



## Documentation

### Fonctionnelle / Technique

-

Marius THIESSET



## **DOSSIER FONCTIONNELLE**

- I- Données :**
  - a. Dictionnaire de données
  - b. Modèle Entité Association
  - c. Schéma Relationnel
  - d. Script d'implémentation
  
- II- Fonctionnalités :**
  - a. Liste des fonctionnalités
  - b. Arborescence
  - c. Cas d'utilisations
  - d. Maquettes

## **DOSSIER TECHNIQUE**

- I- Présentation de deux fonctionnalités**
  - a. Question
  - b. Modification du profil
  
- II- Veille et auto-formation**
  
- III- Conduite de projet**
  
- IV- Compte rendu de réunion**



## Introduction

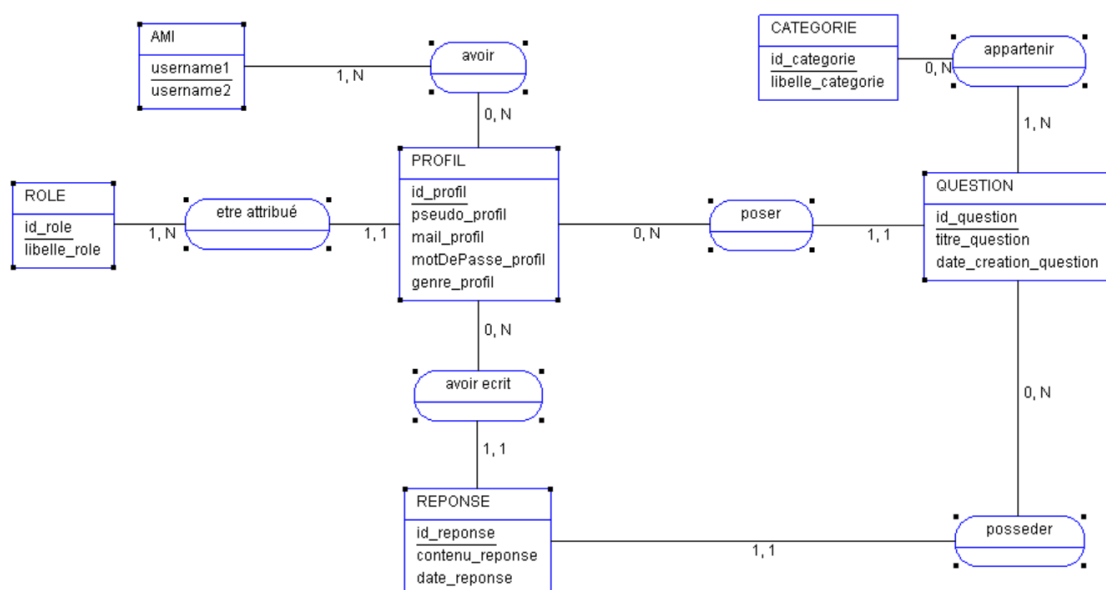
LiveQuestion est une plateforme qui permet aux utilisateurs inscrits de se poser des questions entre eux et d'y répondre. D'autres fonctionnalités ont été ajoutées, telles que la modification de son profil, ou bien, l'ajout d'amis. C'est un projet réalisé en 1ère année de BTS dans le cadre d'une « semaine piscine », sous HTML, CSS, Bootstrap et PHP.

## Données

### Dictionnaire de données

Table	Attribut	Type	Description
<b>profil</b>	id_profil	int	
	pseudo_profil	string	
	mail_profil	string	
	MotDePasse_profil	string	
	Genre_profil	string	Femme / Homme
<b>question</b>	id_question	int	
	titre_question	string	
	date_creation_question	date	
<b>categorie</b>	id_categorie	int	
	libelle_categorie	string	Exemple : Cinema, animaux..
<b>ami</b>	username 1	string	Pseudo ami 1
	username 2	string	Pseudo ami 2
<b>reponse</b>	id_reponse	int	
	contenu_reponse	string	
	date_reponse	date	Date de création de la réponse
<b>role</b>	id_role	int	
	libelle_role	string	Administrateur / Utilisateur

### Modèle entité association





## Schéma Relationnel

AMI (username1, username2)

Clé primaire : username1

CATEGORIE (id\_categorie, libelle\_categorie)

Clé primaire : id\_categorie

ROLE (id\_role, libelle\_role)

Clé primaire : id\_role

REPONSE(id\_reponse, contenu\_reponse, date\_reponse)

Clé primaire : id\_reponse

QUESTION(id\_question, titre\_question, date\_creation\_question)

Clé primaire : id\_question

Clé étrangère : id\_reponse en référence à id\_reponse de la table REPONSE

PROFIL(id\_profil, pseudo\_profil, mail\_profil, motDePasse\_profil, genre\_profil, id\_role, id\_reponse, id\_question)

Clé primaire : id\_profil

Clé étrangère : id\_role en référence à id\_role de la table ROLE

id\_reponse en référence à id\_reponse de la table REPONSE

id\_question en référence à id\_question de la table QUESTION

AVOIR (username1, id\_profil)

Clé primaire : username1 et id\_profil

Clé étrangère : username 1 en référence à username1 de la table Ami

id\_profil en référence à id\_profil de la table PROFIL

APPARTENIR (id\_categorie, id\_question)

Clé primaire : id\_categorie, id\_question

Clé étrangère : id\_categorie en référence à id\_categorie de la table CATEGORIE

id\_question en référence à id\_question de la table QUESTION



## Script d'implémentation

→ Table CATEGORIE

```
CREATE TABLE `categorie` (  
  `Id_categorie` int(11) NOT NULL,  
  `Libelle_categorie` varchar(255) CHARACTER SET latin1 NOT NULL  
)
```

→ Table AMI

```
CREATE TABLE `ami` (  
  `username_1` int(11) NOT NULL,  
  `username_2` int(11) NOT NULL,  
  `is_pending` tinyint(1) NOT NULL DEFAULT '0'  
)
```

→ Table PROFIL

```
CREATE TABLE `profil` (  
  `Id_profil` int(11) NOT NULL,  
  `Pseudo_profil` varchar(255) CHARACTER SET latin1 NOT NULL,  
  `Mail_profil` varchar(255) CHARACTER SET latin1 NOT NULL,  
  `MotDePasse_profil` varchar(255) CHARACTER SET latin1 NOT NULL,  
  `Genre_profil` varchar(50) CHARACTER SET latin1 NOT NULL,  
  `#Id_role` int(11) NOT NULL DEFAULT '1'  
)
```

→ Table QUESTION

```
CREATE TABLE `question` (  
  `Id_question` int(11) NOT NULL,  
  `Titre_question` varchar(255) CHARACTER SET latin1 NOT NULL,  
  `Date_creation_question` date NOT NULL,  
  `#Id_profil` int(11) NOT NULL,  
  `#Id_categorie` int(11) NOT NULL,  
  `type_question` int(11) NOT NULL DEFAULT '0'  
)
```

→ Table REPONSE

```
CREATE TABLE `reponse` (  
  `Id_reponse` int(11) NOT NULL,  
  `Contenu_reponse` varchar(255) CHARACTER SET latin1 NOT NULL,  
  `Date_reponse` date NOT NULL,  
  `#Id_profil` int(11) NOT NULL,  
  `#Id_question` int(11) NOT NULL  
)
```

→ Table ROLE

```
CREATE TABLE `role` (  
  `Id_role` int(11) NOT NULL,  
  `Libelle_role` varchar(50) CHARACTER SET latin1 NOT NULL  
)
```



## Fonctionnalités

### Liste des fonctionnalités

- Formulaire d'inscription

- Formulaire de connexion

- Page de profil :

Possibilité de modifier son mot de passe, son pseudo, et son email.

Visibilité des questions déjà posées.

- Page du profil d'un autre utilisateur :

Possibilité de voir ses informations telles que le pseudo, le genre, et les questions posées.

- Ajout ami :

Possibilité d'ajouter un ami, d'accepter les demandes, et d'en envoyer.

- Ajout question :

L'utilisateur peut poser une question à la communauté de LiveQuestion ou bien, décider de la poser uniquement à ses amis.

- Ajout réponse :

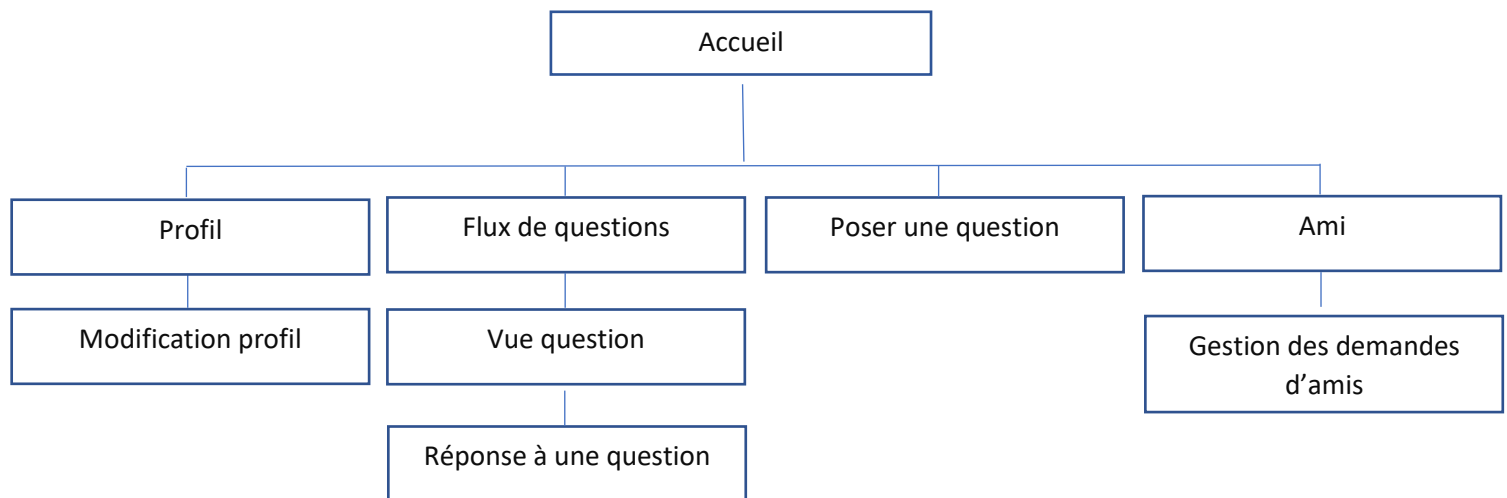
Possibilité de répondre à une question posée par un autre utilisateur.

- Backoffice :

Accessible uniquement par les modérateurs : permet de modifier/ supprimer / ajouter une catégorie et supprimer une question. (Les réponses associées à la question supprimées seront aussi supprimées.)

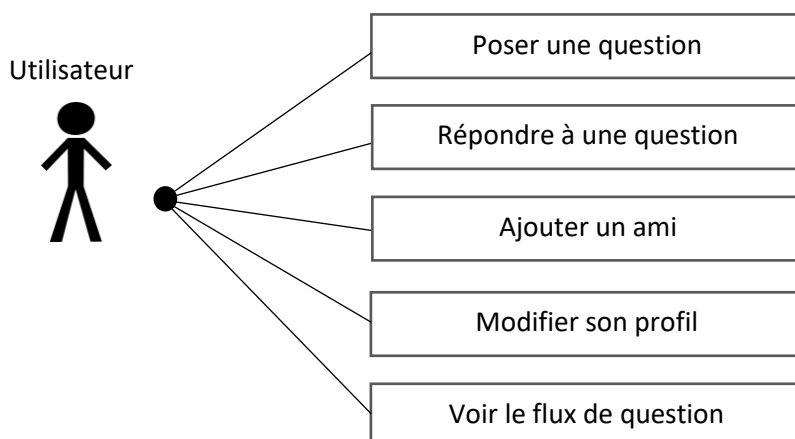


## Arborescence



## Cas d'utilisations

	ID	Fonctionnalité	User story	Priorité	Ressource	Statut	Pré-Prod	Prod
	1	Base de donnée	Création de la table profil	1	Hugo , Marius, Elisa	Validé	Testé	Testé
	2		Création de la table question	1		Validé	Testé	Testé
	3		Création de la table Réponse	1		Validé	Testé	Testé
	4		Création de la table rôle	1		Validé	Testé	Testé
	5		Création de la table ami			Validé	Testé	Testé
	6	Connexion	Création de la table catégorie	1	Hugo MONTEIRO	Validé	Testé	Testé
	7		Formulaire de connexion	1		Validé	Testé	Testé
	8		Formulaire d'inscription	1		Validé	Testé	Testé
	9	Gestion des utilisateurs	Page profil personnel	2	Elisa RIDEL	Validé	Testé	Testé
	10		Page utilisateur	2		Validé	Testé	Testé
	11	Gestion des permission	Modification du profil	2	Marius THIESSET	Validé	Testé	Testé
	12		Création du back-office	3		Validé	Testé	Testé
	13		Création d'un administrateur	3		Validé	Testé	Testé
	14	Ajout de fonctionnalités	Ajout question	1	Hugo MONTEIRO	Validé	Testé	Testé
	15		Ajout réponse	2		Validé	Testé	Testé
	16	Interface	Ajout ami	3	Hugo MONTEIRO	Validé	Testé	Testé
	17		Flux de question	1	Hugo MONTEIRO	Validé	Testé	Testé
	18		Pagination	3	Elisa RIDEL	Validé	Testé	Testé
	19		Vue question	2	Hugo MONTEIRO	Validé	Testé	Testé
	20		Vue réponse	2	Marius Thiesset	Validé	Testé	Testé
	21		Page d'accueil	2	Hugo MONTEIRO	Validé	Testé	Testé





Accès au backoffice

- Précisions sur les cas d'utilisation utilisateur



Poser une question

La question peut être posée aux amis uniquement ou à tout le monde



Répondre à une question

Possibilité de répondre à une question posée par un utilisateur.



Ajouter un ami

Possibilité de rechercher une personne. Puis de lui envoyer une demande. Possibilité d'accepter ou refuser une demande d'ami.



Modifier son profil

Possibilité de modifier son pseudo, email et mot de passe



Voir le flux de questions

Vue sur toutes les questions posées par les utilisateurs. Possibilité de voir 1 question et ses réponses, de répondre à une question. De visiter le profil d'un utilisateur en cliquant sur son pseudo.

- Précisions sur les cas d'utilisation administrateur



Accès au backoffice

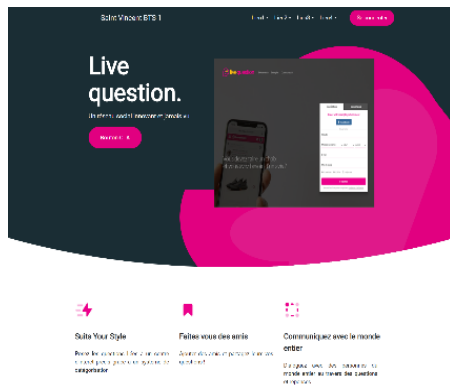
Ajouter/ Modifier / Supprimer une catégorie. Supprimer une question, et les réponses qui lui sont associées.



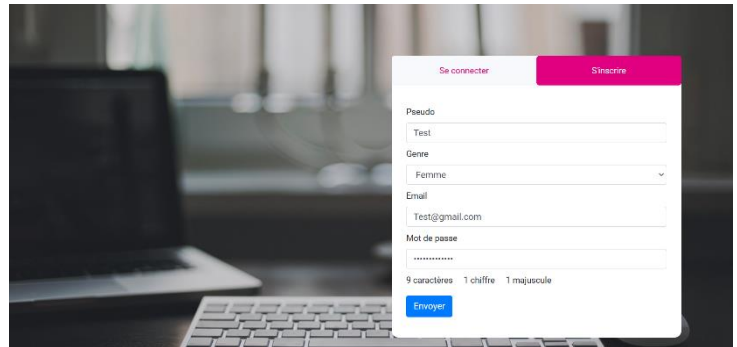


## Maquettes

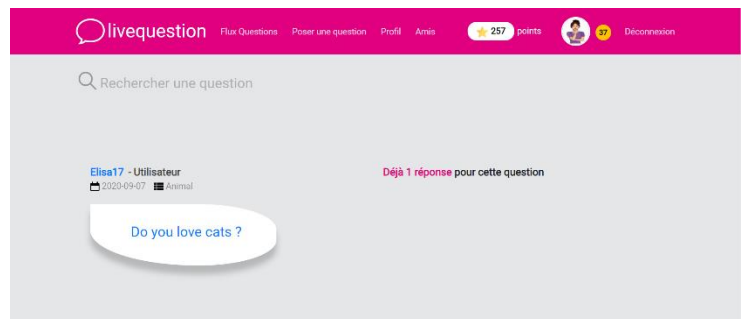
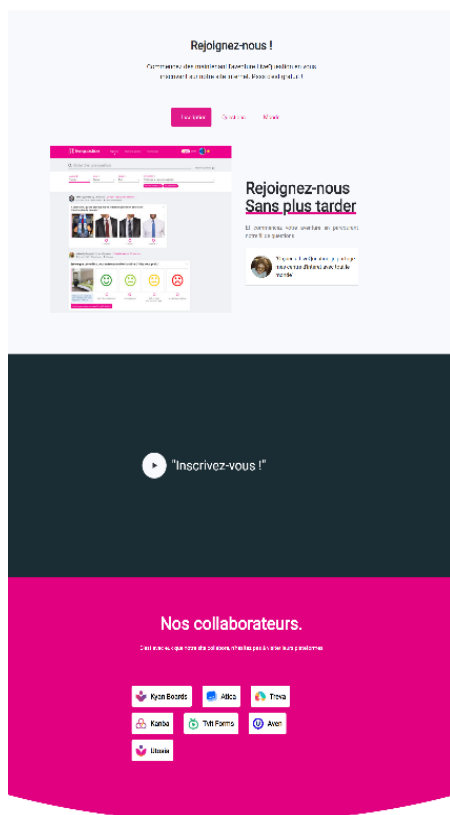
Page d'accueil → Connexion / Inscription



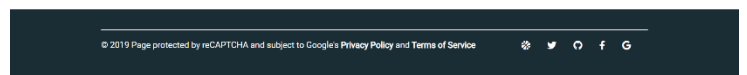
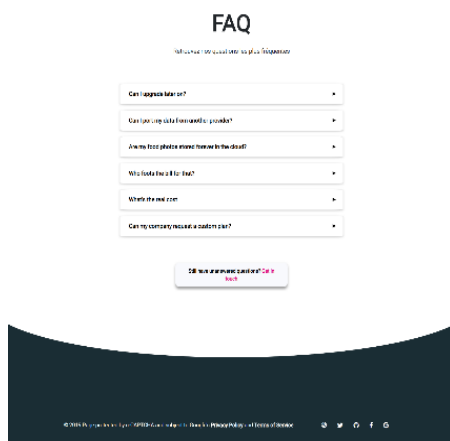
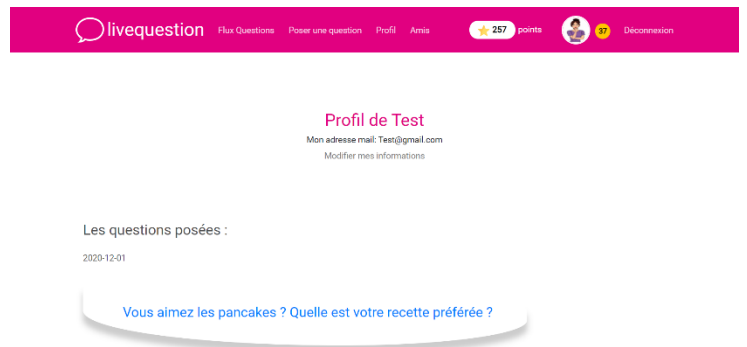
Page de connexion et d'inscription



Page d'accueil une fois connecté → Contient le flux de question



Vue d'un profil





Modification du profil → accessible depuis la page profil

Ajout d'un ami

## Dossier technique

### Focus sur deux fonctionnalités

- Ajout d'une question

```
if (!empty($_POST) AND isset($_POST['type_question'])) {
    // Envoie du formulaire question
    $query = $db-
>prepare('INSERT INTO question (`Titre_question`, `Date_creation_question`, `#
Id_profil`, `#Id_categorie`, `type_question`) VALUES (:Titre_question, :Date_c
reation_question, :Id_profil, :Id_categorie, :type_question)');

```

On vérifie si le formulaire n'est pas vide.

Si ce n'est pas le cas, on envoie une requête qui va permettre l'insertion de toutes les données rentrées dans le formulaire :

- Le titre
- La date de création de la question
- Le profil à l'origine de la question
- La catégorie choisie
- Le type de question (publique ou privée)

```
$titreQuestion = $_POST['Titre_question'];
$idCategorie = $_POST['Id_categorie'];
$type_question = $_POST['type_question'];

$query->bindParam(':Titre_question', $titreQuestion);
$query-
>bindParam(':Date_creation_question', $dateCreationQuestion);
$query->bindParam(':Id_profil', $idProfil);
$query->bindParam(':Id_categorie', $idCategorie);
$query->bindParam(':type_question', $type_question);

```

On initialise des variables qui vont contenir les valeurs du formulaire.



```
date_default_timezone_set('Europe/Paris');
$dateCreationQuestion = date('Y-m-d');
$idProfil = $_SESSION['id'];
```

On initialise la date et le profil de la session à l'aide de ces lignes car elles ne sont pas liées au formulaire.

```
$query->execute();
$query->closeCursor();
$last = $db->lastInsertId();
```

Puis on exécute afin de tous les envoyer en base de données grâce à la requête précédente.

- Modification du profil

```
$reqprofil = connexionBdd()-
>prepare("SELECT * FROM profil WHERE Id_profil = ?");
$reqprofil->execute(array($_SESSION['id']));
$profil = $reqprofil->fetch();
```

On fait une requête qui récupère toutes les données de l'utilisateur qui souhaite modifier son profil

```
if(isset($_POST['newPseudo']))
{
    $newPseudo = $_POST['newPseudo'];
    $reqPseudo = connexionBdd()-
>prepare("SELECT * FROM profil WHERE Pseudo_profil = ?");
    $reqPseudo->execute(array($newPseudo));
    $PseudoExist = $reqPseudo->fetch();
```

Lorsque le formulaire est rempli, on stock le nouveau pseudo dans la variable \$newPseudo.

```
if($PseudoExist == false)
{
    if(isset($_POST['newPseudo']) AND !empty($_POST['newPseudo']) AND $_POST['newPseudo'] != $profil['Pseudo_profil'])
    {
        $newPseudo = htmlspecialchars($_POST['newPseudo']);
        $insertPseudo = connexionBdd()-
>prepare("UPDATE profil SET Pseudo_profil = ? WHERE Id_profil = ? ");
        $insertPseudo->execute(array($newPseudo, $_SESSION['id']));
        header('Location: profil.php?id='.$_SESSION['id']);
    }
}
else{
    $erreurPseudo = "Ce pseudo existe déjà";
}
```

Puis on vérifie si le nouveau pseudo n'est pas déjà utilisé par quelqu'un d'autre on vérifie ensuite qu'il est différent de l'ancien. Si ces conditions sont respectées, on peut remplacer l'ancien pseudo dans la base de données par le nouveau. Si ce n'est pas le cas, un message d'erreur affiche « Ce pseudo existe déjà ».



## Veille et autoformation

Approfondissement des connaissances déjà acquises en PHP à l'aide de veilles réalisées sur internet. Notamment pour envoyer des informations en base de données.

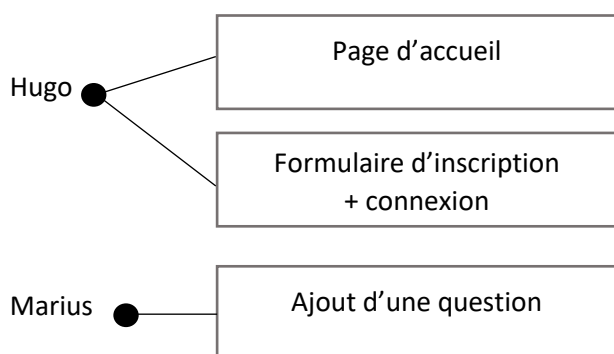
Veille sur les requêtes SQL.

## Conduite de projet

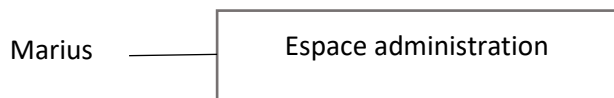
Nous étions trois à travailler sur le projet : Elisa, Hugo et moi-même, Marius.  
Voici notre répartition des tâches :

- L'ensemble du groupe a réalisé : le dictionnaire de données, le MEA ainsi que la base de données et les règles de gestions.

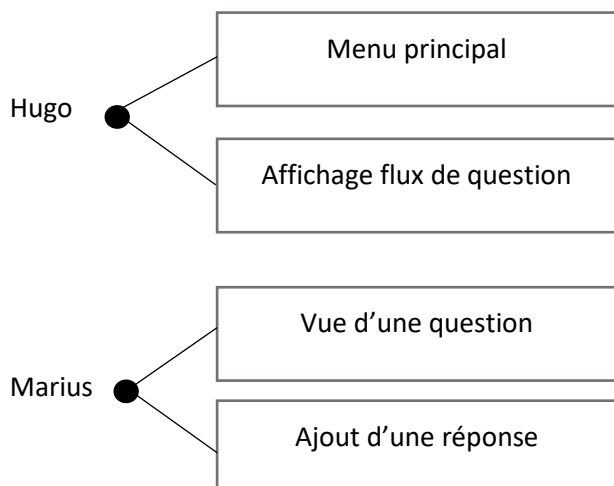
### - 17 - 29 février

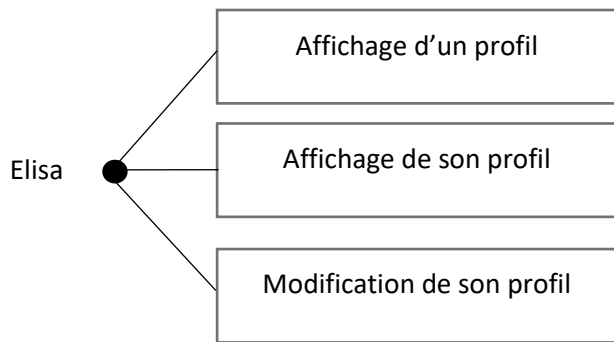


### - 29 février - 15 mars

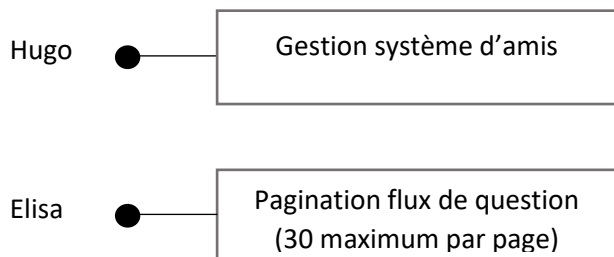


### - 15 mars - 5 avril





#### - Ajouts de fonctionnalités



### Compte rendu de réunion

#### LiveQuestion

- Participants : Elisa RIDEL, Hugo MONTEIRO et Marius THIESSET.

#### Rapport

- Gestion des délais : Respectés.
- Gestion des ressources humaines et compétences : Travail collaboratif, bonne répartition des tâches.  
→ Départ d'un membre qui a occasionné du retard.
- Pilotage du projet : Gestion global maitrisée.
- Gestion de la communication : Globalement correcte.  
→ Problèmes de communication sur la répartition des tâches au début.

#### Bilan

Avec le respect des répartitions des tâches et des délais, le projet à pu être fini dans les temps impartis et comportait tous les éléments demandés.