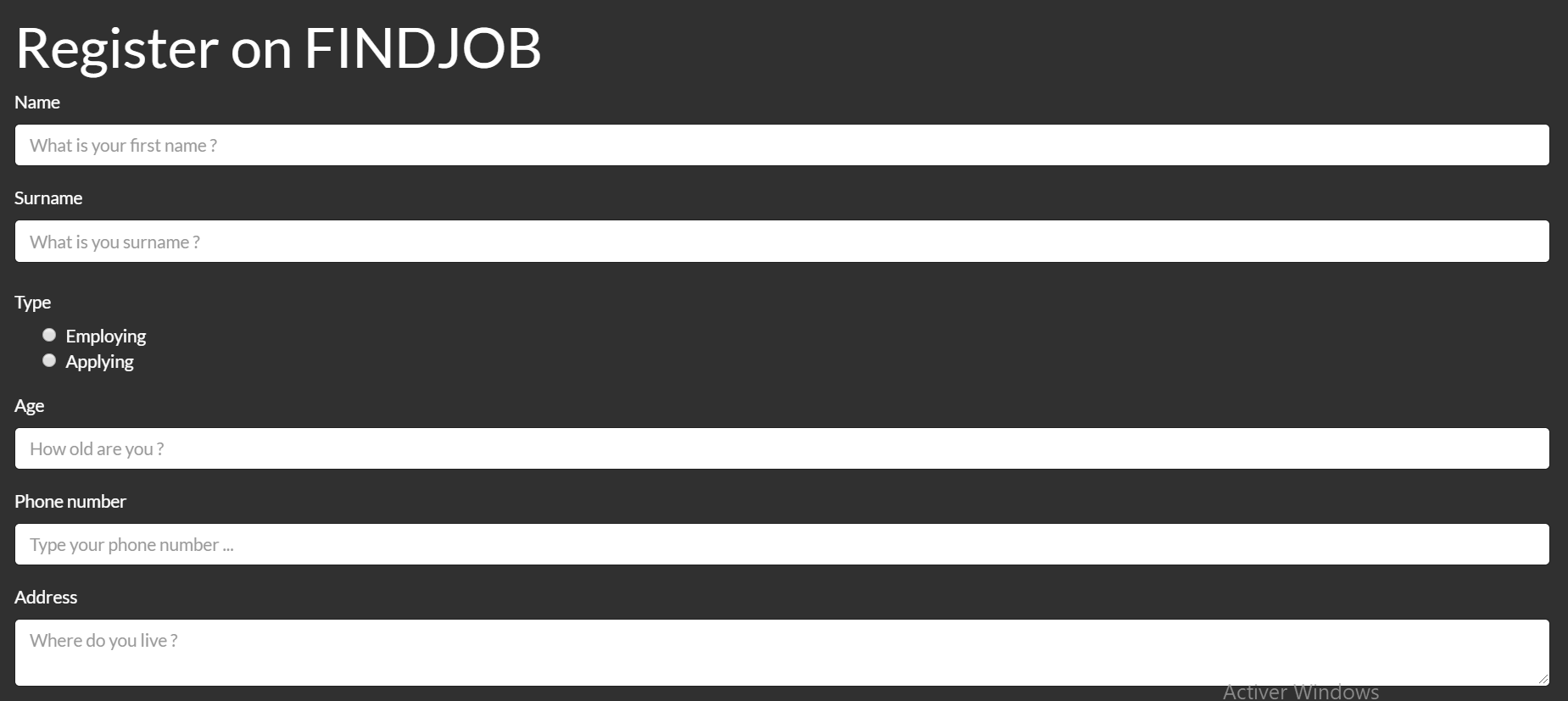
* On ne peut entrer qu’un int
* Il faut être connecté

#### 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>

### M’inscrire sur le site

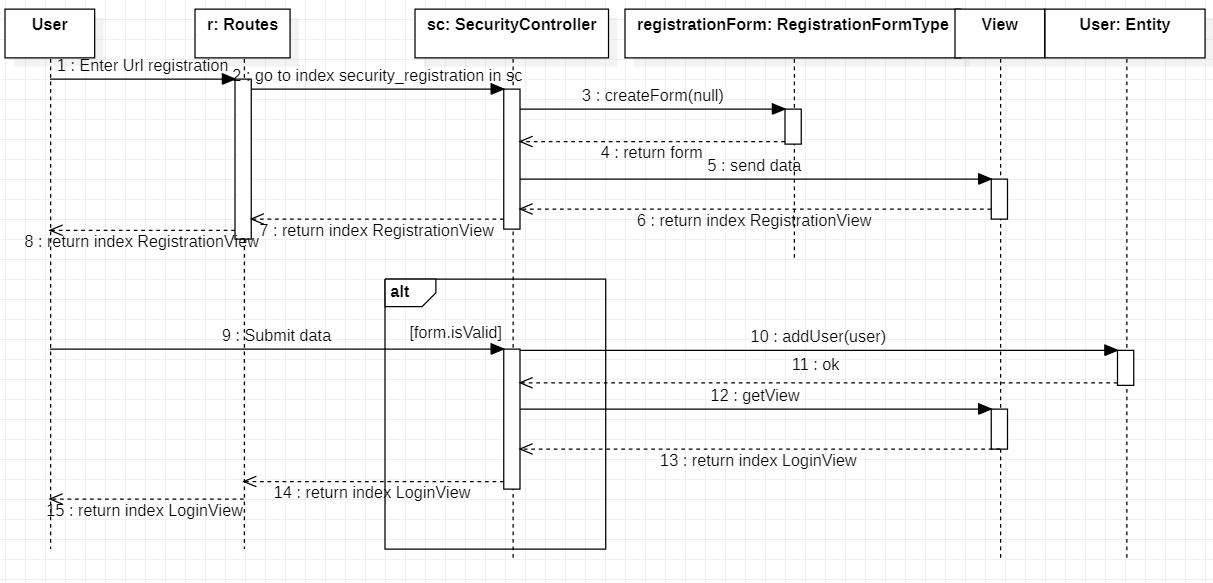
#### explication de la fonctionnalité





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

#### Diagramme de séquence



#### 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

#### Liens vers git hub et/ou partie du code significative

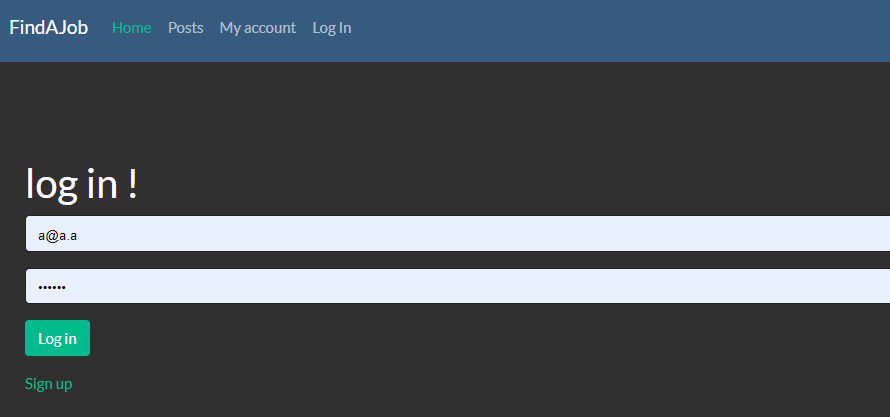
<https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Controller/SecurityController.php>

<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>

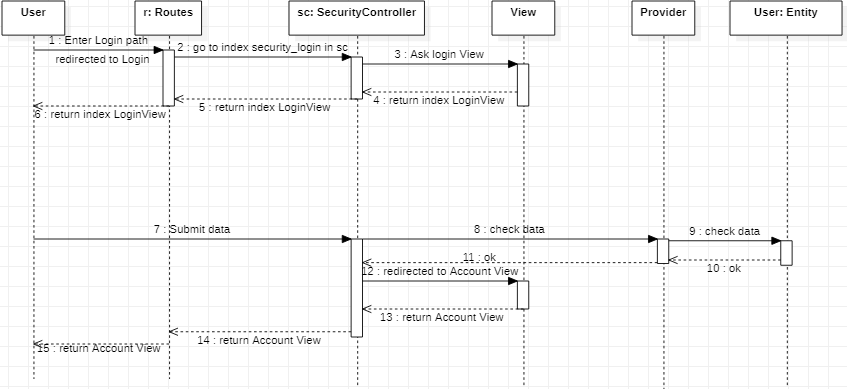
### Me connecter sur le site

#### 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

#### Diagramme de séquence



#### Sécurité et gestion d’erreur

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

#### Liens vers git hub et/ou partie du code significative

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

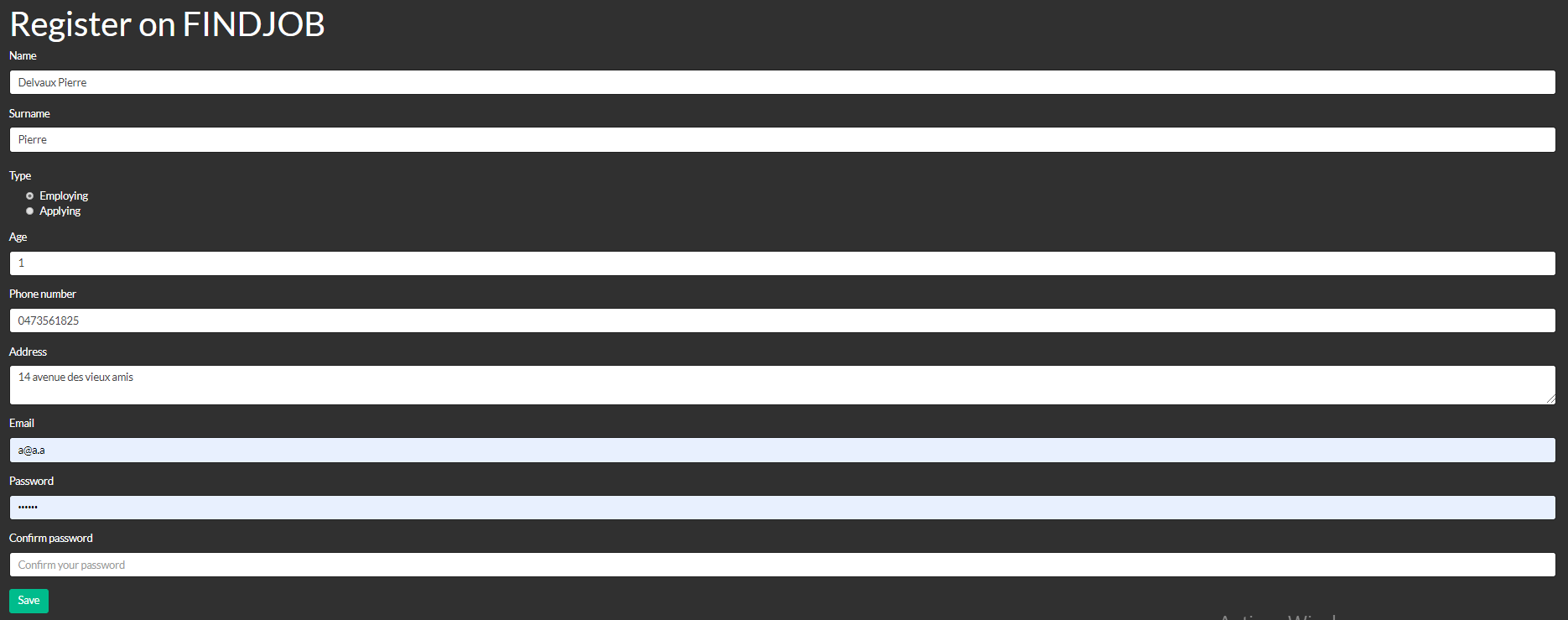
<https://github.com/Horsefactor/Architecture_Web/blob/master/demo/templates/security/login.html.twig>

<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>

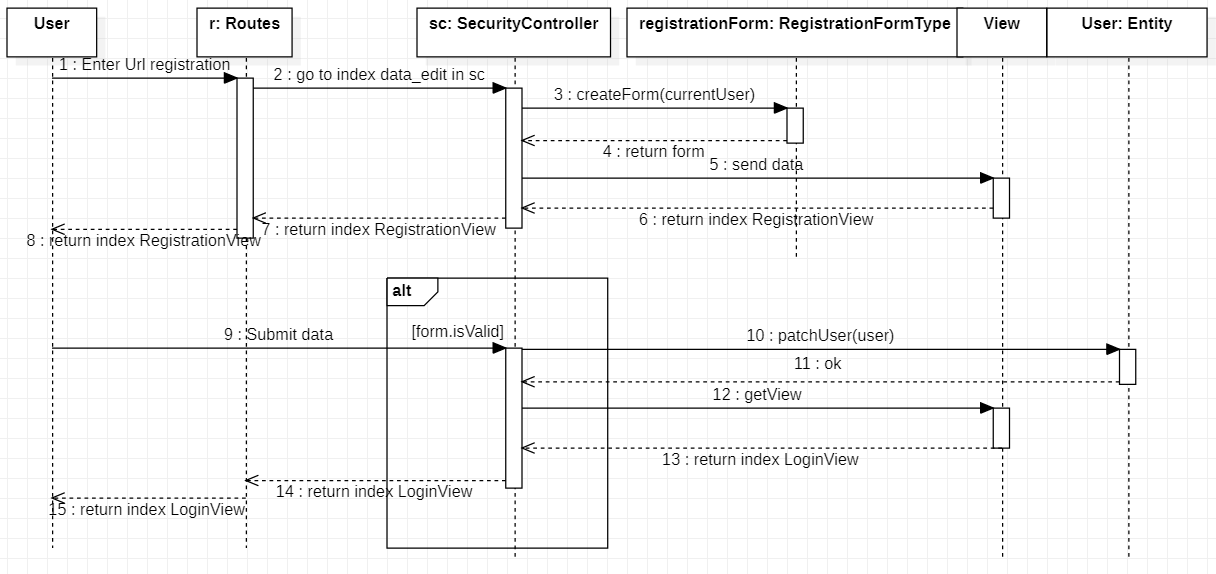
### Modifier mes informations

#### explication de la fonctionnalité



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

#### Diagramme de séquence



#### 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

#### Liens vers git hub et/ou partie du code significative

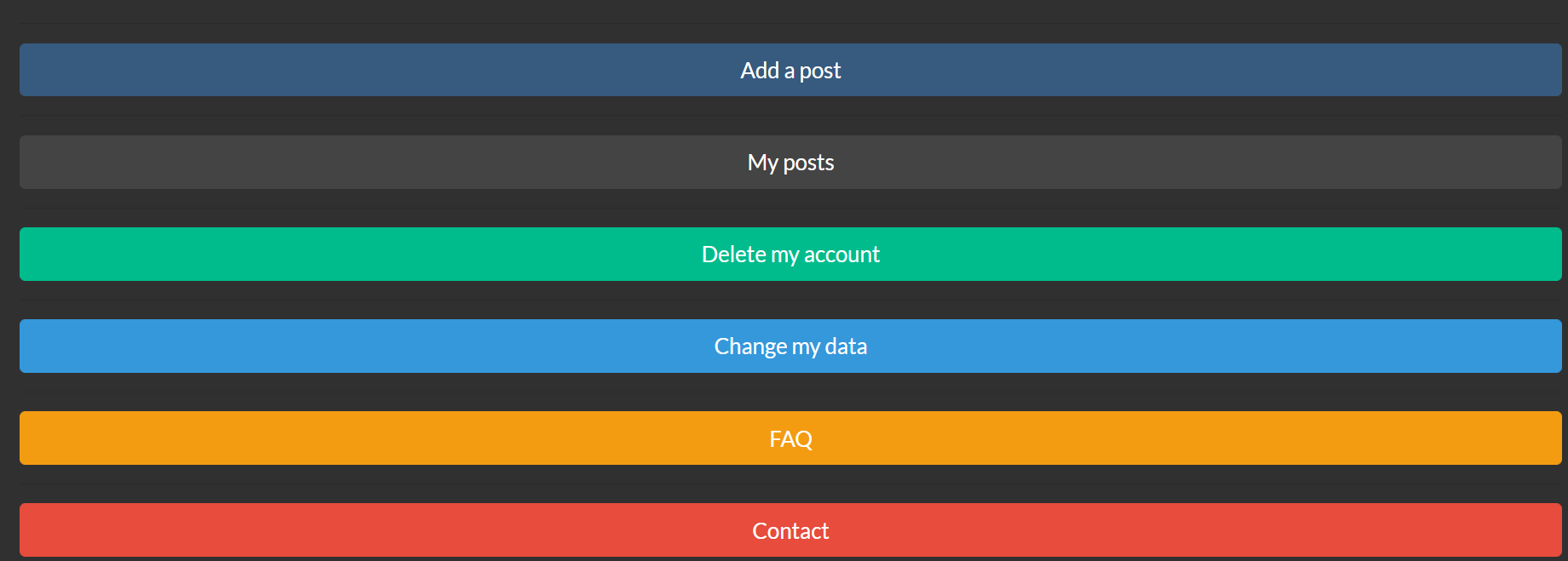
<https://github.com/Horsefactor/Architecture_Web/blob/master/demo/src/Controller/SecurityController.php>

<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>

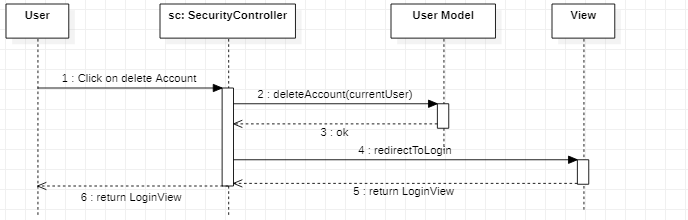
### Supprimer mon compte

#### 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

#### Diagramme de séquence



#### Sécurité et gestion d’erreur

* On ne peut évidemment que supprimer son propre compte

#### 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>

## 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.

### 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 | [..](http://localhost:8000/api)/annonces | Donne toutes les annonces |
| DELETE | [..](http://localhost:8000/api)/annonces/{id} | Supprime une annonce grâce à son id |
| PATCH | [..](http://localhost:8000/api)/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 | [..](http://localhost:8000/api)/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 corresponde à l’utilisateur (sauf l’adresse et le mot de passe par sécurité) |

### 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",

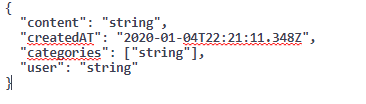
"hydra:member": [ **\*ici se situe la liste des objets renvoyés dont la structure est expliqué plus bas\***]

"hydra:totalItems": 6

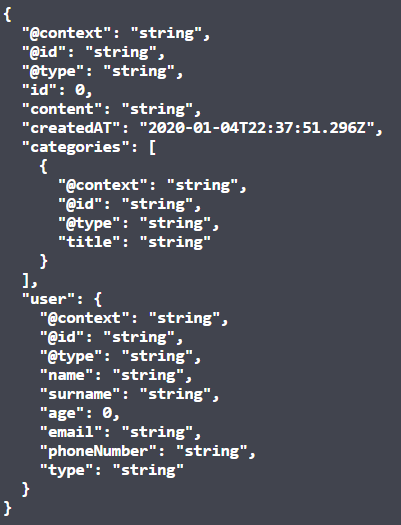
}

#### Requêtes d’annonces

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



La réponse d’une requête GET ressemblera à ça :

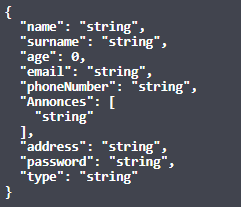


Cette fois-ci j’ai choisi de personnalisé la réponse pour ne pas devoir faire pleins de requêtes lors de l’affichage d’une annonce.

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

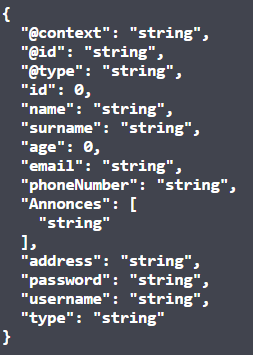
<https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/posts/post.service.ts>

#### Requêtes d’users

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

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

La réponse d’une requête GET ressemblera à ça :

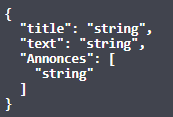


Le service qui gère les requêtes d’user dans angular 8 :

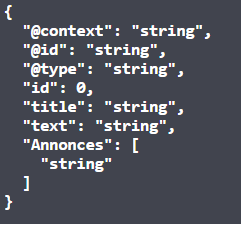
<https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/users/user.service.ts>

#### Requêtes de catégories

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

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

La réponse d’une requête GET ressemblera à ça :



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>

#### 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 :

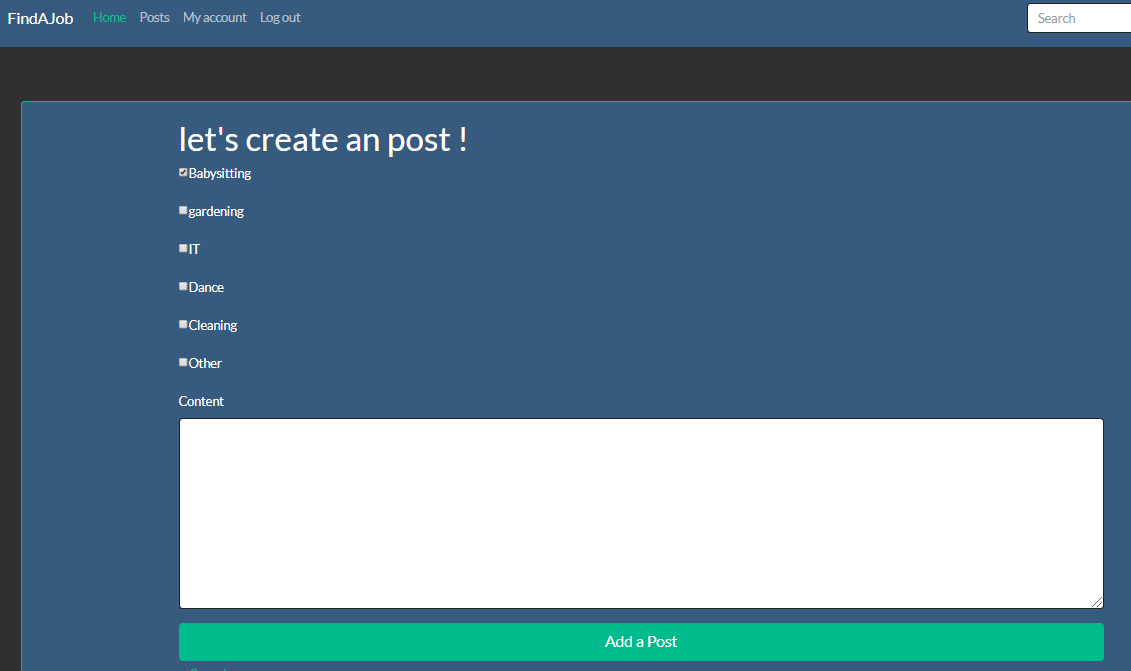
* <https://github.com/Horsefactor/Architecture_Web/tree/master/demo/config/jwt>

## Angular

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

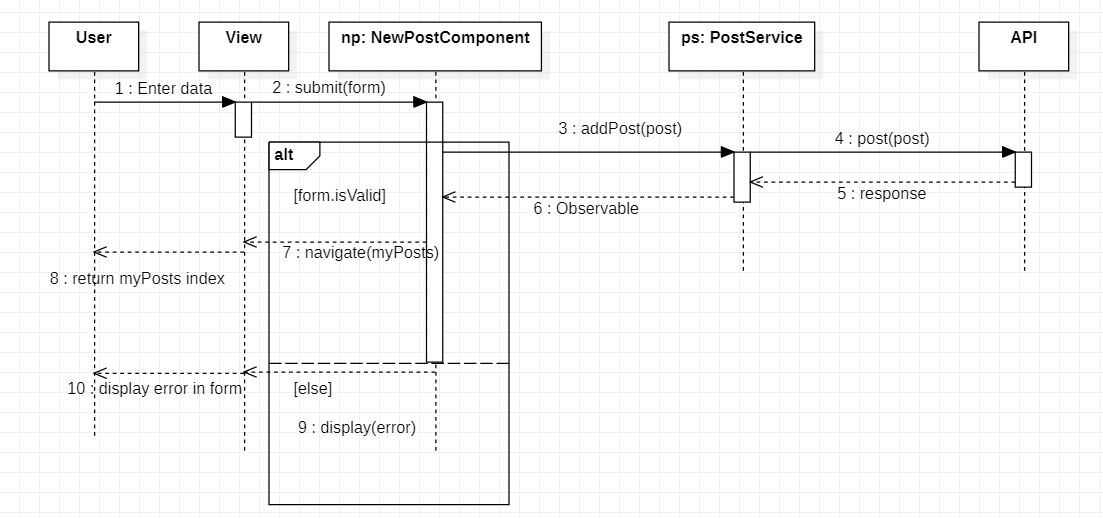
### Créer un post

#### explication de la fonctionnalité



1. L’utilisateur choisit ajouter un post dans le menu « My account ».
2. Le formulaire apparait.
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.

#### Diagramme de séquence



#### Sécurité et gestion d’erreur

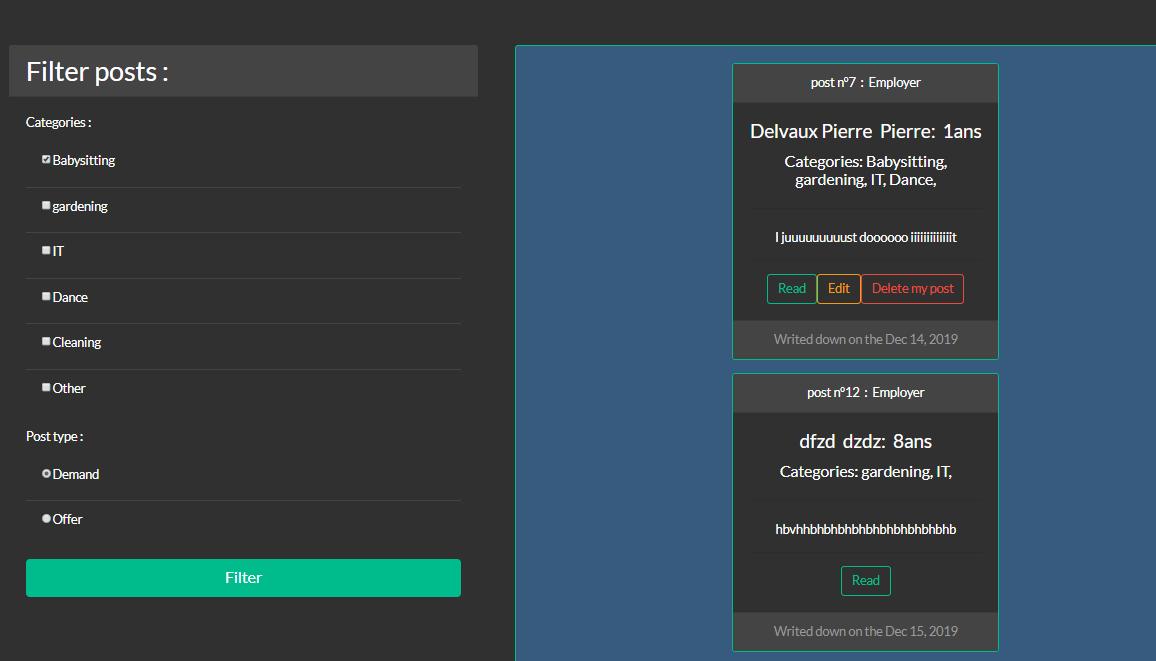
* Accessible que si on est connecté sur un compte
* On doit cocher au moins une catégorie
* Le contenu doit contenir au moins 20 caractères

#### 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>

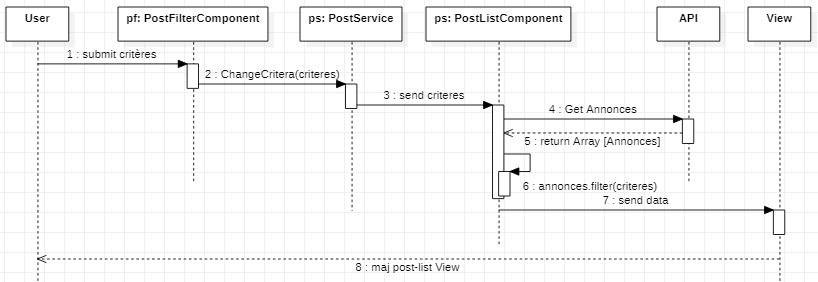
### Lister et trier les posts par catégories, demande et offre d’emploi

#### 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

#### Diagramme de séquence



#### Sécurité et gestion d’erreur

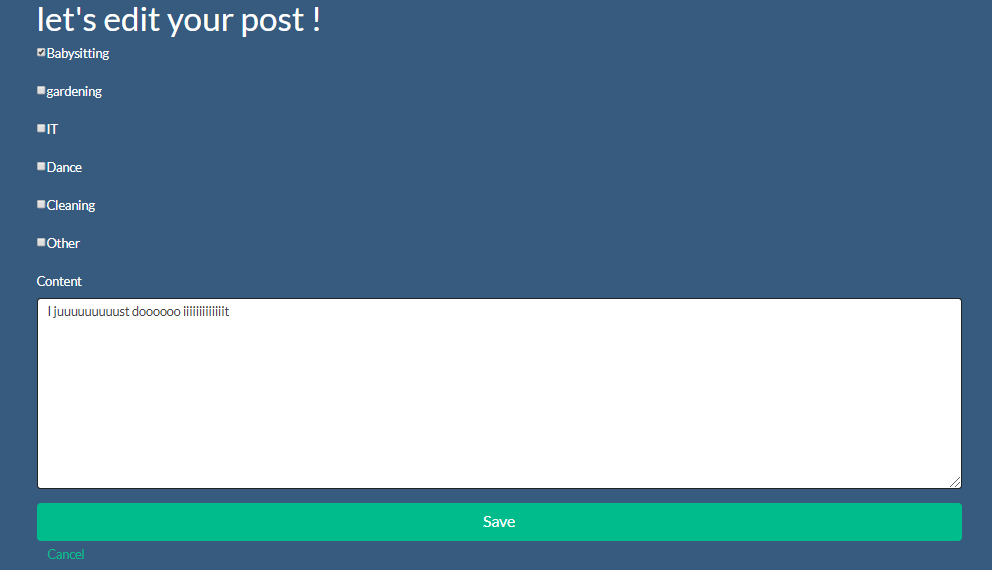
* On ne peut pas cliquer sur « Read » si on n’est pas connecté sur un compte car on afficherait les données pour joindre l’auteur de l’annonce
* On ne peut éditer et supprimer que ses annonces
* Par défaut on affiche toutes les annonces

#### 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>

### Modifier mes posts

#### explication de la fonctionnalité



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

#### 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.

#### Sécurité et gestion d’erreur

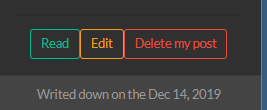
* Accessible que si on est connecté sur un compte et que c’est son post
* On doit cocher au moins une catégorie
* Le contenu doit contenir au moins 20 caractères

#### 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>

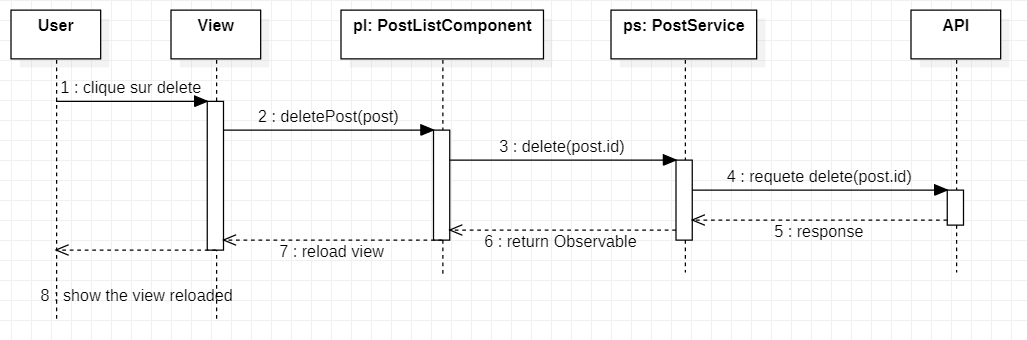
### Supprimer mon post

#### explication de la fonctionnalité



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

#### Diagramme de séquence



#### Sécurité et gestion d’erreur

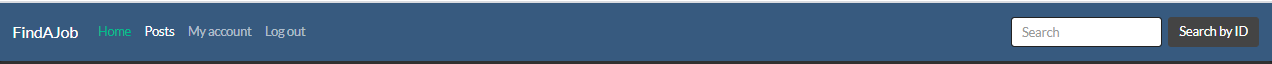
* On ne peut supprimer que ses posts

#### Liens vers git hub et/ou partie du code significative

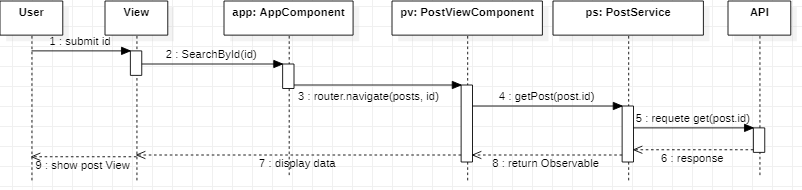
* <https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/post-list/post-list.component.html>
* <https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/shared/posts/post.service.ts>
* <https://github.com/Horsefactor/Architecture_Web/blob/master/project/src/app/posts/post-list/post-list.component.ts>

### Rechercher un post par son id

#### explication de la fonctionnalité



#### Diagramme de séquence



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)

#### Sécurité et gestion d’erreur

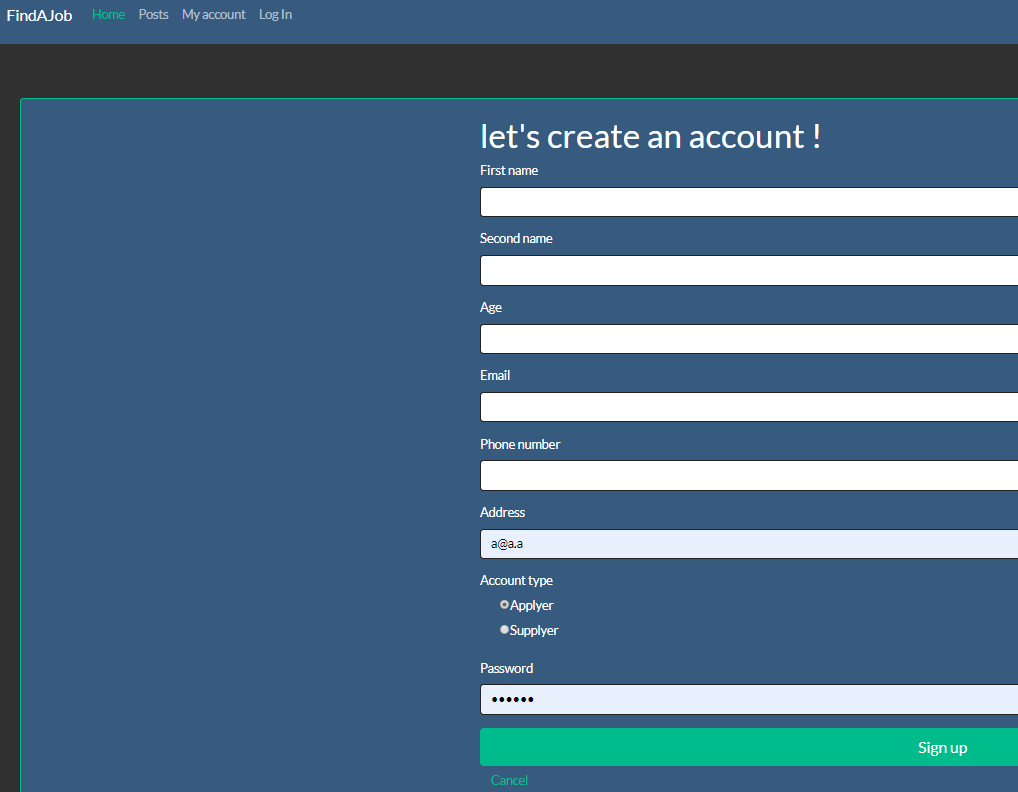
* On doit être connecté, car on affiche la vue d’une annonce avec les coordonnées de l’auteur
* On doit taper un entier

#### 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>

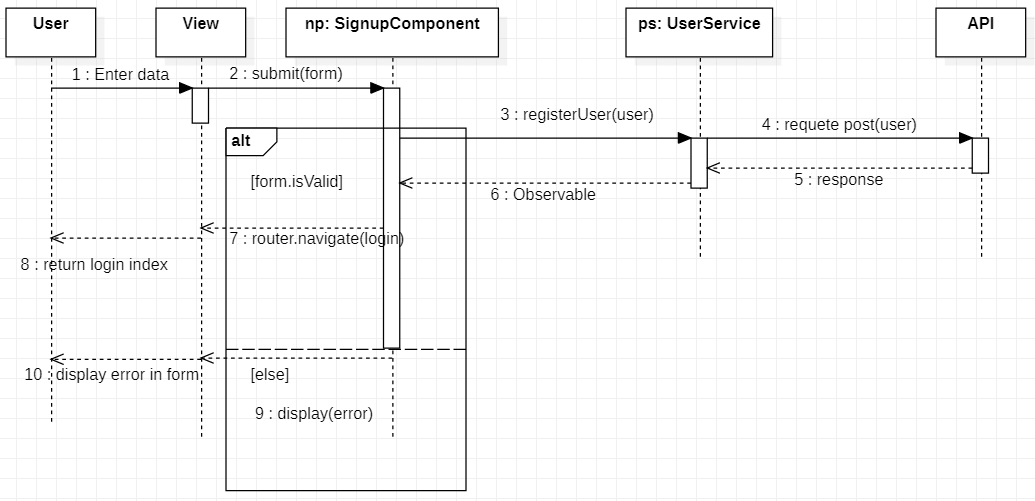
### M’inscrire sur le site

#### explication de la fonctionnalité



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

#### Diagramme de séquence



#### Sécurité et gestion d’erreur

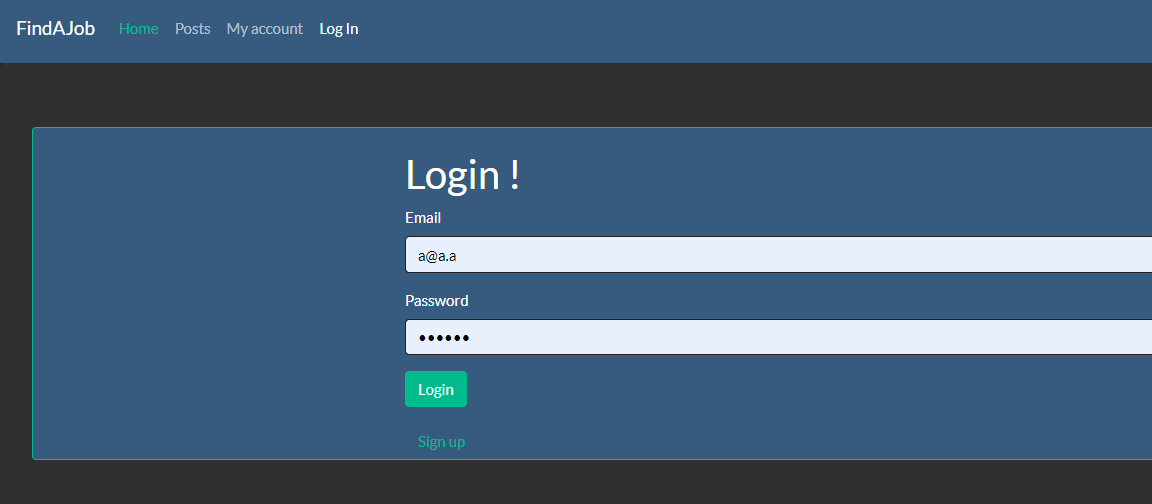
* Le mot de passe est flouté et doit faire au moins 6 caractères
* Tous les champs doivent être remplis
* L’email doit être un email valide

#### 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>

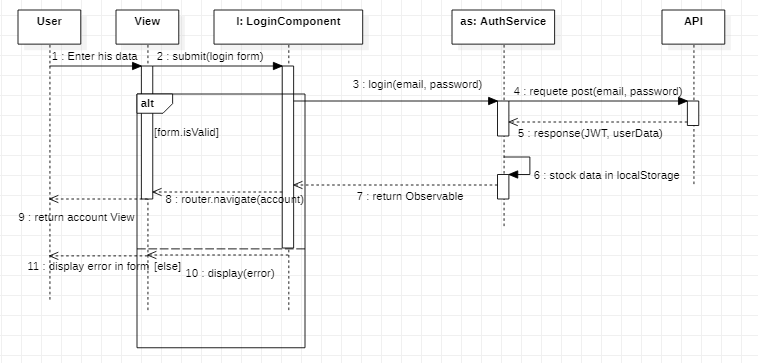
### Me connecter sur le site

#### 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

#### Diagramme de séquence



#### Sécurité et gestion d’erreur

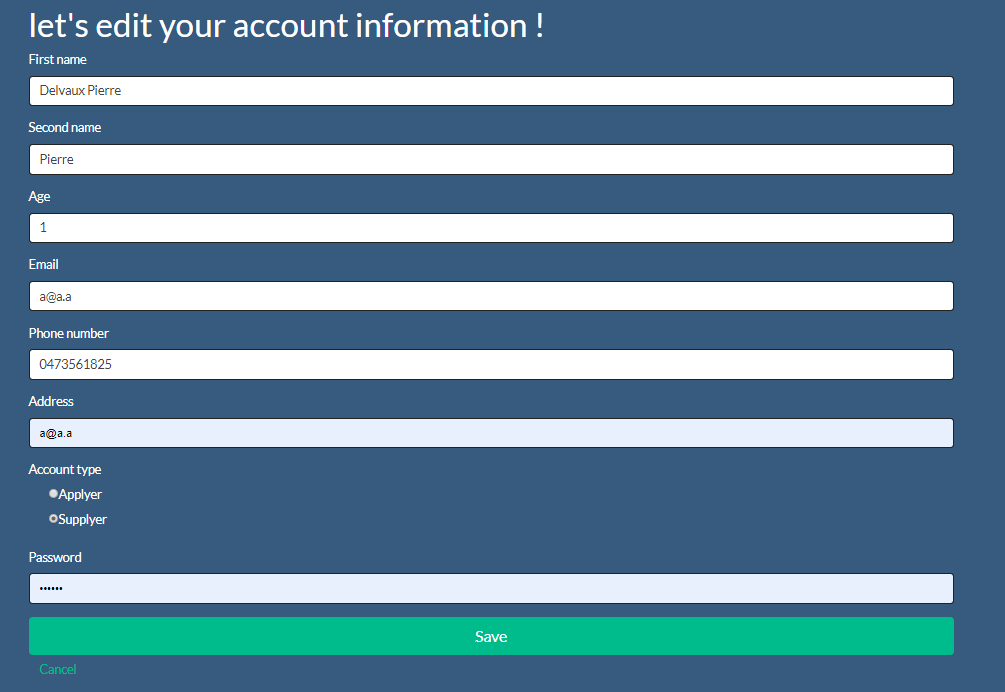
* On doit avoir un compte pour se connecter sinon on doit s’inscrire
* On est redirigé vers cette page quand on tente d’accéder à des parties du site ou il faut être connecté

#### 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>

### Modifier mes informations

#### explication de la fonctionnalité



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

#### 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.

#### Sécurité et gestion d’erreur

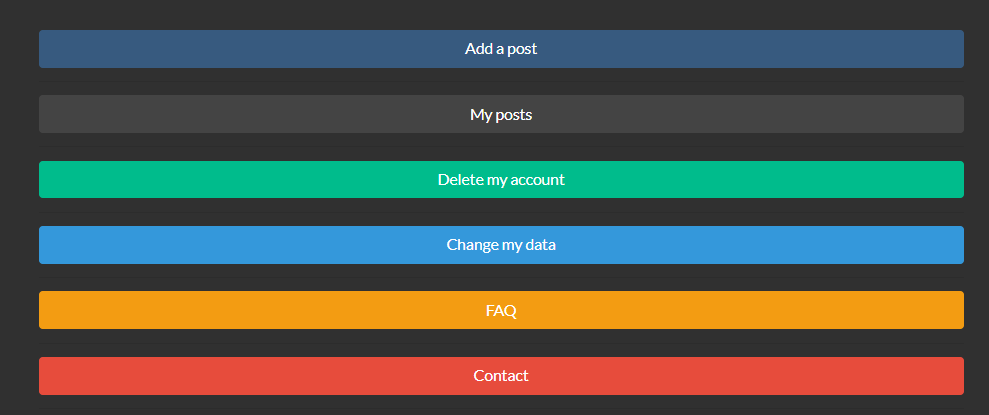
* Le mot de passe est flouté et doit faire au moins 6 caractères
* Tous les champs doivent être remplis
* L’email doit être un email valide

#### 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>

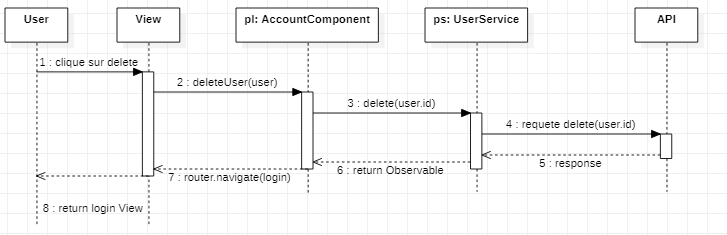
### Supprimer mon compte

#### 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

#### Diagramme de séquence



#### Sécurité et gestion d’erreur

* On ne peut évidemment que supprimer son propre compte

#### 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>

## Conclusion

### 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 serrait 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 checkboxs 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.

### 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 serra 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 tapant en path a la main | 70 | Problème de connexion des nouveaux utilisateurs  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 |

### 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…

## Bibliographie

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

<https://angular.io/>

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