

Web architecture

Décembre 2019

SMITS Victor 16107

Ecole centrale des arts et métiers

MIN1

Année académique 2019-2020

Contents

1	Spécification	5
1	Objectif du site	5
2	Acteur	5
3	User Stories	5
3.1	Compatible Symfony-Twig et Symfony-Angular	5
3.2	Compatible Symfony-Angular uniquement	6
4	Contrainte non fonctionnelles	6
5	Mockup	7
5.1	Login	7
5.2	Accueil	7
5.3	Inscription	8
5.4	Month	9
5.5	Admin - Session	9
5.6	Admin - Nouvelle session	10
5.7	Admin - Abonnement	10
5.8	Admin - Type Session	11
2	Base de donnée	12
1	Diagramme Entité relation	12
2	Diagramme Relationnel	13
3	Description des tables et des relations entre tables	13
4	Description précise des champs	14
4.1	Table : Person	14
4.2	Table : Session	14
4.3	Table : Type_Session	15
4.4	Table : inscription	15
4.5	Table : lien_person_type_session	15
3	MVC	16
1	User Stories compatible Angular - Twig	16
1.1	En tant qu'utilisateur, je dois pouvoir me connecter sur le site	16
1.2	En tant qu'utilisateur, je dois pouvoir me créer un compte sur le site	18

1.3	En tant qu'utilisateur, je dois pouvoir sélectionner le mois et l'année pour afficher les sessions qui m'intéresse	20
1.4	En tant qu'utilisateur, je dois pouvoir m'inscrire à une session un certain jour	20
1.5	En tant qu'utilisateur, je dois pouvoir être inscrit automatiquement à une session	21
1.6	En tant qu'utilisateur, je dois pouvoir me désinscrire à une session	22
1.7	En tant qu'utilisateur, je dois pouvoir voir qui est inscrit pour chaque session	23
1.8	En tant qu'utilisateur, je veux pouvoir voir dynamiquement le nombre de séance qui reste dans mon abonnement	24
1.9	En tant qu'utilisateur, je veux pouvoir modifier mon profile . .	25
1.10	En tant qu'administrateur, je dois pouvoir annuler une session	26
1.11	En tant qu'administrateur, je dois pouvoir créer une nouvelle session peut importe la date	28
1.12	En tant qu'administrateur, je dois pouvoir gérer les abonnement des utilisateurs	30
4	API	32
1	Tableau recapitulatif	32
2	GET : /api	34
2.1	Requête	34
2.2	Réponse	34
3	GET : /api/month	35
3.1	Requête	35
3.2	Réponse	36
4	GET : /api/profile	37
4.1	Requête	37
4.2	Réponse	38
5	GET : /api/admin/abonnement	40
5.1	Requête	40
5.2	Réponse	40
6	GET : /api/TypeSession	43
6.1	Requête	43
6.2	Réponse	43
7	PUT : /api/admin/renewAbo	44
7.1	Requête	44
7.2	Réponse	45
8	PUT : /api/admin/editAbo	45
8.1	Requête	45
8.2	Réponse	46
9	PUT : /api/admin/cancel	47
9.1	Requête	47
9.2	Réponse	48

10	PUT : /api/admin/recreate	48
10.1	Requête	48
10.2	Réponse	49
11	PUT : /api/admin/autocreate	50
11.1	Requête	50
11.2	Réponse	50
12	PUT : /api/editProfile	51
12.1	Requête	51
12.2	Réponse	52
13	PUT : /api/TypeSession	53
13.1	Requête	53
13.2	Réponse	54
14	DELETE : /api/admin/session	55
14.1	Requête	55
14.2	Réponse	55
15	DELETE : /api/Desinscription	56
15.1	Requête	56
15.2	Réponse	56
16	DELETE : /api/TypeSession	57
16.1	Requête	57
16.2	Réponse	57
17	DELETE : /api/admin/user	58
17.1	Requête	58
17.2	Réponse	58
18	POST : /api/login	59
18.1	Requête	59
18.2	Réponse	59
19	POST : /api/admin/session	61
19.1	Requête	61
19.2	Réponse	61
20	POST : /api/Inscription	62
20.1	Requête	62
20.2	Réponse	63
21	POST : /api/TypeSession	63
21.1	Requête	63
21.2	Réponse	64
22	POST : /api/register	65
22.1	Requête	65
22.2	Réponse	66
5	Séparation backend frontend	67
1	User Stories compatible Angular uniquement	68
1.1	En tant qu'administrateur, je dois pouvoir créer un type de session	68

1.2	En tant qu'administrateur, je veux pouvoir modifier un type de session	71
1.3	En tant qu'administrateur, je veux pouvoir supprimer un type de session	73
1.4	En tant qu'administrateur, je veux pouvoir supprimer un utilisateur du système	74
1.5	En tant qu'administrateur, je dois pouvoir générer les sessions pour un nombre d'année	76
2	User Stories compatible Symfony-Angular	79
2.1	En tant qu'utilisateur, je dois pouvoir me connecter sur le site	79
2.2	En tant qu'utilisateur, je dois pouvoir me créer un compte sur le site	81
2.3	En tant qu'utilisateur, je dois pouvoir sélectionner le mois et l'année pour afficher les sessions qui m'intéresse	82
2.4	En tant qu'utilisateur, je dois pouvoir m'inscrire à une session un certain jour	82
2.5	En tant qu'utilisateur, je dois pouvoir être inscrit automatiquement à une session	83
2.6	En tant qu'utilisateur, je dois pouvoir me désinscrire à une session	85
2.7	En tant qu'utilisateur, je dois pouvoir voir qui est inscrit pour chaque session	85
2.8	En tant qu'utilisateur, je veux pouvoir voir dynamiquement le nombre de séance qui reste dans mon abonnement	86
2.9	En tant qu'utilisateur, je veux pouvoir modifier mon profile . .	87
2.10	En tant qu'administrateur, je dois pouvoir annuler une session	88
2.11	En tant qu'administrateur, je dois pouvoir créer une nouvelle session peut importe la date	88
2.12	En tant qu'administrateur, je dois pouvoir gérer les abonnement des utilisateurs	89
6	Conclusion	90
1	Objectif atteints et non atteints	90
1.1	Compatible Symfony-Twig et Symfony-Angular	90
1.2	Compatible Symfony-Angular uniquement	92
2	Piste d'amélioration	92
7	Annexe	94
1	Code source	94

1 | Spécification

1 Objectif du site

Le site doit permettre à des personnes de s'inscrire et se désinscrire de leur groupe à des session de sport.

2 Acteur

- Utilisateur : personne qui utilise le site et s'inscrivant à des session de sport
- Administrateur : personne administrant le site et ayant tout contrôle sur les sessions de sport organisé

3 User Stories

3.1 Compatible Symfony-Twig et Symfony-Angular

User Stories	Priority
En tant qu'utilisateur, je dois pouvoir me créer un compte sur le site	2
En tant qu'utilisateur, je dois pouvoir me connecter sur le site	2
En tant qu'utilisateur, je dois pouvoir sélectionner le mois et l'année pour afficher les sessions qui m'intéresse	2
En tant qu'utilisateur, je dois pouvoir m'inscrire à une session un certain jour	1
En tant qu'utilisateur, je dois pouvoir être inscrit automatiquement à une session	1
En tant qu'utilisateur, je dois pouvoir me désinscrire à une session	1
En tant qu'utilisateur, je dois pouvoir voir qui est inscrit pour chaque session	2

En tant qu'utilisateur, je veux pouvoir voir dynamiquement le nombre de séance qui reste dans mon abonnement	2
En tant qu'utilisateur, je veux pouvoir modifier mon profile	1
En tant qu'utilisateur, je veux pouvoir m'inscrire a une session uniquement si il me reste des abonnements a disposition.	2
En tant qu'administrateur, je dois pouvoir annuler une session	2
En tant qu'administrateur, je dois pouvoir gérer les abonnement des utilisateurs	2

3.2 Compatible Symfony-Angular uniquement

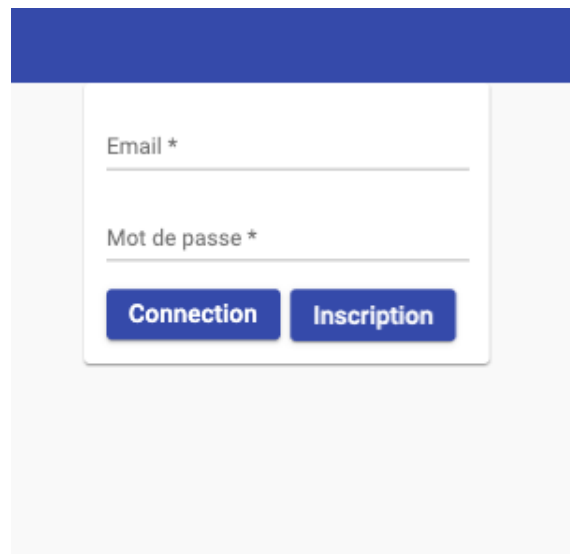
En tant qu'administrateur, je dois pouvoir créer un type de session	2
En tant qu'administrateur, je dois pouvoir générer les sessions pour un nombre d'année	1
En tant qu'administrateur, je veux pouvoir supprimer un utilisateur du système	1
En tant qu'administrateur, je veux pouvoir gérer les types de sessions disponible pour l'inscription automatique	2
En tant qu'administrateur, je veux pouvoir modifier un type de session	1
En tant qu'administrateur, je veux pouvoir supprimer un type de session	1

4 Contrainte non fonctionnelles

- Le site doit être user-friendly pour des personnes non adepte de la technologie.
- Le site doit être responsive afin de pouvoir l'utiliser sur tout type de d'écran.

5 Mockup

5.1 Login



The mockup shows a login interface with a blue header bar. Below it, a white box contains two input fields: 'Email *' and 'Mot de passe *'. At the bottom of the box are two blue buttons: 'Connection' and 'Inscription'.

Figure 1.5.1: Page de connection

5.2 Accueil

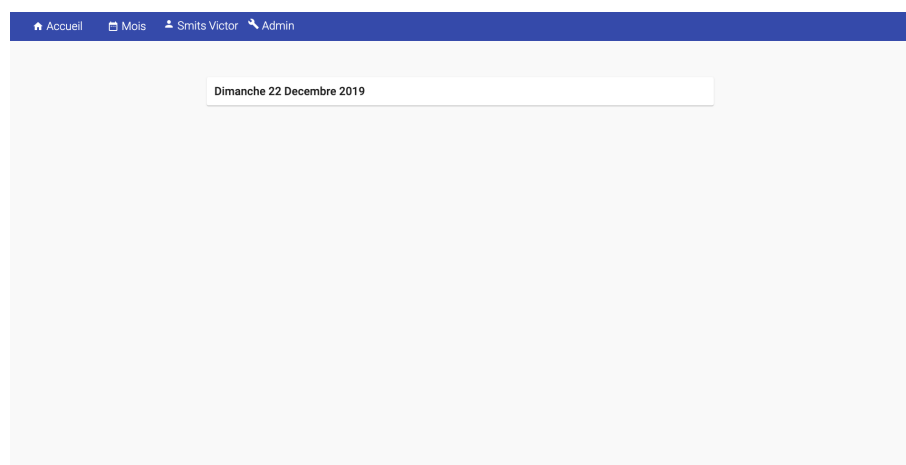
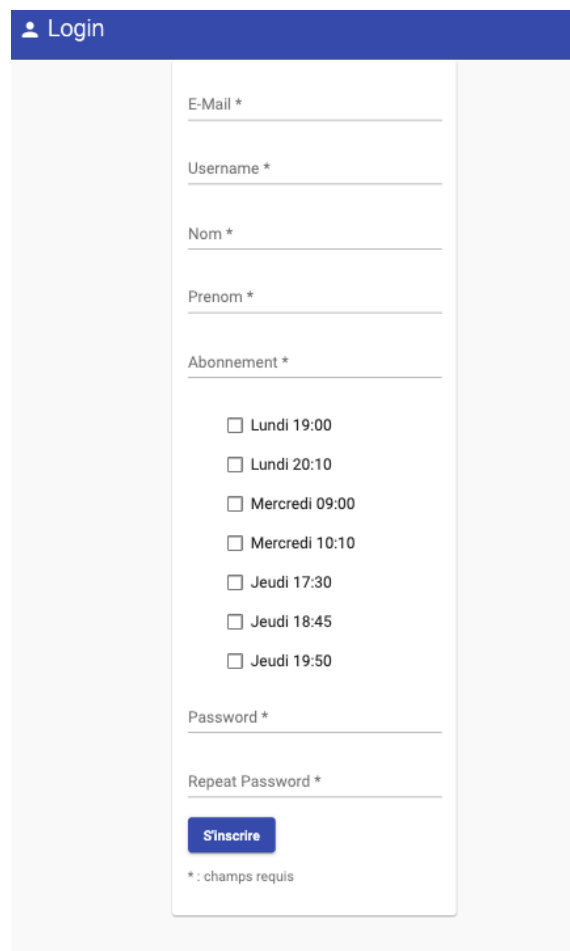


Figure 1.5.2: Page d'accueil

5.3 Inscription



Login

E-Mail *

Username *

Nom *

Prenom *

Abonnement *

☐ Lundi 19:00

☐ Lundi 20:10

☐ Mercredi 09:00

☐ Mercredi 10:10

☐ Jeudi 17:30

☐ Jeudi 18:45

☐ Jeudi 19:50

Password *

Repeat Password *

S'inscrire

* : champs requis

Figure 1.5.3: Page d'inscription au site

5.4 Month

Date	Heure	Velo(s) libre(s)	Cours confirmé	Liste Participant(e)s	Inscription/Désinscription
Lundi 23	19:00	8	✓		
Lundi 23	20:10	7	✓		
Mercredi 25	09:00	9	✓		
Mercredi 25	10:10	9	✓		
Jeudi 26	17:30	9	✓		
Jeudi 26	18:45	9	✓		
Jeudi 26	19:50	9	✓		
Lundi 30	19:00	8	✓		
Lundi 30	20:10	8	✓		
Mercredi 1	09:00	9	✓		
Mercredi 1	10:10	9	✓		

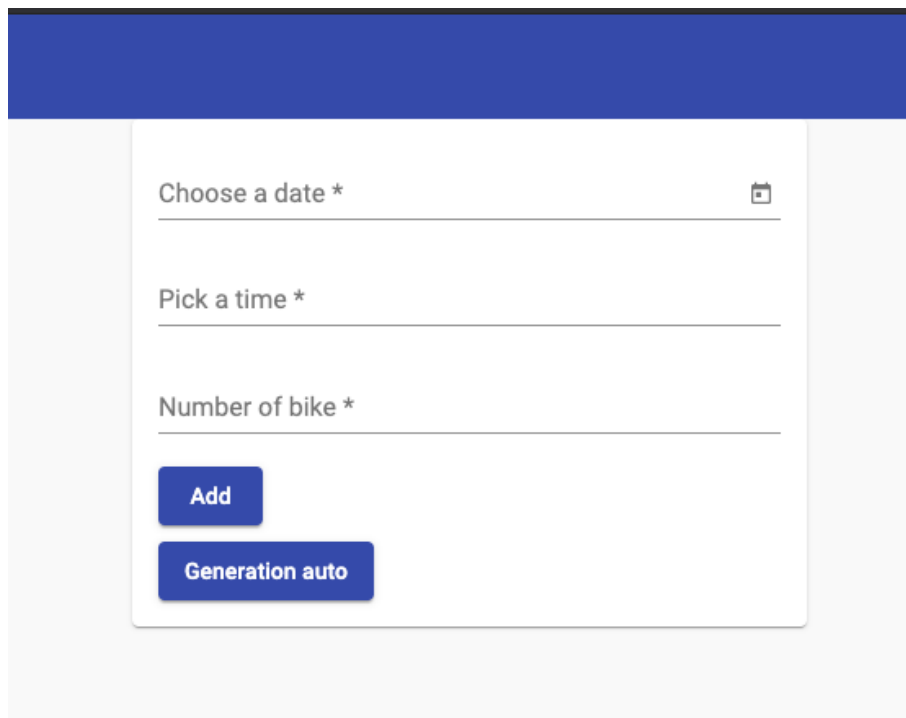
Figure 1.5.4: Page d’affichage des sessions du mois

5.5 Admin - Session

Date	Heure	Velo libre	Cours confirmé	List Inscrit	Annulation / Suppression
Lundi 9	20:10	9	✓		
Lundi 9	19:00	8	✓		
Mercredi 11	09:00	9	✓		
Mercredi 11	10:10	9	✓		
Jeudi 12	17:30	9	✓		
Jeudi 12	18:45	9	✓		
Jeudi 12	19:50	9	✓		
Dimanche 15	12:00	9	✓		
Lundi 16	20:10	9	✓		
Lundi 16	19:00	9	✓		
Mercredi 18	09:00	9	✓		
Mercredi 18	10:10	9	✓		
Jeudi 19	17:30	9	✓		

Figure 1.5.5: Page de gestion des sessions du mois pour l’administrateur

5.6 Admin - Nouvelle session



Choose a date *

Pick a time *

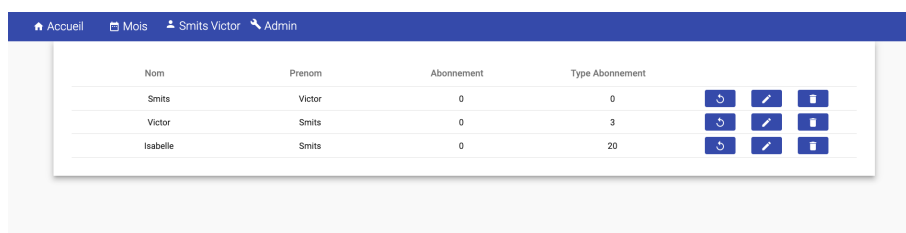
Number of bike *

Add

Generation auto

Figure 1.5.6: Page de création d'une nouvelle session

5.7 Admin - Abonnement












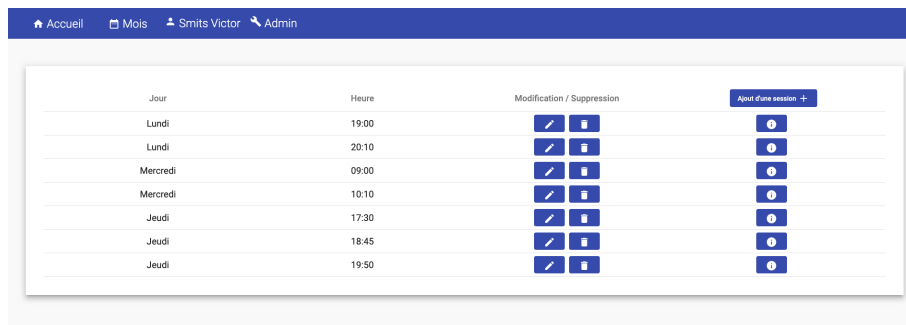
Nom	Prenom	Abonnement	Type Abonnement	
Smits	Victor	0	0	  
Victor	Smits	0	3	  
Isabelle	Smits	0	20	  

Figure 1.5.7: Page de gestion des abonnements des utilisateurs

5.8 Admin - Type Session
























Jour	Heure	Modification / Suppression	Ajout d'une session +
Lundi	19:00	 	
Lundi	20:10	 	
Mercredi	09:00	 	
Mercredi	10:10	 	
Jeudi	17:30	 	
Jeudi	18:45	 	
Jeudi	19:50	 	

Figure 1.5.8: Page de gestion des types de sessions

2 | Base de donnée

1 Diagramme Entité relation

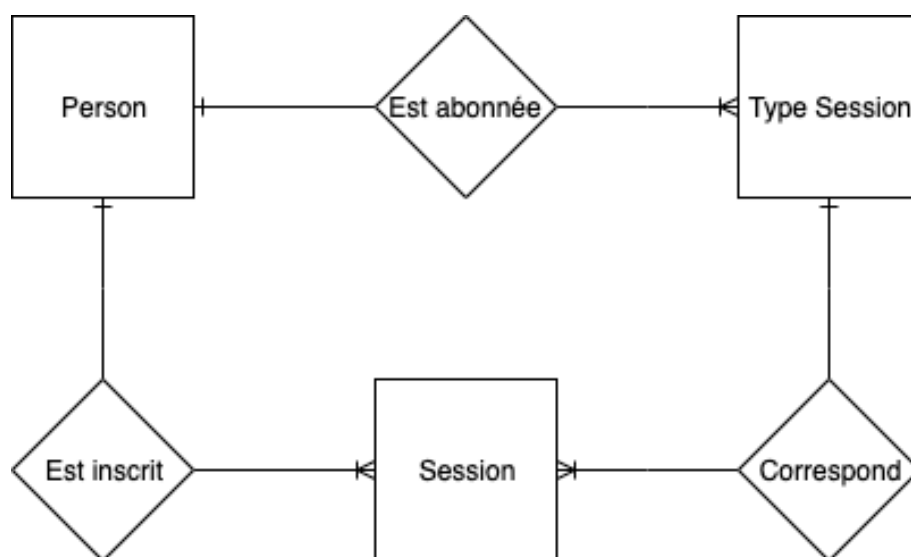


Figure 2.1.1: Diagramme Relationnel

2 Diagramme Relationnel

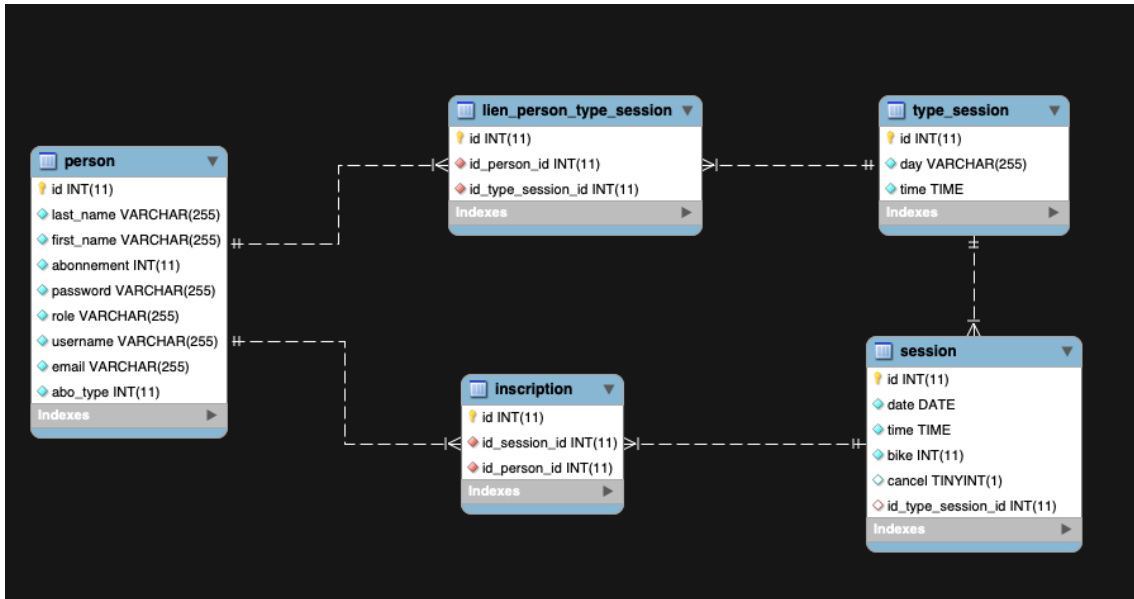


Figure 2.2.2: Diagramme Relationnel

3 Description des tables et des relations entre tables

Table	Table	Type	Description
Person	Inscription	1 à N	Une personne possède plusieurs inscriptions.
Session	Inscription	1 à N	Une session à plusieurs inscriptions.
Session	Type Session	N à 1	Plusieurs sessions corresponde à un type de sessions, un type de session peut correspondre à plusieurs sessions.
Person	Lien person type session	1 à N	Une Personne possède plusieurs liens, plusieurs liens corresponde à une personne.
Type Session	Lien person type session	1 à N	Un type de session possède plusieurs liens, plusieurs liens corresponde à un seul type de session.

4 Description précise des champs

4.1 Table : Person

Nom	Key	NN	Type	Référence	Intitulé	Commentaire
id	PK	O	int		Identifiant	
last_name		O	varchar(45)		Nom	
frist_name		O	varchar(45)		Prenom	
abonnement		O	int		Nombre de session restante	
password		O	varchar(45)		Mot de passe crypté	
role		O	varchar(45)		Role sur le site	Default = role_user
username		O	varchar(45)		Nom d'utilisateur	Unique
email		O	varchar(45)		Adresse Email	Unique
abo_type		O	int		abonnement origine	

4.2 Table : Session

Nom	Key	NN	Type	Référence	Intitulé	Commentaire
id	PK	O	int		Identifiant	
date		O	DATE		date	
time		O	Time		heure	
bike		O	int		Nombre de vélo restante	
cancel		O	Boolean		Status de la session	Default = Null
id type session	FK	O	int	id / typeSession	Lien avec le type de session	Default = Null

4.3 Table : Type_Session

Nom	Key	NN	Type	Référence	Intitulé	Commentaire
id	PK	O	int		Identifiant	
day		O	varchar(45)		code du jour	
time		O	Time		heure	

4.4 Table : inscription

Nom	Key	NN	Type	Référence	Intitulé	Commentaire
id	PK	O	int		Identifiant	
id_session	FK	O	int	id / session	lien avec la session	
id_person	FK	O	int	id / person	lien avec la person	

4.5 Table : lien_person_type_session

Nom	Key	NN	Type	Référence	Intitulé	Commentaire
id	PK	O	int		Identifiant	
id_type_session	FK	O	int	id / type_session	lien avec le type de session	
id_person	FK	O	int	id / person	lien avec la person	

3 | MVC

1 User Stories compatible Angular - Twig

Seul l'aspect graphique changera entre la version Twig et la version Angular. Pour plus de détail rendez-vous au chapitre 5

1.1 En tant qu'utilisateur, je dois pouvoir me connecter sur le site

1. L'utilisateur se rend sur la page de connection en cliquant sur *Login*
2. Il entre son Nom d'utilisateur et son mot de passe.
3. Après vérification l'utilisateur est rediriger vers la page d'accueil.

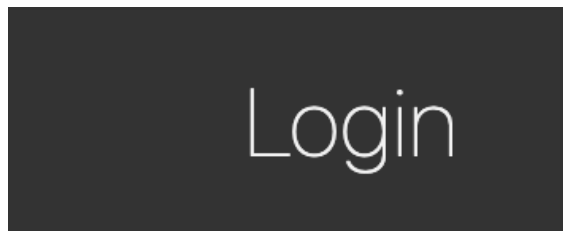


Figure 3.1.1: Bouton de connection

Gestion des erreurs et des risques

En cas d'erreur dans les informations de connection fournis par l'utilisateurs, une erreurs sera produite et afficher sur le site.

Figure 3.1.2: Formulaire de connection

Diagramme de séquence

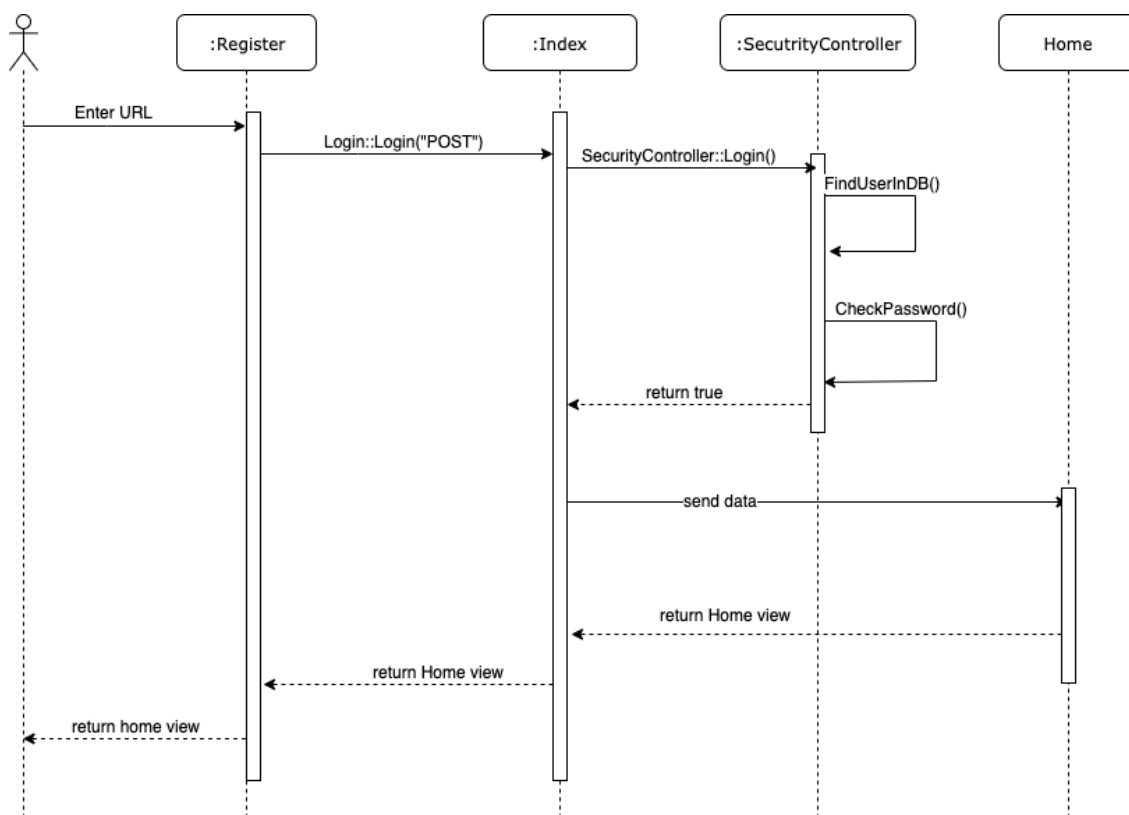
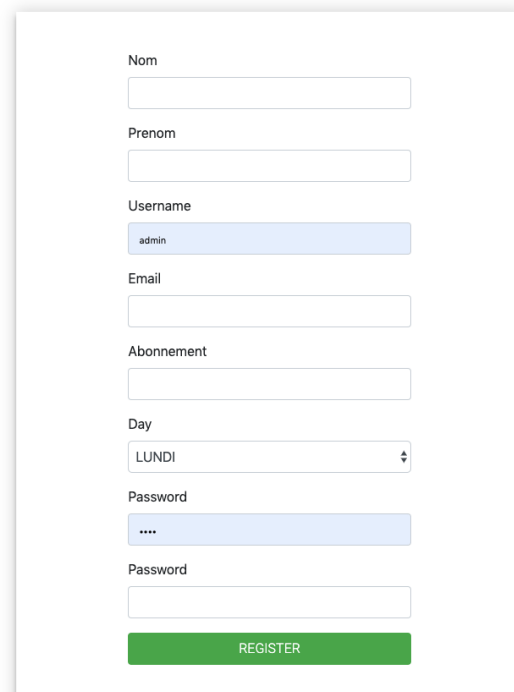


Figure 3.1.3: Diagramme de séquence de la connections d'un utilisateur

Script concernés

- login.html.twig
- Person.php
- SecurityController.php

1.2 En tant qu'utilisateur, je dois pouvoir me créer un compte sur le site



The image shows a registration form with the following fields and elements:

- Nom**: A text input field.
- Prenom**: A text input field.
- Username**: A text input field with the value "admin" displayed.
- Email**: A text input field.
- Abonnement**: A text input field.
- Day**: A dropdown menu with "LUNDI" selected.
- Password**: A text input field with four asterisks "****" displayed.
- Password**: A second text input field for password confirmation.
- REGISTER**: A green button at the bottom of the form.

Figure 3.1.4: Formulaire d'inscription

1. L'utilisateur rentre s'est information dans le formulaire.
2. L'utilisateur sélectionne le type de session au quelle il est inscrit.
3. L'utilisateur clique sur *Register* et attend la redirection.

Gestion des erreurs et sécurité

- L'utilisateur doit remplir tout les champs, si un champs est manquant une erreur apparaîtra.
- Le nom d'utilisateur et l'adresse Email son unique dans le système, si il entre un Email ou nom d'utilisateur deja existant une erreur lui dira de changer la faute.
- L'utilisateur doit confirmer sont mot de passe, si les mots de passe sont différent ou plus petit que 6 caractère il y aura une erreur.

Diagramme de séquence

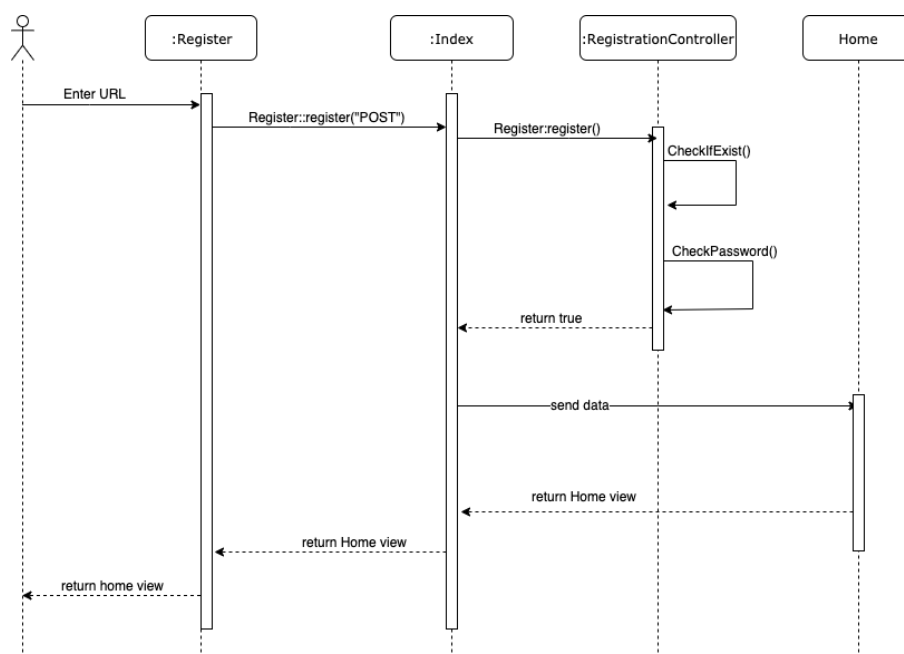
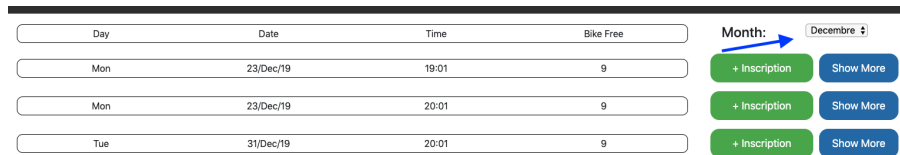


Figure 3.1.5: Formulaire d'inscription

Script concernés

- RegistrationController.php
- register.html.twig
- Person.php

1.3 En tant qu'utilisateur, je dois pouvoir sélectionner le mois et l'année pour afficher les sessions qui m'intéresse



Day	Date	Time	Bike Free
Mon	23/Dec/19	19:01	9
Mon	23/Dec/19	20:01	9
Tue	31/Dec/19	20:01	9

Month: Decembre ▾

+ Inscription Show More

+ Inscription Show More

+ Inscription Show More

Figure 3.1.6: selection du mois et de l'année

1. L'utilisateur choisit le mois et l'année.
2. la liste se rafraîchis afin de correspondre à la selection

1.4 En tant qu'utilisateur, je dois pouvoir m'inscrire à une session un certain jour

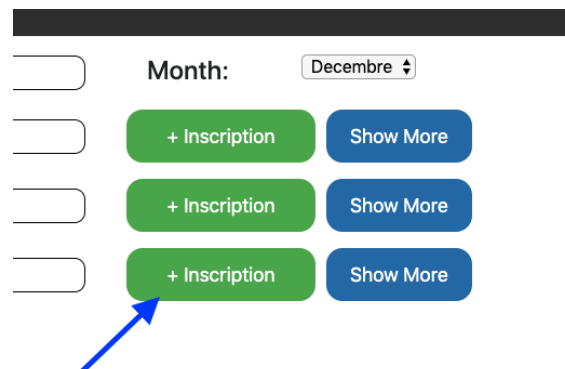


Figure 3.1.7: bouton d'inscription

1. l'utilisateur choisit la session correspondante à sa demande.
2. l'utilisateur click sur le bouton *+ Inscription*

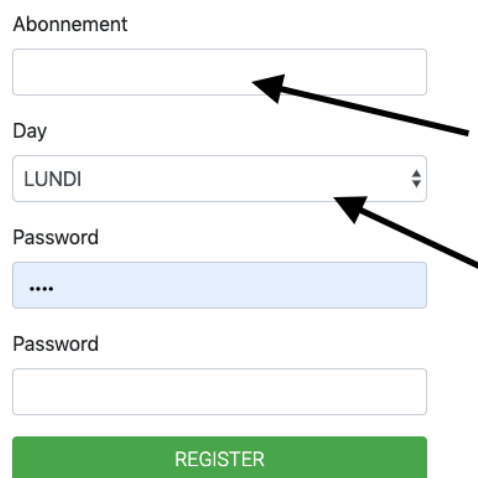
Gestion des erreurs

- Si l'utilisateur ne possède plus d'abonnement une erreur sera afficher lui signalant le problème.
- Si la session est complète un message d'erreur sera affiché.

Script concerné

- MonthController.php
- month.html.twig
- Session.php

1.5 En tant qu'utilisateur, je dois pouvoir être inscrit automatiquement à une session



The image shows a registration form with the following elements:

- Abonnement**: A text input field with a black arrow pointing to it from the right.
- Day**: A dropdown menu showing "LUNDI" with a black arrow pointing to it from the right.
- Password**: A password input field with a light blue background and masked characters "....".
- Password**: A second, empty password input field.
- REGISTER**: A green button with white text.

Figure 3.1.8: Selection pour l'inscription automatique

1. L'utilisateur introduit le nombre de session pour le quelle il paye.
2. Il selection les sessions pour les quelle il souhaite s'inscrire.
3. Il clique sur *Register*.

Gestion des erreurs

Si une erreur arrive durant l'inscription l'utilisateur ne sera pas rediriger et aura un message d'erreur.

Script concernés

- RegistrationController.php
- register.html.twig
- Person.php
- Inscription.php

1.6 En tant qu'utilisateur, je dois pouvoir me désinscrire à une session



Figure 3.1.9: Bouton de désinscription à une session

1. L'utilisateur trouve la session qui l'intéresse.
2. Le check vert indique qu'il est inscrit.
3. Il clique sur le bouton.

Script concernés

- MonthController.php
- month.html.twig
- Person.php
- Inscription.php

1.7 En tant qu'utilisateur, je dois pouvoir voir qui est inscrit pour chaque session

1. L'utilisateur trouve la session qui l'intéresse.
2. L'utilisateur clique sur le bouton *Show More* dans la dernière colonne.
3. Un tableau s'affiche avec la liste des inscrit(e)s.

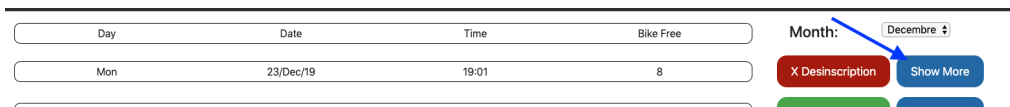


Figure 3.1.10: Bouton d'affichage

Nom	Prenom	Date	23/Dec/19
Smits	Victor	Time	19:01
		Bike Free	8

Figure 3.1.11: Liste Participant(e)s

Script concernés

- MonthController.php
- month.html.twig
- Person.php
- Inscription.php

1.8 En tant qu'utilisateur, je veux pouvoir voir dynamiquement le nombre de séance qui reste dans mon abonnement

1. L'utilisateur clique sur son nom dans la bar de navigation.
2. L'utilisateur est redirigé vers sa page de profil.
3. L'utilisateur retrouve l'information dans le cadre afficher sur la page.



Figure 3.1.12: Bouton de navigation vers le profil

Day	Date	Time	Bike Free		
Thu	10/Oct/19	19:01	8	X Desinscription	Show More
Wed	16/Oct/19	19:01	8	X Desinscription	Show More
Wed	16/Oct/19	20:01	8	X Desinscription	Show More
Thu	17/Oct/19	19:01	7	X Desinscription	Show More
Thu	17/Oct/19	20:01	7	X Desinscription	Show More
Mon	23/Dec/19	19:01	8	X Desinscription	Show More

Nom
Victor
Prenom
Email
Abonnement
Jour
Edit Profile

Smits
Victor
victor-sm@hotmai.fr
13
Mon

Figure 3.1.13: Nombre de séance restante dans l'abonnement

Script concernés

- ProfileController.php
- profile.html.twig
- Person.php

1.9 En tant qu'utilisateur, je veux pouvoir modifier mon profil

1. L'utilisateur clique sur son nom dans la bar de navigation.
2. L'utilisateur clique sur le bouton edit en dessous de son profil.
3. Il est rediriger vers la page de modification.
4. L'utilisateur rentre les information qu'il souhaite changer.
5. L'utilisateur confirm en cliquant sur *Edit*



Figure 3.1.14: Bouton de navigation vers le profil

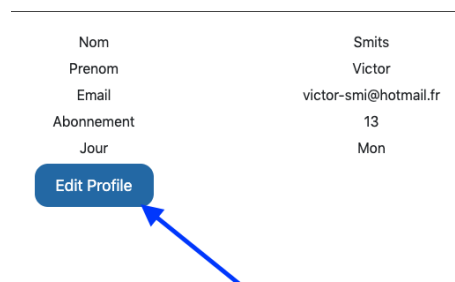
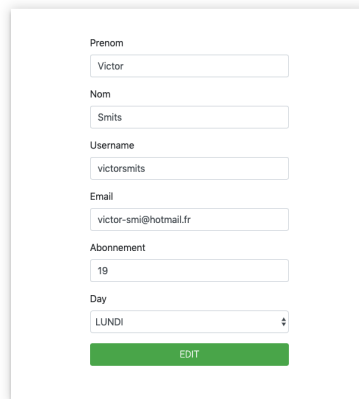


Figure 3.1.15: Bouton de modification du profil



Prenom
Victor

Nom
Smits

Username
victorsmits

Email
victor-smi@hotmail.fr

Abonnement
19

Day
LUNDI

EDIT

Figure 3.1.16: Fenêtre de modification des données du profil

Script concernés

- ProfileController.php
- profile.html.twig
- Person.php

1.10 En tant qu'administrateur, je dois pouvoir annuler une session

1. L'administrateur se connecte au site avec ses identifiants.
2. Un nouveau bouton de navigation apparaît dans la barre de navigation.
3. L'administrateur clique sur le bouton *Admin*
4. Il sélectionne *Session* dans le menu déroulant.
5. L'administrateur atterrit sur la page de gestion des Sessions.
6. Il clique sur la croix correspondante à la session qu'il souhaite annuler.

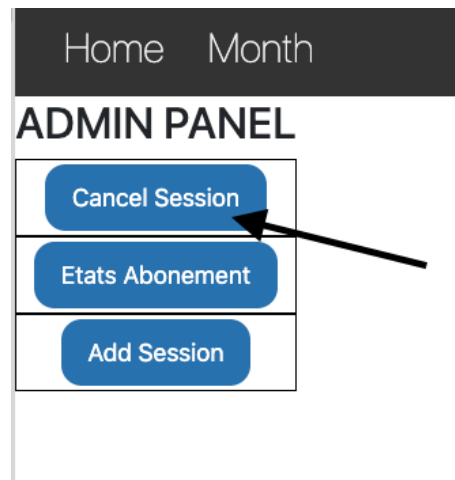


Figure 3.1.17: Bouton de navigation de l'administrateur

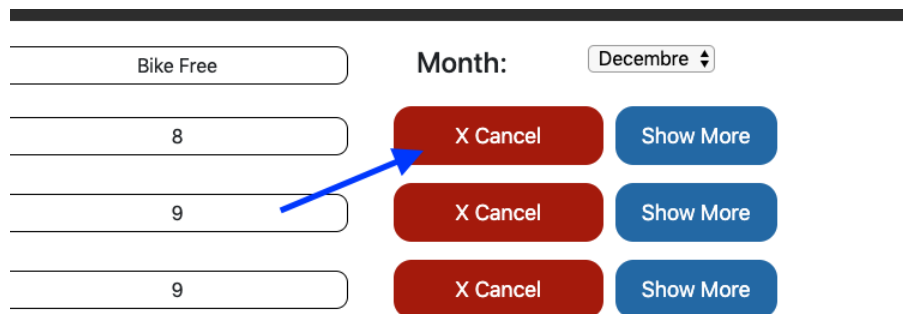


Figure 3.1.18: Bouton d'annulation de la session

Script concernés

- SessionAdministrationController.php
- SessionAdmin.html.twig
- Session.php

1.11 En tant qu'administrateur, je dois pouvoir créer une nouvelle session peut importe la date

1. L'administrateur se connecte au site avec ses identifiant.
2. Un nouveau bouton de navigation apparait dans la bar de navigation.
3. L'administrateur clique sur le bouton *Admin*
4. Il sélectionne *Nouvelle Session* dans le menu déroulant.
5. L'administrateur atterris sur la page d'ajout de Sessions.
6. Il remplis le formulaire avec les bonne informations.
7. Il clique sur *Add*

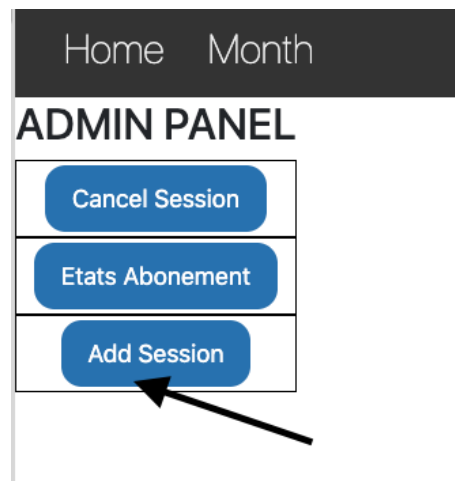


Figure 3.1.19: Bouton de navigation de l'administrateur

The form contains the following elements:

- Date:** A text input field with the placeholder text "jj/mm/aaaa".
- Time:** Two dropdown menus, each showing "00", separated by a colon.
- Bike:** A text input field.
- ADD:** A green button with the text "ADD".

Figure 3.1.20: Formulaire d'ajout d'une session

Gestion des erreurs

- Tout les champs doivent être remplis, si ce n'est pas le cas il sera impossible d'envoyer le formulaire et l'administrateur aura une erreur afficher.

Diagramme de séquence

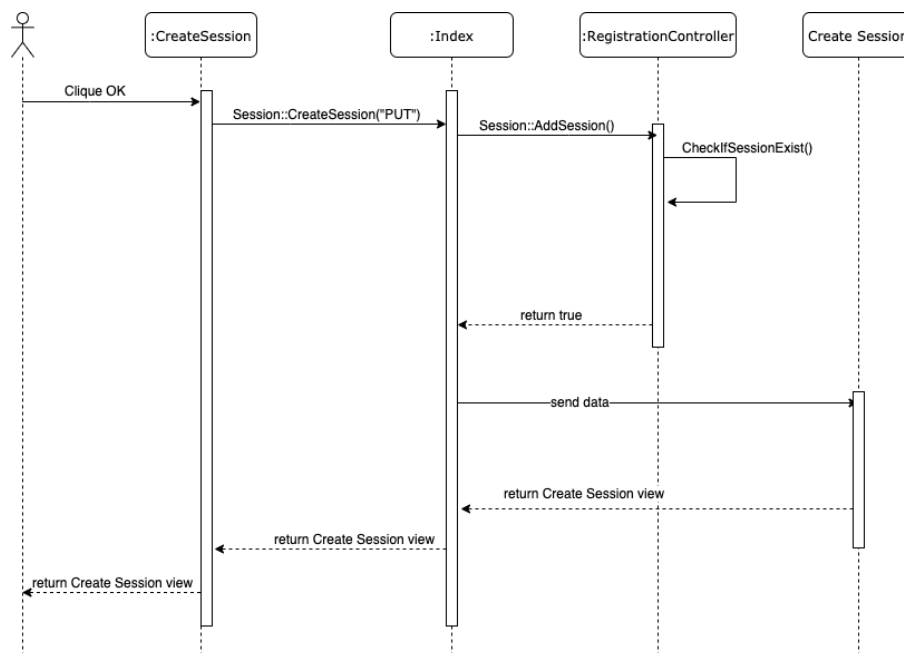


Figure 3.1.21: Diagramme de séquence de l'ajout d'une session

Script concerné

- Session.php
- CreateSessionController.php
- session.html.twig

1.12 En tant qu'administrateur, je dois pouvoir gérer les abonnements des utilisateurs

1. L'administrateur se rend sur le panel admin.
2. Il sélectionne *Etats Abonnement* dans le menu.
3. L'administrateur atterrit sur la page de gestion des abonnements
4. Il clique sur *Renouveler* pour relancer l'abonnement d'un utilisateur.

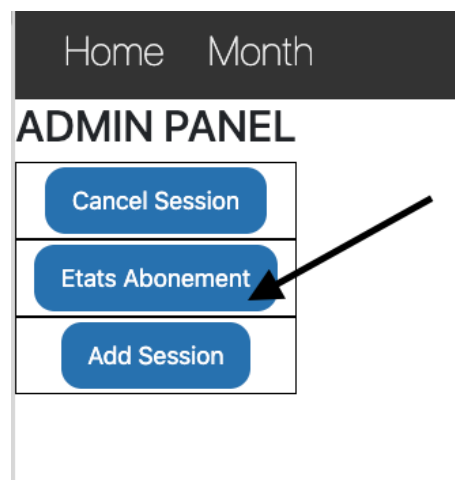


Figure 3.1.22: Bouton Etats Abonnement

Nom	Prenom	Email	Abonnement	
Smiths	Victor	victor-smith@hotmail.fr	14	+ Renouvellement
Smiths	Victor	victor-smith@hotmail.com	0	+ Renouvellement
smiths	victor	16107@ecan.be	16	+ Renouvellement
Isabelle	Smiths	test@test.com	20	+ Renouvellement

Figure 3.1.23: Bouton de renouvellement de l'abonnement

Script concerné

- Person.php
- AbonnementController.php
- abonnement.html.twig

4 | API

1 Tableau recapitulatif

Méthode HTTP	Requête	Contrôleur	Méthode	Action
GET	/api	HomeControllerAPI	index	renvoie les informations des sessions d'aujourd'hui.
GET	/api/month/{month?}/{year?}	MonthControllerAPI	index	renvoie la liste des sessions du mois.
GET	/api/profile/{username?}	ProfileControllerAPI	index	Renvoie le profile de l'utilisateur.
GET	/api/admin/abonnement	AbonnementControllerAPI	index	renvoie la liste des utilisateurs.
GET	/api/TypeSession	RegistrationControllerApi	getTypeSession	Renvoie la liste des types de session.
PUT	/api/admin/renewAbo	AbonnementControllerAPI	RenewAbo	renouvelle l'abonnement de l'utilisateurs et l'inscrit à ses sessions.
PUT	/api/admin/editAbo	AbonnementControllerAPI	editAbo	Change l'abonnement de l'utilisateur.
PUT	/api/admin/Cancel	SessionAdministration-ControllerApi	cancel-Session	Annule la session et désinscrit les utilisateurs.
PUT	/api/admin/recreate	SessionAdministration-ControllerApi	recreate-Session	Recrée la session.
PUT	/api/admin/autocreate	CreateSessionControllerApi	createSession	Génère des sessions pour x années.

PUT	/api/editProfile	ProfileControllerApi	editProfile	Modifie les données de l'utilisateur.
PUT	/api/TypeSession	RegistrationControllerApi	editTypeSession	Modifie les données du type de session.
DELETE	/api/admin/session/{id?}	SessionAdministration-ControllerApi	deleteSession	Supprime la session.
DELETE	/api/Desinscription/{username?}/{id?}	MonthControllerAPI	remove-Inscription	Désinscrit l'utilisateur de la session.
DELETE	/api/TypeSession/{id?}	RegistrationControllerApi	Delete-TypeSession	Supprime le type de session.
DELETE	/api/admin/user/{id?}	AbonnementControllerApi	delAbo	Supprime l'utilisateur et ses références.
POST	/api/login	SecurityControllerApi	login	Authentifie l'utilisateur.
POST	/api/admin/session	CreateSessionControllerApi	index	Créer une nouvelle session.
POST	/api/Inscription	MonthControllerAPI	create-Inscription	Inscrit l'utilisateur à la session.
POST	/api/TypeSession	RegistrationControllerApi	setType-Session	Créer un nouveau type de session.
POST	/api/register	RegistrationControllerApi	register	Créer un nouveau utilisateur et l'inscrit a ses sessions.

2 GET : /api

Renvoie un JSON contenant toutes les sessions du jour ainsi que les inscriptions.

2.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

text/plain

Body

Pas de body pour la requête.

2.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id de la session.
Date	DateTime	O	Date de la session.
Time	DateTime	O	Heure de la session.
Bike	int	O	Nombre de vélo disponible pour la session.
Cancel	Boolean	O	Etats de la session.
IdTypeSession	int	O	id du type de session correspondant.
idInscription	List	O	Liste des participants.
idPerson	Person	O	Identité du participant.
lastName	String	O	Nom du participant.
firstName	String	O	Prénom du participant.

Exemple JSON

```
1 {
2   id: 15,
3   Date: "2019/12/23 00:00",
4   time: "1970/01/01 19:00",
5   bike: 8,
6   Cancel: false,
7   IdTypeSession: 1,
8   idInscription: [{
9     0: {
10       id: 45,
11       idSession: 15,
12       idPerson: {
13         id: 49,
14         lastName: "Isabelle",
15         firstName: "Smits"
16       }
17     },
18   }]
19 }
```

Code

HomeControllerApi.php

3 GET : /api/month

Renvoie une liste de JSON contenant toute les sessions du mois.

3.1 Requête

paramètre du path

Propriété	Type	Obligatoire	Description
Month	int	O	numéro du mois sélectionné.
Year	int	O	Année sélectionné.

Content type

text/plain

Body

Pas de body pour la requête.

3.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id de la session.
Date	DateTime	O	Date de la session.
Time	DateTime	O	Heure de la session.
Bike	int	O	Nombre de vélo disponible pour la session.
Cancel	Boolean	O	Etats de la session.
idInscription	List	O	Liste des participants.
idPerson	Person	O	Identité du participant.
lastName	String	O	Nom du participant.
firstName	String	O	Prénom du participant.

Exemple JSON

```
1 {
2   id: 15,
3   Date: "2019/12/23 00:00",
4   time: "1970/01/01 19:00",
5   bike: 8,
6   Cancel: false,
7   idInscription: [{
8     0: {
9       id: 45,
10      idSession: 15,
11      idPerson: {
12        id: 49,
13        lastName: "Isabelle",
14        firstName: "Smits"
15      }
16    },
17  ]
18 }
```

Code

MonthControllerApi.php

4 GET : /api/profile

Renvoie un de JSON contenant toute les informations de l'utilisateur

4.1 Requête

paramètre du path

Propriété	Type	Obligatoire	Description
Username	String	O	Nom d'utilisateur à afficher.

Content type

text/plain

Body

Pas de body pour la requête.

4.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id de l'utilisateur.
lastName	String	O	Nom du participant.
firstName	String	O	Prénom du participant.
Abonnement	int	O	Nombre de session restante dans l'abonnement.
UserName	String	O	Nom d'utilisateur.
Email	String	O	Email de l'utilisateur.
AboType	int	O	Type d'abonnement de l'utilisateur.
Role	String	O	Role donnant les accès a l'utilisateur.
idInscription	List<Session>	O	Liste des session auquel l'utilisateur est inscrit.
id	int	O	Id de la session.
Date	DateTime	O	Date de la session.
Time	DateTime	O	Heure de la session.
Bike	int	O	Nombre de vélo disponible pour la session.
Cancel	Boolean	O	Etats de la session.

Exemple JSON

```
1 {  
2   id: 49,  
3   LastName: "Isabelle",  
4   FirstName: "Smits",  
5   Abonnement: 0,  
6   role: "ROLE_USER",  
7   Username: "isa",  
8   Email: "isa_smi@hotmail.com",  
9   AboType: 20,  
10  idInscription: [{  
11    0: {  
12      id: 45,  
13      idSession: {  
14        id: 15,  
15        date: "2019/12/23 00:00",  
16        time: "1970/01/01 19:00",  
17        bike: 8,  
18        cancel: false  
19      }  
20    }  
21  }]  
22 }
```

Code

ProfileControllerApi.php

5 GET : /api/admin/abonnement

Renvoie une liste de JSON contenant toute les informations des l'utilisateurs.

5.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

text/plain

Body

Pas de body pour la requête.

5.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id de l'utilisateur.
lastName	String	O	Nom du participant.
firstName	String	O	Prénom du participant.
Abonnement	int	O	Nombre de session restante dans l'abonnement.
UserName	String	O	Nom d'utilisateur.
Email	String	O	Email de l'utilisateur.
AboType	int	O	Type d'abonnement de l'utilisateur.
Role	String	O	Role donnant les accès a l'utilisateur.
idTypeSession	List<LienPerson- TypeSession>	O	Liste des session auquel l'utilisateur est inscrit.
id	int	O	Id du lien entre la person et le type de session.
idPerson	int	O	Id de la person.
IdTypeSession	JSON<TypeSession>	O	Type de session.
id	int	O	Id du type de session.
day	String	O	Jour du type de session.
Time	DateTime	O	Heure du type de session.

Exemple JSON

```
1 {  
2   id: 49,  
3   LastName: "Isabelle",  
4   FirstName: "Smits",  
5   Abonnement: 0,  
6   role: "ROLE_USER",  
7   Username: "isa",  
8   Email: "isa_smi@hotmail.com",  
9   AboType: 20,  
10  idTypeSession: [{  
11    0: {  
12      id: 55  
13      IdPerson: 1  
14      IdTypeSession: {  
15        id: 2  
16        day: "Mon"  
17        time: "1970/01/01 20:10"  
18      }  
19    }  
20  }]  
21 }
```

Code

AbonnementControllerApi.php

6 GET : /api/TypeSession

Renvoie une liste de JSON contenant toute les types de sessions.

6.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

text/plain

Body

Pas de body pour la requête.

6.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id du type de session.
day	String	O	Jour du type de session.
Time	DateTime	O	Heure du type de session.
idTypeSession	List<LienPerson- TypeSession>	O	Liste des session auquel l'utilisateur est inscrit.
id	int	O	Id du lien entre la person et le type de session.
idPerson	int	O	Id de la person.
id	int	O	Id de l'utilisateur.
lastName	String	O	Nom du participant.
firstName	String	O	Prénom du participant.

Exemple JSON

```
1 {
2   id: 2
3   day: "Mon"
4   time: "1970/01/01 20:10"
5   idTypeSession: [{
6     0: {
7       id: 55
8       IdPerson: {
9         id: 1,
10        LastName: "Isabelle",
11        FirstName: "Smits"
12      }
13    }
14  }]
15 }
```

Code

ProfileControllerApi.php

7 PUT : /api/admin/renewAbo

Requête POST pour le renouvellement de l'abonnement d'un utilisateur.

7.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

application/json

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id de l'utilisateur.

Exemple JSON

```
1 {  
2   id: 1  
3 }
```

7.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

AbonnementControllerApi.php

8 PUT : /api/admin/editAbo

Requête POST pour modifier l'abonnement d'un utilisateur.

8.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

text/plain

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id de l'utilisateur.
FirstName	String	N	Prénom de l'utilisateur.
LastName	String	N	Nom de l'utilisateur.
aboType	int	O	Type d'abonnement de l'utilisateur.

Exemple JSON

```
1 {  
2   FirstName: "Victor"  
3   LastName: "Smits"  
4   Id: 1  
5   aboType: 0  
6 }
```

8.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

AbonnementControllerApi.php

9 PUT : /api/admin/cancel

Requête POST pour annuler une Session et automatiquement désinscrire toute les personnes etant inscrite.

9.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

application/json

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id de la session.

Exemple JSON

```
1 {  
2   id: 1  
3 }
```


9.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

AbonnementControllerApi.php

10 PUT : /api/admin/recreate

Requête POST pour annuler une Session et automatiquement désinscrire toute les personnes etant inscrite.

10.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

application/json

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id de la session.
bike	int	O	Nombre de vélo dans la session.

Exemple JSON

```
1 {  
2   id: 1,  
3   Bike : 9,  
4 }
```

10.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

SessionAdministrationControllerApi.php

11 PUT : /api/admin/autocreate

Requête POST pour générer automatiquement les sessions, correspondante au type de session sélectionné, sur x année .

11.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

application/json

Body

Propriété	Type	Obligatoire	Description
year	int	O	Nombre d'année pour la génération automatique.
bike	int	O	Nombre de vélo dans la session.
idTypeSession	List<int>	O	Liste des id des type de sessions à générer.

Exemple JSON

```
1 {  
2   Bike: 9  
3   year: 1  
4   idTypeSession: [1,2,3,4]  
5 }
```

11.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

CreateSessionControllerApi.php

12 PUT : /api/editProfile

Requête POST permettant à un utilisateur modifier son profil.

12.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

application/json

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id de l'utilisateur.
lastName	String	O	Nom de l'utilisateur.
firstName	String	O	Prénom de l'utilisateur.
password	String	O	Nouveau mot de passe de l'utilisateur.
typeSessions	List<TypeSession>	O	Liste des types de sessions auquel l'utilisateur est inscrit.
id	int	O	Id du type de session.
Day	String	O	Jour du type de session.
Time	String	O	Heure du type de session.

Exemple JSON

```
1 {  
2   id: 1  
3   lastName: "Smits"  
4   firstName: "Victor"  
5   Email: "victor-smi@hotmail.com"  
6   password: null  
7   typeSessions: [{  
8     0: {  
9       Id: 2  
10      Day: "Lundi"  
11      Time: "20:10"  
12    }  
13  }]  
14 }
```

12.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

EditProfileControllerApi.php

13 PUT : /api/TypeSession

Requête POST pour la modification d'un type de session. Après modification, les sessions correspondante à l'ancien type de session sont supprimée, pour pouvoir en créer de nouvelle il faut passer par la génération automatique.

13.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

application/json

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id de l'utilisateur.
Day	String	O	Jour du type de la session.
Time	String	O	Heure du type de la session.

Exemple JSON

```
1 {  
2   id: 1  
3   Day: "MON"  
4   Time: "20:10"  
5 }
```

13.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

RegistrationControllerApi.php

14 DELETE : /api/admin/session

Requête permettant d'effacer une session du système.

14.1 Requête

paramètre du path

Propriété	Type	Obligatoire	Description
id	int	O	id de la session à effacer.

Content type

text/plain

Body

Pas de body pour la requête.

14.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

SessionAdministrationControllerApi.php

15 DELETE : /api/Desinscription

Requête permettant de désinscrire un utilisateur d'une session.

15.1 Requête

paramètre du path

Propriété	Type	Obligatoire	Description
username	String	O	Nom d'utilisateur.
id	int	O	Id de la session.

Content type

text/plain

Body

Pas de body pour la requête.

15.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

MonthControllerApi.php

16 DELETE : /api/TypeSession

Requête permettant de supprimer un type de session ainsi qu'automatiquement les sessions qui y sont liées.

16.1 Requête

paramètre du path

Propriété	Type	Obligatoire	Description
id	int	O	Id du type de session.

Content type

text/plain

Body

Pas de body pour la requête.

16.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

RegistrationControllerApi.php

17 DELETE : /api/admin/user

Requête permettant de supprimer un utilisateur ainsi qu'automatiquement le désinscrire des sessions qui lui sont liées.

17.1 Requête

paramètre du path

Propriété	Type	Obligatoire	Description
id	int	O	Id de l'utilisateur.

Content type

text/plain

Body

Pas de body pour la requête.

17.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

AbonnementControllerApi.php

18 POST : /api/login

Requête POST permettant à un utilisateur de se connecter au site.

18.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

application/json

Body

Propriété	Type	Obligatoire	Description
Username	String	O	Nom d'utilisateur servant à la connexion.
password	String	O	Mot de passe de l'utilisateur.

Exemple JSON

```
1 {  
2   Username: "victor-smi@hotmail.com"  
3   password: "0000"  
4 }
```

18.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

SecurityControllerApi.php

19 POST : /api/admin/session

Requête PUT permettant l'ajout d'une nouvelle session dans le système.

19.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

application/json

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id de l'utilisateur.
Date	String	O	Date de la session.
Time	String	O	Heure de la session.
Bike	int	O	Nombre de vélos disponible.
Cancel	Boolean	O	Etats de la session.

Exemple JSON

```
1 {  
2   id: 0,  
3   Day: "Mon Dec 23 2019",  
4   Time: "20:10",  
5   Bike : 9,  
6   Cancel : false  
7 }
```

19.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

CreateSessionControllerApi.php

20 POST : /api/Inscription

Requête PUT permettant l'ajout d'une nouvelle session dans le système.

20.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

application/json

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id de la session.
Username	String	O	Nom d'utilisateur.

Exemple JSON

```
1 {  
2   Username : "admin",  
3   Id : 262  
4 }
```

20.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

CreateSessionControllerApi.php

21 POST : /api/TypeSession

Requête PUT permettant l'ajout d'un nouveau type de session.

21.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

application/json

Body

Propriété	Type	Obligatoire	Description
id	int	O	Id de la session.
Day	String	O	Jour du type de la session.
Time	String	O	Heure du type de la session.

Exemple JSON

```
1 {  
2   id: 1  
3   Day: "MON"  
4   Time: "20:10"  
5 }
```

21.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {  
2   result: true  
3 }
```

Code

RegistrationControllerApi.php

22 POST : /api/register

Requête PUT permettant l'enregistrement d'un nouvelle utilisateur dans le système.

22.1 Requête

paramètre du path

Pas de Paramètre pour le path.

Content type

application/json

Body

Propriété	Type	Obligatoire	Description
email	String	O	Email de l'utilisateur.
username	String	O	Nom d'utilisateur.
lastName	String	O	Nom de l'utilisateur.
firstName	String	O	Prénom de l'utilisateur.
Abonnement	int	O	Nombre de session disponible dans son abonnement.
password	String	O	mot de passe de l'utilisateur.
passwordConfirmation	String	O	vérification du mot de passe de l'utilisateur.
typeSession	List<TypeSession>	O	Liste des type de session auquel l'utilisateur s'est inscrit.
id	int	O	Id du type de session.
Day	String	O	Jour du type de session.
Time	String	O	Heure du type de session.

Exemple JSON

```
1 {
2   username : "victorsmits",
3   lastName: "Smits",
4   firstName: "Victor",
5   Email: "victor-smi@hotmail.com",
6   password: "123456",
7   passwordConfirmation : "123456",
8   typeSessions: [{
9     0: {
10      Id: 2
11      Day: "Lundi"
12      Time: "20:10"
13    }
14  }]
15 }
```

22.2 Réponse

Réussite

Retourne code 200

Content-type

application/json

Body

Propriété	Type	Obligatoire	Description
result	Boolean	O	Status de la requête.

Exemple JSON

```
1 {
2   result: true
3 }
```

Code

RegistrationControllerApi.php

5 | Séparation backend frontend

Pour la séparation backend-frontend nous avons utiliser l'api REST qui nous permet de séparer les action sur la base de donnée et l'interface de l'application web tout en permettant la transmission des donnée grâce au format JSON. L'ensemble des requête réaliser si dessous ainsi que leur explication est expliqué dans le chapitre 4.

La majorité des User Stories sont comparable et applicable au 2 architectures, cependant certaine de c'est User Stories ne sont implémenté qu'uniquement dans la partie Angular. C'est User Stories sont expliqués et illustrer ci-dessous.

1 User Stories compatible Angular uniquement

C'est User Stories on été ajouter part après la création du twig, soit pour faciliter l'exécution d'élément, soit dans le but d'ajouter de nouvelle fonctionnalité.

1.1 En tant qu'administrateur, je dois pouvoir créer un type de session

1. L'administrateur se connecte au site avec ses identifiant.
2. Un nouveau bouton de navigation apparait dans la bar de navigation.
3. L'administrateur clique sur le bouton *Admin*
4. Il sélectionne *Type Session* dans le menu déroulant.
5. L'administrateur atterris sur la page de gestion des type de sessions.
6. Il clique sur *Ajout d'une session +*
7. Un formulaire apparait et il le remplis avec les bonne information (jour - heure [hh:mm])
8. Il clique sur *ok*



Figure 5.1.1: Bouton de navigation de l'administrateur

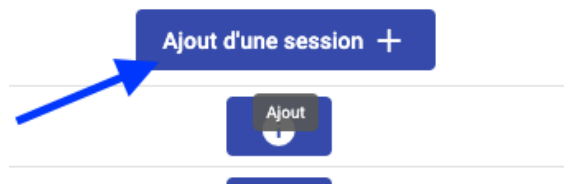


Figure 5.1.2: Bouton d'ajout d'un type de session

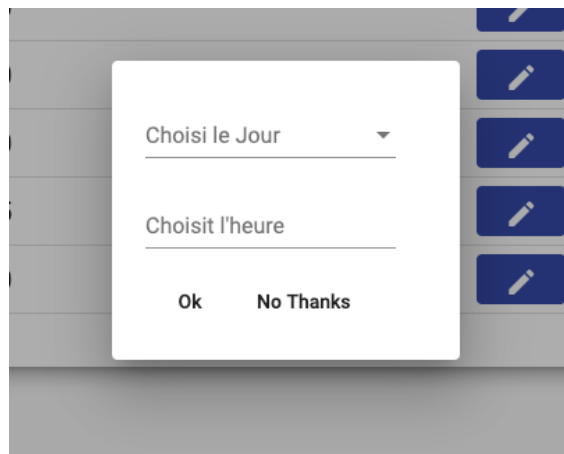


Figure 5.1.3: Page d'ajout du type de session

Gestion des erreurs

- 2 types de session identique ne peuvent exister, lors de la création d'un type de session une vérification se fait et si le type de session existe déjà une erreur apparaît.
- Si une erreur apparaît durant la création du type des sessions, elle s'affichera au dessus du formulaire.

Diagramme de séquence

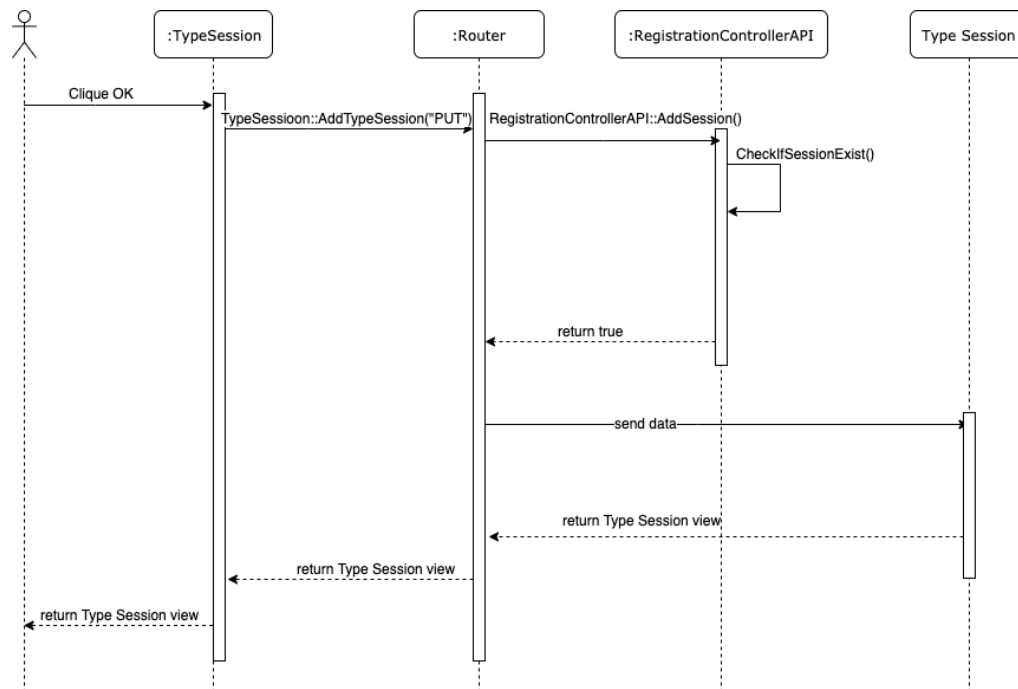


Figure 5.1.4: Diagramme de séquence de l'ajout d'un type de session

Script concerné

- RegistrationControllerApi.php
- TypeSession.php
- add-type-session.component.ts
- add-type-session.component.html

1.2 En tant qu'administrateur, je veux pouvoir modifier un type de session

1. L'administrateur se connecte au site avec ses identifiant.
2. Un nouveau bouton de navigation apparait dans la bar de navigation.
3. L'administrateur clique sur le bouton *Admin*
4. Il sélectionne *Type Session* dans le menu déroulant.
5. L'administrateur atterris sur la page de gestion des type de sessions.
6. Il clique sur le bouton edit
7. Un formulaire apparait et il le remplis avec les bonne information (jour - heure [hh:mm])
8. Il clique sur *ok*

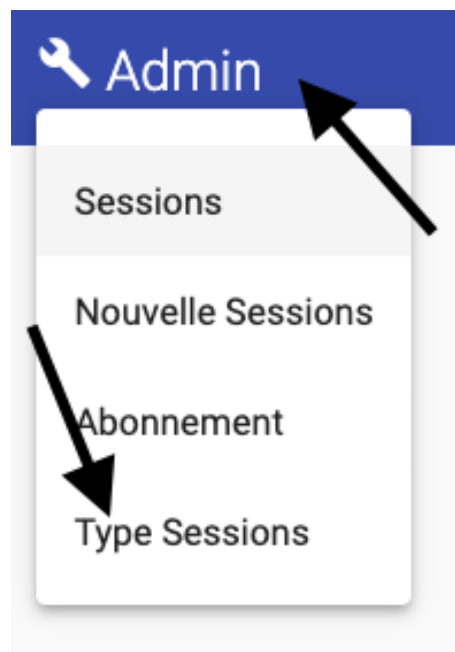
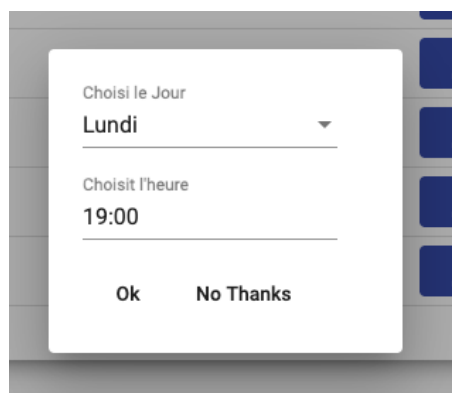


Figure 5.1.5: Menu de l'administrateur

Jour	Heure	Modification / Suppression	Ajout d'une session +
Lundi	19:00	 	
Lundi	20:10	 	

Figure 5.1.6: Bouton de modification du type de session



A modal dialog box with a white background and a grey border. It contains two sections: 'Choisi le Jour' with a dropdown menu showing 'Lundi' and a small downward arrow, and 'Choisit l'heure' with a text input field showing '19:00'. At the bottom, there are two buttons: 'Ok' and 'No Thanks'.

Figure 5.1.7: Formulaire de modification du type de session

Gestion des erreurs

Chaque type de sessions est unique, si l'administrateur rentre des informations correspondant à un type de sessions déjà existant, une erreur apparaîtra au dessus du formulaire.

Script concerné

- RegistrationControllerApi.php
- TypeSession.php
- edit-type-session.component.ts
- edit-type-session.component.html

1.3 En tant qu'administrateur, je veux pouvoir supprimer un type de session

1. L'administrateur se connecte au site avec ses identifiant.
2. Un nouveau bouton de navigation apparait dans la bar de navigation.
3. L'administrateur clique sur le bouton *Admin*
4. Il sélectionne *Type Session* dans le menu déroulant.
5. L'administrateur atterris sur la page de gestion des type de sessions.
6. Il clique sur le bouton supprimer
7. Une fenêtre de confirmation apparait.
8. Il clique sur *OUI!*

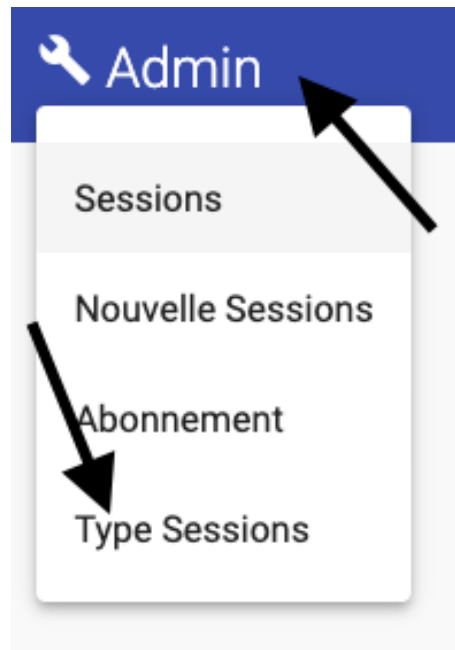


Figure 5.1.8: Menu de l'administrateur







Jour	Heure	Modification / Suppression	Ajout d'une session +
Lundi	19:00	 	
Lundi	20:10	 	

Figure 5.1.9: Bouton de suppression du type de session

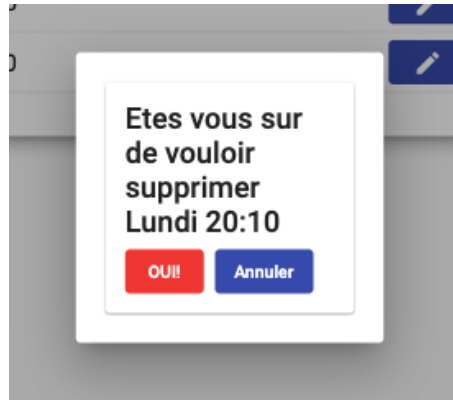


Figure 5.1.10: Fenêtre de confirmation de suppression

Script concerné

- RegistrationControllerApi.php
- TypeSession.php
- del-type-session.component.ts
- del-type-session.component.html

1.4 En tant qu'administrateur, je veux pouvoir supprimer un utilisateur du système

1. L'administrateur se connecte au site avec ses identifiant.
2. Un nouveau bouton de navigation apparait dans la bar de navigation.
3. L'administrateur clique sur le bouton *Admin*
4. Il sélectionne *Abonnement* dans le menu déroulant.
5. L'administrateur atterris sur la page de gestion des type des utilisateurs.
6. Il clique sur le bouton supprimer
7. Une fenêtre de confirmation apparait.
8. Il clique sur *OUI!*

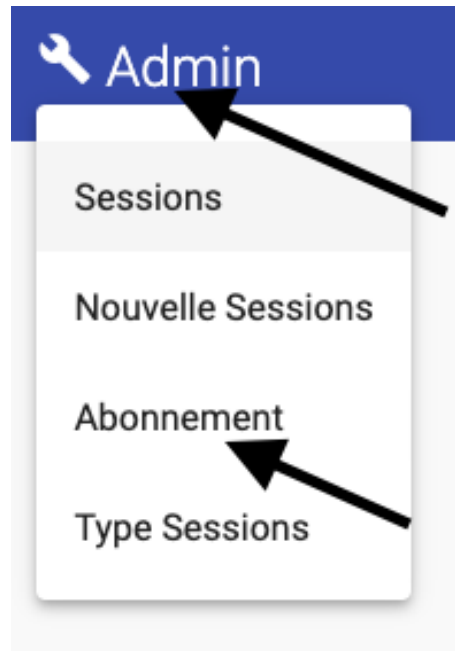


Figure 5.1.11: Menu de l'administrateur

Nom	Prenom	Abonnement	Type Abonnement	
Smits	Victor	0	0	  
Isabelle	Smits	0	20	  

Figure 5.1.12: Bouton de suppression de l'utilisateur



Figure 5.1.13: Fenêtre de confirmation de suppression

Script concerné

- AbonnementControllerApi.php
- Person.php
- del-abo.component.ts
- del-abo.component.html

1.5 En tant qu'administrateur, je dois pouvoir générer les sessions pour un nombre d'année

1. L'administrateur se connecte au site avec ses identifiant.
2. Un nouveau bouton de navigation apparait dans la bar de navigation.
3. L'administrateur clique sur le bouton *Admin*
4. Il sélectionne *Nouvelle sessions* dans le menu déroulant.
5. L'administrateur atterris sur la page de création de session.
6. Il clique sur *Génération auto*
7. Un formulaire lui demandant sur combien d'année ainsi que quelle type de session il souhaite générer apparait.
8. Il sélectionne les types de sessions qu'il souhaite générer.
9. Il indique le nombre d'année voulus
10. Il clique sur *Je confirme la génération*

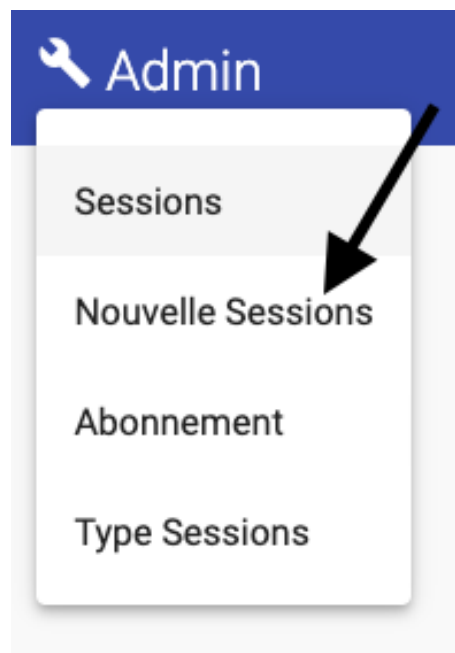


Figure 5.1.14: Menu de l'administrateur

Choose a date *

Pick a time *

Number of bike *

Add

Generation auto

Figure 5.1.15: Bouton de génération automatique

Génération automatique de sessions

- ☐ Lundi 20:10
- ☐ Mercredi 09:00
- ☐ Mercredi 10:10
- ☐ Jeudi 17:30
- ☐ Jeudi 18:45
- ☐ Jeudi 19:50
- ☐ Lundi 19:00
- ☐ Lundi 10:00

nombre d'année

je confirme la génération

annuler

Figure 5.1.16: Formulaire pour le génération automatique de session

Gestion des erreurs

- La génération automatique suppose que tout les types de sessions on été préalablement configurer si ce n'est pas le cas une erreur apparaitra à la confirmation.
- Si une erreur se créer durant la génération elle est interrompus et un message d'erreur est afficher à l'administrateur.

Diagramme de séquence

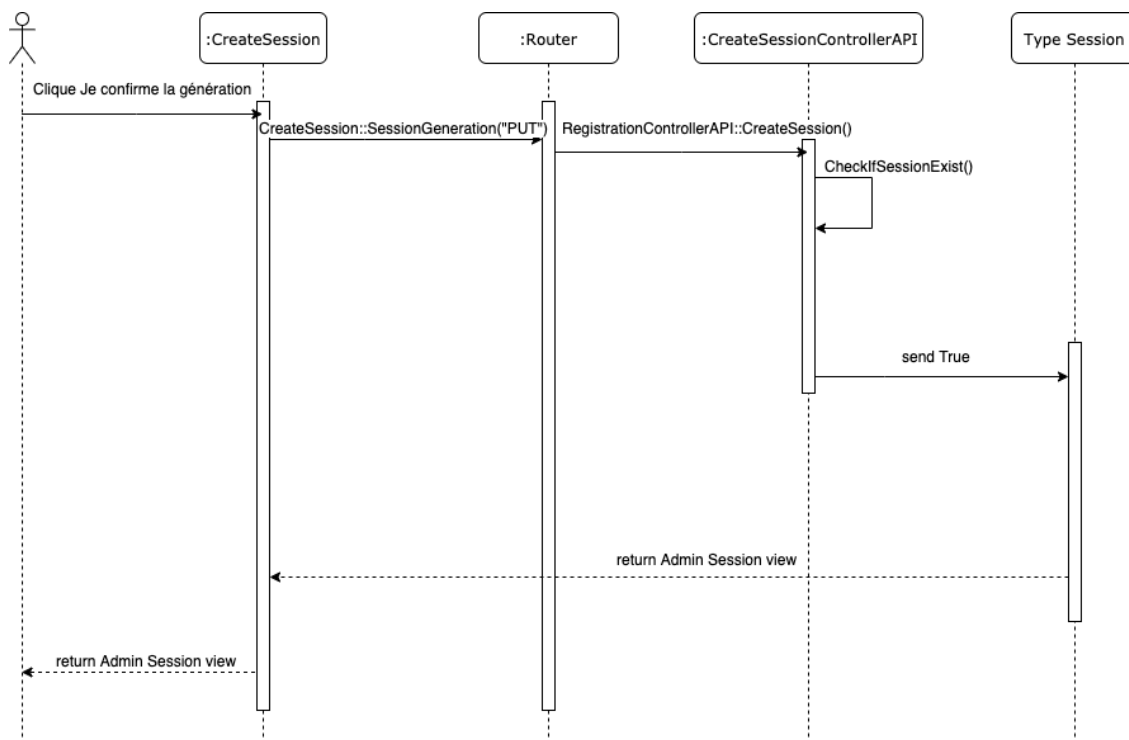


Figure 5.1.17: Diagramme de séquence de la génération automatique de session

Script concernés

- Session.php
- CreateSessionControllerApi.php
- admin-create-session.component.ts
- admin-create-session.component.html

2 User Stories compatible Symfony-Angular

2.1 En tant qu'utilisateur, je dois pouvoir me connecter sur le site

Différence

- Réalisation d'une requête POST via l'api REST permettant d'effectuer l'authentification dans le backend.
- Différence au niveau de l'aspect graphique de la page de connection.
- Connection via l'adresse Email et non le nom d'utilisateur.
- Redirection automatique de l'utilisateur vers la page de connection l'ors de l'accès au site si l'utilisateur n'est pas connecter.
- Utilisation d'un LoginAuthenticator afin de pouvoir authentifier l'utilisateur dans le backend.
- Utilisation de cookie pour préserver les informations de l'utilisateur.

Diagramme de séquence

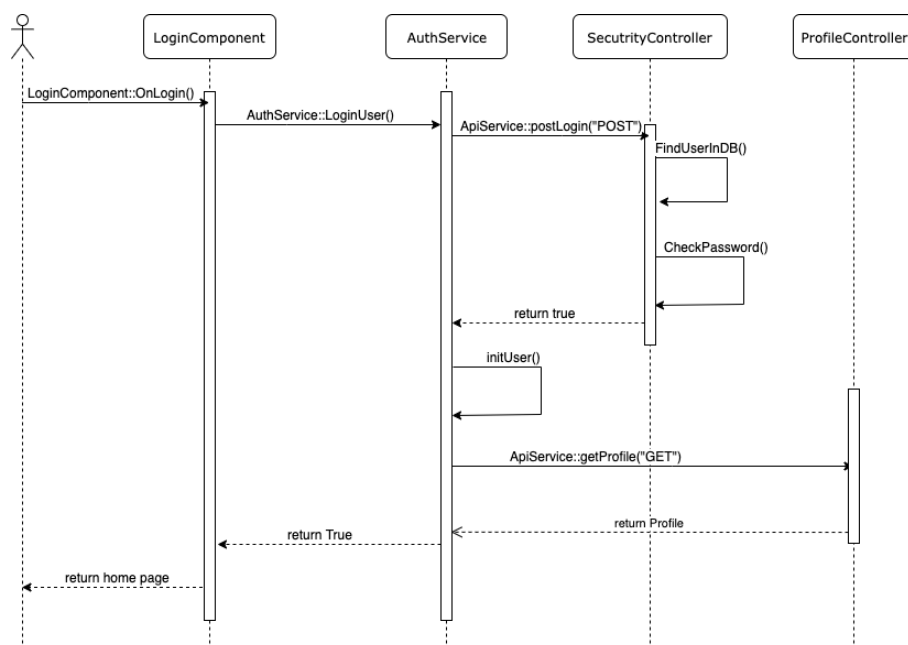


Figure 5.2.18: Diagramme de séquence de la connections d'un utilisateur

Script concernés

- api.service.ts
- auth.service.ts
- SecurityControllerAPI.php
- ProfileControllerAPI.php
- login.component.ts
- login.component.html
- Person.php

2.2 En tant qu'utilisateur, je dois pouvoir me créer un compte sur le site

Différence

- Réalisation d'une requête PUT via l'api REST.
- Amélioration de l'aspect graphique.
- Utilisation du *SecurityAuthenticator* du backend pour ajouter l'utilisateur dans le système.

Diagramme de séquence

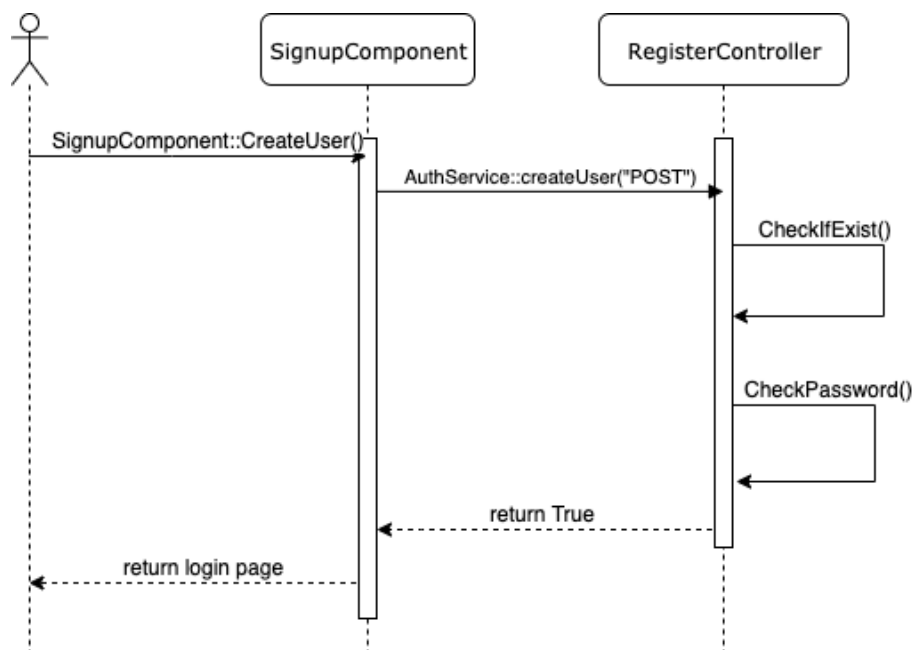


Figure 5.2.19: Diagramme de séquence de l'inscription d'un utilisateur.

Script concernés

- `auth.service.ts`
- `SecurityControllerAPI.php`
- `Person.php`
- `signup.component.ts`
- `signup.component.html`

2.3 En tant qu'utilisateur, je dois pouvoir sélectionner le mois et l'année pour afficher les sessions qui m'intéresse

Différence

- Réorganisation de l'affichage.
- Ajout de la possibilité de sélectionner l'année à afficher.

Script concernés

- api.service.ts
- MonthControllerAPI.php
- Session.php
- month.component.ts
- month.component.html

2.4 En tant qu'utilisateur, je dois pouvoir m'inscrire à une session un certain jour

Différence

- Modification de l'aspect des boutons d'inscription.

Script concernés

- api.service.ts
- MonthControllerAPI.php
- Session.php
- month.component.ts
- month.component.html

2.5 En tant qu'utilisateur, je dois pouvoir être inscrit automatiquement à une session

Différences

- Réalisation de l'inscription automatique au session lors de l'enregistrement de l'utilisateurs.
- Utilisation de type de session afin de facilité et de multiplier le choix des abonnements possible.
- Possibilité pour un utilisateur de s'inscrire à plus de un type de session par semaine.
- Sélection du type de session parmi une liste.

E-Mail *

Username *

Nom *

Prenom *

Abonnement *

☐ Lundi 20:10

☐ Mercredi 09:00

☐ Mercredi 10:10

☐ Jeudi 17:30

☐ Jeudi 18:45

☐ Jeudi 19:50

☐ Lundi 19:00

☐ Lundi 10:00

Password *

Repeat Password *

S'inscrire

* : champs requis

Figure 5.2.20: Formulaire d'inscription version angular

Diagramme de séquence

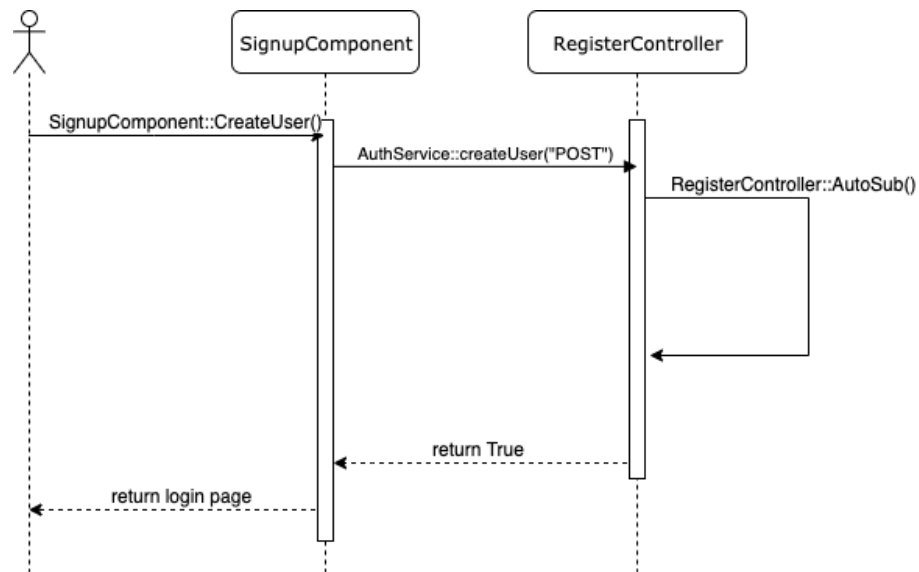


Figure 5.2.21: Diagramme de séquence de l'inscription automatique d'un utilisateur.

Script concernés

- auth.service.ts
- SecurityControllerAPI.php
- TypeSession.php
- Session.php
- Inscription.php
- Person.php
- signup.component.ts
- signup.component.html

2.6 En tant qu'utilisateur, je dois pouvoir me désinscrire à une session

Différence

- Modification du boutons de désinscription à un session.

Script concernés

- api.service.ts
- MonthControllerAPI.php
- Session.php
- month.component.ts
- month.component.html

2.7 En tant qu'utilisateur, je dois pouvoir voir qui est inscrit pour chaque session

Différence

- Affichage de la liste des participant(e)s ne ce fait plus sur une page séparé mais sur un fenêtre de dialogue qui apparait devant le tableau.
- Modification du bouton permettant l'affichage des participant(e)s.

Script concernés

- api.service.ts
- MonthControllerAPI.php
- Session.php
- month.component.ts
- month.component.html

2.8 En tant qu'utilisateur, je veux pouvoir voir dynamiquement le nombre de séance qui reste dans mon abonnement

Différence

- Modification graphique de l'affichage du profil et des informations.

Script concernés

- api.service.ts
- ProfileControllerAPI.php
- Person.php
- profile.component.ts
- profile.component.html

2.9 En tant qu'utilisateur, je veux pouvoir modifier mon profil

Différence

- Modification graphique du formulaire.
- Possibilité de s'abonner à plusieurs types de sessions différentes.

Email
victor-smi@hotmail.com

Nom
Smits

Prenom
Victor

Mot de passe

☒ Lundi 20:10

☐ Mercredi 09:00

☐ Mercredi 10:10

☐ Jeudi 17:30

☐ Jeudi 18:45

☐ Jeudi 19:50

☐ Lundi 19:00

☐ Lundi 10:00

No Thanks Ok

Figure 5.2.22: Formulaire de modification du profil de l'utilisateur

Script concernés

- ProfileController.php
- profile.html.twig
- Person.php

2.10 En tant qu'administrateur, je dois pouvoir annuler une session

Différence

- Modification graphique du bouton d'annulation de la session.

Script concernés

- api.service.ts
- SessionAdministrationControllerApi.php
- Session.php
- admin-session.component.ts
- admin-session.component.html

2.11 En tant qu'administrateur, je dois pouvoir créer une nouvelle session peut importe la date

Différence

- Modification graphique du formulaire de création de session.

Script concernés

- api.service.ts
- CreateSessionControllerApi.php
- Session.php
- admin-create-session.component.ts
- admin-create-session.component.html

2.12 En tant qu'administrateur, je dois pouvoir gérer les abonnements des utilisateurs

Différence

- Modification graphique de l'affichage des utilisateurs enregistré dans le système.

Script concernés

- api.service.ts
- AbonnementControllerApi.php
- Person.php
- admin-abo.component.ts
- admin-abo.component.html

6 | Conclusion

1 Objectif atteints et non atteints

1.1 Compatible Symfony-Twig et Symfony-Angular

User Stories	Priorité	Réalisation	Reste à faire
En tant qu'utilisateur, je dois pouvoir me créer un compte sur le site	2	100%	
En tant qu'utilisateur, je dois pouvoir sélectionner le mois et l'année pour afficher les sessions qui m'intéresse	2	80%	Sélection de l'année dans la partie symfony-twig et affichage des message d'erreur sur le site Angular.
En tant qu'utilisateur, je dois pouvoir m'inscrire à une session un certain jour	1	90%	Afficher le message d'erreur sur le site Angular.
En tant qu'utilisateur, je dois pouvoir être inscrit automatiquement à une session	1	80%	Effectuer la boucle d'inscription pour la partie Symfony-Twig.
En tant qu'utilisateur, je dois pouvoir me désinscrire à une session	1	90%	Afficher le message d'erreur sur le site Angular.
En tant qu'utilisateur, je dois pouvoir voir qui est inscrit pour chaque session	2	100%	

En tant qu'utilisateur, je veux pouvoir voir dynamiquement le nombre de séance qui reste dans mon abonnement	2	100%	
En tant qu'utilisateur, je veux pouvoir modifier mon profile	1	100%	
En tant qu'utilisateur, je veux pouvoir m'inscrire a une session uniquement si il me reste des abonnements a disposition.	2	100%	
En tant qu'administrateur, je dois pouvoir annuler une session	2	90%	Afficher les messages d'erreur sur le site Angular.
En tant qu'administrateur, je dois pouvoir gérer les abonnement des utilisateurs	2	100%	
En tant qu'administrateur, je dois pouvoir créer une nouvelle session peut importe la date	2	100%	

1.2 Compatible Symfony-Angular uniquement

En tant qu'administrateur, je dois pouvoir générer les sessions pour un nombre d'année	1	100%	
En tant qu'administrateur, je dois pouvoir créer un type de session	2	100%	
En tant qu'administrateur, je veux pouvoir gérer les types de sessions disponible pour l'inscription automatique	2	100%	
En tant qu'administrateur, je veux pouvoir modifier un type de session	1	100%	
En tant qu'administrateur, je veux pouvoir supprimer un type de session	1	90%	Afficher le message d'erreur sur le site Angular.
En tant qu'administrateur, je veux pouvoir supprimer un utilisateur du système	1	90%	Afficher le message d'erreur sur le site Angular.

2 Piste d'amélioration

- Optimisation des requêtes de base de donnée.
- Nettoyage du code source (duplication, réorganisation, ...).
- Amélioration/Finissions de la responsivité du site.
- Triage des sessions reçus dans angular pour un affichage trié en fonction du jour et de l'heure.
- Renommage des clés des json envoyer au frontend.
- Renommage de certain champs de la base de donnée.
- Utilisation d'une base de donnée orienté graph.

- Lors de la creation d'une session, trouver un type de session compatible si existant sinon mettre le paramètre à NULL.

7 | Annexe

1 Code source

Retrouvez l'entièreté du code source sur le Github suivant : [Aquabike](#)