

Lukas BLANCHOT
Hugo PAGE
Camille RADE

PROJET CUBE - LIVRABLE 2
“(RE)Sources Relationnelles”
GROUPE C

SOMMAIRE

1. Rappel du projet et les objectifs

2. Choix technologiques réalisés

3. Méthodologie de travail

4. Cahier de recette

4.1 Stratégie globale de test

4.2 Mise en œuvre des tests

4.4 Macro-Planning

4.5 Scénario de test

5. Architecture de l'application WEB

5.1 Fonctionnement global de l'application Web

5.2 Partie Front de l'application WEB

6. Architecture de l'application mobile

7. Budget

8. Conclusion

Annexes

1. Rappel projet et les objectifs

Le ministère des Solidarités et de la Santé souhaite mettre en place un réseau social, permettant la création et le partage de ressources entre utilisateurs, nommé << **(Re)Sources Relationnelles** >>. Ce réseau social et ses ressources doivent être modérés pour éviter des débordements, et sécurisés afin de protéger les données sensibles et de coller aux normes en vigueur (RGPD, CNIL).

L'objectif principal de (RE)Sources Relationnelles est de proposer une plateforme autour de la qualité des liens relationnels que nous pouvons tisser pour une meilleure qualité de vie. L'application doit permettre aux citoyens connectés de créer, modifier, commenter, partager des ressources (textes, images, vidéos, articles, etc...). Elle se doit d'être sécurisée par le biais de l'administration des comptes, la sécurisation des données et la modération des échanges. Mais elle doit également permettre l'édition de statistiques en rapport aux ressources à disposition (consultations, recherches, partages).

2. Choix technologiques réalisés

Pour l'application Web :

Symfony

Symfony est un Framework open source de développement web pour PHP. Voici ses points forts :

- Sécurité : Symfony offre de nombreuses fonctionnalités de sécurité, telles que l'authentification et l'autorisation, la protection contre les failles de sécurité courantes, etc.
- Modularité : Symfony est très modulaire et permet de composer des applications web à partir de bundles (modules) réutilisables.
- Performances élevées : Symfony est connu pour ses performances élevées, notamment grâce à l'utilisation de cache et d'autres techniques d'optimisation.
- Facilité de maintenance : Symfony est facile à maintenir grâce à sa structure modulaire et à sa documentation complète.

Bootstrap

- Responsive : Bootstrap est conçu pour être facilement adaptatif à toutes les tailles d'écran, ce qui signifie que vos pages Web seront facilement accessibles à partir de toutes les tailles d'appareils, de la plus petite des téléphones portables aux écrans d'ordinateur de bureau.

- Composants prêts à l'emploi : Bootstrap fournit une grande variété de composants prêts à l'emploi tels que des boutons, des formulaires, des menus déroulants, des carrousels, des modales, etc. Ces composants peuvent être utilisés directement dans votre code, ce qui permet de gagner du temps et de l'effort.
- Flexibilité : Bootstrap est conçu pour être facilement personnalisable. Vous pouvez modifier les variables Sass pour changer la couleur, la police, la taille, etc. Vous pouvez également créer vos propres styles personnalisés pour les composants.
- Documentation : Bootstrap est livré avec une documentation très complète qui est facile à comprendre et à utiliser. Elle comprend des exemples de code, des descriptions de fonctionnalités, des tutoriels, etc.
- Grande communauté : Bootstrap est une bibliothèque très populaire avec une grande communauté de développeurs. Cela signifie qu'il existe de nombreux exemples de code, des modules complémentaires tiers, des ressources de formation, etc., disponibles en ligne pour vous aider dans votre travail.
- Compatibilité : Bootstrap est compatible avec la plupart des navigateurs modernes, ce qui signifie que vos pages Web fonctionnent correctement sur la plupart des appareils et des navigateurs.

Pour l'application mobile :

Flutter

Flutter est un Framework open source de développement d'applications mobiles pour Android, iOS et le web. Voici ses points forts :

- Performances élevées : Flutter utilise son propre moteur de rendu graphique, ce qui permet d'obtenir des performances élevées et une grande fluidité d'interface utilisateur.
- Hot-reload : Flutter offre une fonctionnalité de hot-reload, qui permet de recharger instantanément l'interface utilisateur en temps réel lorsqu'une modification de code est effectuée.
- Widgets personnalisables : Flutter propose une grande variété de widgets prédéfinis, ils sont simples d'utilisation et offrent une grande possibilité de création
- Large communauté : Flutter bénéficie d'une communauté en constante croissance, ce qui se traduit par de nombreuses bibliothèques, packages et ressources disponibles pour les développeurs.

3.Méthodologie de travail

Nous nous sommes dirigés vers une méthodologie agile pour gérer notre projet, en effet, toutes les communications ainsi que les réunions ont été effectuées sur Discord. Mais également au CESI pendant nos semaines d'alternance. Le travail a été effectué en équipe de manière organisée, chacun avait des tâches prédéfinies. On a su également s'adapter et être flexible face aux difficultés que nous avons pu rencontrées. Une importance à la qualité du produit a été effectuée afin de permettre de livrer un produit fonctionnel qui correspondait à la demande du client et à ses besoins.

Répartition des tâches de notre projet :

Tâches	Mobile	Symfony
Gestion base de données	Lukas Blanchot	Camille Rade
Back	Lukas Blanchot	Camille Rade
Front	Lukas Blanchot	Hugo Page

Planning des réunions effectuées depuis le début de la phase 2 du projet :



 Réunions

Remarque : Il y a eu de la communication écrite via Discord en dehors de ces réunions ainsi que les semaines au CESI.

4. Cahier de Recettes :

PV de recettes :

Client : Ministère de la Santé

Auteur : LHC Corporate

Nom de l'équipe qui a effectué les tests : LHC Corporate

Date de passage des tests : 01/03/2023 au 20/04/2023

PV de Recette

Résultats attendus :

N°	Point de vérification pour la version mobile et web	Conformité	
		OUI	NON
1	Visualisation ressources (non connectés)	OUI	
2	Inscription	OUI	
3	Connexion	OUI	
4	Déconnexion	OUI	
5	Affichage des ressources	OUI	
6	Mise en favoris	OUI	
7	Ajout de ressource (PDF, images etc....)	OUI	
8	Ajout de commentaire	OUI	
9	Suppression de commentaire	OUI	
10	Affichage profil	OUI	
11	Affichage des ressources mises en favoris	OUI	
12	Affichage de l'application	OUI	
13	Connexion a la base de données	OUI	

Remarques :

Signature client :

Ministère de la santé

Signature chef de projet :

Camille RADE

Version	Qui ?	Initialisation	Date
---------	-------	----------------	------

V1	Camille Rade Lukas Blanchot Hugo Page	Début de rédaction du cahier de recette	07/02/2023
V2			
V3			

4.1 Stratégie globale de test :

Les tests ont été effectués tout le long du développement et à la fin par Lukas Blanchot pour la partie mobile et Camille Rade pour la partie WEB.

Action à tester	Résultat attendu	Résultat du test mobile	Résultat du test web	Commentaire
Test super-admin	Ajouter des comptes administrateur			
Test super-admin	Ajouter des comptes modérateur			
Test super-admin	Suppression de compte administrateur			
Test super-admin	Suppression de compte modérateur			
Test administrateur	Succès de la connexion administrateur			

Test administrateur	Refus de la connexion administrateur : affichage pour remplir ses informations de connexion ou création de compte			
Test administrateur	Succès de la déconnexion administrateur			
Test administrateur	Modification des commentaires			
Test administrateur	Validation ressources : ajout effectué			
Test administrateur	Refus Ressources : ajout non effectué			
Test administrateur	Visualisation des tickets			
Test administrateur	Suppression de compte			
Test administrateur	Ajout de compte modérateur			
Test administrateur	Création des tickets			
Test administrateur	Suppression des commentaires			
Test administrateur	Ajouter un type de ressource			
Test administrateur	Supprimer un type de ressource			

Test administrateur	Modifier des données dans la base			
Test modérateur	Visualisation des tickets			
Test modérateur	Modification des commentaires			
Test modérateur	Suppression des commentaires			
Test modérateur	Validation ressources : ajout effectué			
Test modérateur	Refus Ressources : ajout non effectué			
Test modérateur	Succès de la connexion modérateur			
Test modérateur	Création des tickets			
Test modérateur	Refus de la connexion modérateur : affichage pour remplir ses informations de connexion ou création de compte			
Test modérateur	Succès de la déconnexion modérateur			

Test modérateur	Modification de la catégorie de ressources d'une ressource déjà publier			
Test citoyen connectés	Succès de la connexion connecté	OK	OK	
Test citoyen connectés	Refus de la connexion citoyen connecté : affichage pour remplir ses informations de connexion ou création de compte	OK	OK	
Test citoyen connectés	Succès de la déconnexion citoyen	OK	OK	
Test citoyen connectés	Création des tickets			
Test citoyen connectés	Modification des commentaires			
Test citoyen connectés	Suppression des commentaires	OK		
Test citoyen connectés	Ajout de commentaires	OK	OK	
Test citoyen connectés	Ajout de ressources	OK	OK	
Test citoyen connectés	Suppression de ressources (celles qu'il a créées uniquement)	OK		

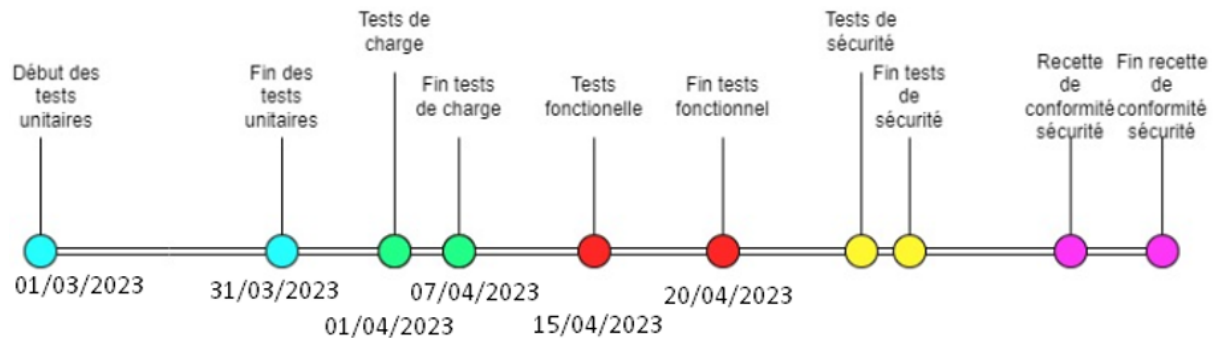
Test citoyen connectés	Modification de ressources (celles qu'il a créées uniquement)	OK		
Test citoyen connectés	Ajout de ressources en favoris	OK	OK	
Test citoyen connectés	Retirer ressources en favoris	OK	OK	
Test citoyen connectés	Partage de ressources		OK	
Test citoyen connectés	Filtre par types ressources	OK		
Test citoyen connectés	Filtre par catégories ressources	OK		
Test citoyen connectés	Ajout de relation			
Test citoyen connectés	Création de tickets			
Test citoyen connectés	Consultation des ressources	OK	OK	
Test citoyen connectés	Modification mot de passe			
Test citoyen connectés	Modification Adresse mail			
Test citoyen connectés	Modification Nom utilisateur			
Test citoyen connectés	Modification photo de profil			
Test citoyen connectés	Suppression relation			

Test citoyen connectés	Supprimer son compte utilisateur			
Test citoyen connectés	Restriction de la « visibilité » des publications par les relations			
Test citoyen connectés	Activation de compte par mail			
Test citoyen connectés	Récupération de mot de passe par mail			
Test citoyen non connectés	Visualisation ressources		OK	
Test citoyen non connectés	Créer un compte utilisateur		OK	
Test général	Connexion à la base de données	OK	OK	
Test général	Mise à jour des ressources	OK	OK	
Test général	Mise à jour des utilisateurs	OK		
Test général	Contrôle RGPD	OK		
Test général	Connexion à internet	OK	OK	
Test général	Affichage de l'application	OK	OK	

4.2 Mise en œuvre des tests :

Type de test	Quoi ?	Acteur / environnement si pertinent
Test unitaire	Un type de test dont l'objectif est la mesure du comportement d'un composant ou système avec une charge croissante, p.ex. nombre d'utilisateurs et/ou nombre de transactions en parallèle pour déterminer quelle charge peut être gérée par le composant ou système	En essai ou pré production par la Direction Production et Services
Test fonctionnel	Tests portant sur les fonctionnalités d'une ou des application(s) et/ou SI sur la base du référentiel des attendus ou des spécifications fonctionnelles générales (par processus métier) ou détaillées (par applicatif)	Par l'équipe de recette et/ou les métiers
Test de charge	Famille de tests visant à mesurer la mise en œuvre d'un applicatif sur son infrastructure cible pour garantir des temps de réponse acceptables. (Voir détails à chacune des rubriques)	Tous
Test de sécurité	Test de vulnérabilité, test d'intrusion, test de sécurité applicative	En pré production ou production par la Direction Production et Services
Recette de conformité sécurité	Vérifier que les mécanismes de sécurité ont été implémentés conformément aux solutions de sécurité définies et validées en amont, que les composants de la solution ont été intégrés dans le plan de sauvegarde du SI et enfin que la solution a été intégrée dans les procédures de secours du SI. Intègre notamment les contraintes RGPD : droit à l'oubli, droit d'accès aux données	Tous

4.4 Macro Planning :



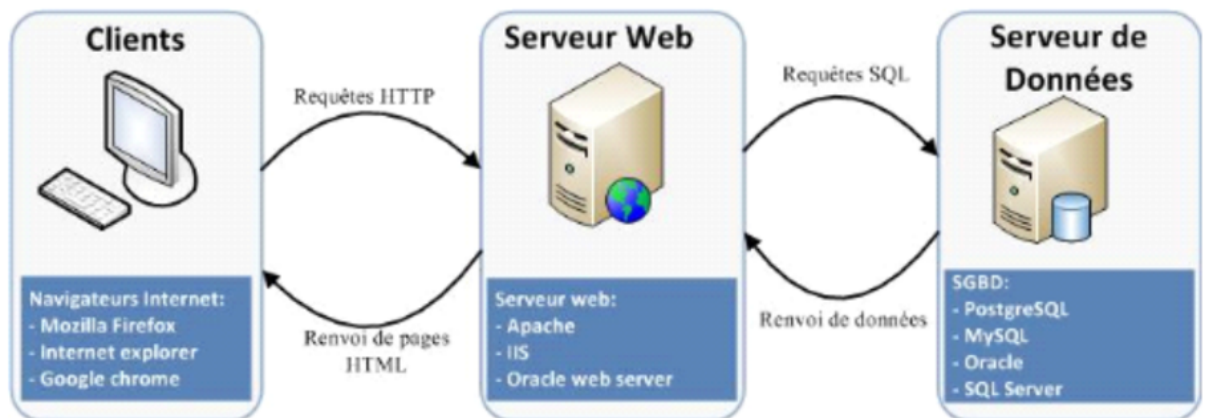
4.5 Scénario de test :

Scénario Connexion :

N°	Utilisateur	Système
1	Aller sur le site ressources relationnelles et cliquer sur le bouton "Se connecter"	Ouverture de la page de connexion
2	Entrer une adresse Email invalide sans @	Veuillez entrer une adresse mail valide
3	Entrée d'une adresse mail mais pas de mot de passe	Veuillez entrer un mot de passe
4	Entrer un mot de passe et une adresse mail mais pas de compte invalide	Aucun compte correspondant
5	Entrer un mot de passe et une adresse mail mais pas de compte valide	Redirection vers la page d'accueil

5. Architecture de l'application WEB :

Pour l'application Web nous avons décidé de travailler avec le Framework Symfony car nous avons tous des connaissances de base en PHP. Le site est accessible depuis un mobile ou depuis un ordinateur. De plus Symfony est basé sur le modèle MVC (Modèle-Vue-Contrôleur) :



Modèle : Cette partie représente la logique métier de l'application. Dans Symfony, les modèles sont généralement représentés par des classes de modèle qui contiennent des fonctions pour interagir avec la base de données et effectuer des opérations liées à la logique métier. Dans Symfony, le modèle est généralement géré à l'aide de l'ORM (Object-Rational Mapping) Doctrine, qui permet de mapper les objets PHP aux tables de la base de données.(Entity et Repository)

Vue : La vue représente la partie de l'application qui est visible par l'utilisateur final. Nous avons utilisé ici les vues Twig qui sont responsables de l'affichage de l'interface utilisateur.

Contrôleur : Le contrôleur est le cœur de l'application et est responsable de recevoir les requêtes de l'utilisateur, de traiter les données en utilisant les modèles appropriés, et de renvoyer une réponse appropriée à la vue. Dans Symfony, les contrôleurs sont représentés par des classes de contrôleur qui contiennent des actions qui sont exécutées en réponse aux requêtes de l'utilisateur. Dans Symfony, le contrôleur est généralement géré à l'aide du composant HttpFoundation, qui fournit des objets de requête et des réponses pour interagir avec le navigateur web.

En utilisant le modèle MVC, Symfony permet de séparer la logique métier de l'application de l'interface utilisateur. Cela rend l'application plus facile à maintenir et à étendre, car chaque composant peut être modifié indépendamment des autres.

Il n'y a pas d'API car pour la gestion des données nous avons utilisé le composant ORM de Symfony évoqué précédemment dans le document, mais cela peut faire partie de nos améliorations pour l'avenir.

5.1 Fonctionnement global de l'application Web :

Dans le contrôleur, le code de la fonctionnalité est généré pour accéder aux données nécessaires, en faisant appel au Entity et repository qui sont des mappages de la base de données. Depuis le contrôleur les données sont envoyées

à la vue. Ensuite dans le Template twig ses données sont accessibles et peuvent être affichées.

Chaque fonctionnalité est accessible via des redirections et des 'routes'.

Pour la partie authentification une partie "sécurité" a été mise en place, il y a un chiffrement de mot de passe, il n'est pas inscrit dans la base de données en clair. Nous avons également utilisé un jeton CSRF* qui est généré lors de l'envoi de formulaire afin de vérifier que la personne est bien authentifiée et qu'elle a les droits pour effectuer cet ajout.

***Le jeton CSRF est une protection qui requiert l'insertion d'une valeur aléatoire et dynamique dans une requête. Cette valeur est ensuite analysée par le serveur pour déterminer si la requête est légitime.**

En ce qui concerne les tests à chaque fonctionnalité développées les tests ont été effectués directement après non régression (impact pas les fonctionnalités développées précédemment) ainsi que des tests unitaires.

Nous avons voulu apporter aux utilisateurs une simplicité dans la consultation, l'ajout, et la manipulation des ressources. Elles sont composées de titres, textes et différents formats tels que des photos, des vidéos ou des fichiers PDF.

Pour conclure la partie application Symfony nous avons principalement utilisé des composants propres au Framework (Security, ORM Doctrine etc...). Mais dans une vision d'évolution et d'amélioration nous pouvons mettre en place une API.

5.2 Partie Front de l'application WEB :

Bootstrap est un Framework CSS open source qui fournit une bibliothèque complète de composants prédéfinis pour la mise en page et la conception d'interfaces utilisateur réactives. L'utilisation de Bootstrap avec Symfony est très simple, car Bootstrap est une bibliothèque frontend qui peut être facilement intégrée dans les Template Twig de Symfony.

En utilisant les classes CSS prédéfinies de Bootstrap dans les Template Twig, nous pouvons créer des designs élégants et réactifs en peu de temps. Nous pouvons également utiliser des plugins jQuery pour des fonctionnalités supplémentaires comme les menus déroulants, les modales et les carrousels.

Bootstrap offre également une compatibilité avec une variété de langages de programmation et de technologies web, notamment HTML, CSS, JavaScript et jQuery, ce qui en fait un excellent choix pour le développement web avec Symfony.

En résumé, Bootstrap est un choix judicieux pour le développement web avec Symfony en raison de sa bibliothèque complète de composants prédéfinis, de sa facilité d'utilisation.

Avec les Template Twig de Symfony, de sa compatibilité avec une variété de langages de programmation et de technologies web, et de sa facilité d'apprentissage.

6. Architecture de l'application mobile :

Flutter est un Framework open-source créé par Google pour le développement d'applications mobiles pour Android, iOS et d'autres plateformes. Il utilise le langage de programmation Dart et permet aux développeurs de créer des applications avec une seule base de code pour plusieurs plateformes.

Le fonctionnement de Flutter est basé sur le concept de "widgets", qui sont des éléments de l'interface utilisateur tels que des boutons, des champs de texte et des images. Les développeurs peuvent utiliser ces widgets pour créer des interfaces utilisateur personnalisées et interactives. Flutter dispose également d'une bibliothèque de widgets prêts à l'emploi pour accélérer le développement.

Les changements de page dans Flutter sont réalisés en utilisant la navigation, cela permet à nos utilisateurs de naviguer entre les différentes pages ou écrans de l'application.

Par ces éléments, flutter était la solution la plus propice pour nos développements. Nous voulions également apporter aux utilisateurs une simplicité dans la consultation, l'ajout, et la manipulation des ressources. Celles-ci sont composées de titres, textes et différents formats tels que des photos, des vidéos ou des fichiers PDF.

Pour répondre à ce besoin, nous avons développé une API qui est reliée à une base de données pour stocker les ressources et les informations utilisateur. L'API communique avec la base de données en utilisant des scripts PHP pour effectuer des requêtes SQL avec des filtres.

L'application mobile a été développée avec Flutter, qui permet de créer une seule application compatible avec iOS et Android en même temps. L'interface utilisateur est conçue avec des fonctionnalités telles que l'inscription, la connexion, l'ajout de ressources, l'ajout de commentaires, la mise en favoris, la consultation de ressources et la recherche d'utilisateurs et de ressources.

En ce qui concerne les tests, nous en avons effectué à chaque étape du développement, comprenant des tests unitaires, des tests d'intégration et des tests de performance pour assurer la qualité de l'application et éviter les bugs et les erreurs dans la production.

Pour parler de la protection, il y a un chiffrement de mot de passe (BCRYPT) donc aucun mot de passe n'est enregistré en brut dans la base de données.

7. Budget

Rappel :

Nous avons un budget de 90 000€ hors taxes et le projet se déroule sur une période de 253 jours travaillés. Soit un an de projet.

Hébergement :

- **Hébergement web** : 50€ / mois soit 600€ / an
- **Réalisation des tests (locaux)** : 0€
- **Certificat SSL** : 50€ / an
- **Nom de domaine** : 0€
- **Système de gestion de contenu** : 0€ (Si WordPress, Joomla, Drupal, PrestaShop, Magento, Odoo)
- **Thème** : 60€ par an.
- **Plugins et fonctionnalités supplémentaires** : 0€

Coût des développeurs par jours (jours hommes) :

2100 / mois x 3 développeurs débutant = 75 600€ pour un an.

Coût des locaux :

Tarif moyen open-space 670€/mois soit 8040€ pour un an

Bureau 2 à 3 postes :

- Contrats de prestations (ni caution, ni dépôt de garantie), Électricité, chauffage et ménage inclus
- Accès 24/24h 7/7j (application Smartphone)
- Internet Très Haut Débit (200 Mbits/s Sym), fibre optique, Ethernet et Wifi
- Accès libre Cuisine (entièrement équipée)
- Accès libre Accueil (agencé pour téléphonie mobile)
- Accès libre Salle de Réunion (hors privatisation)

(Forfaits "Résidents")

- Téléphonie fixe avec numéro personnel attribué et facturation mensuelle détaillée

Estimation Finale : 84 350€

8. Conclusion

Pour conclure ce second livrable, nous avons pris en compte la demande du client afin de lui fournir ces deux applications. Nous avons réparti le travail que nous avons précédemment divisé en trois parties. Lukas s'est occupé de la partie mobile et du serveur du projet, Camille s'est occupé de la partie Back ainsi que de la partie Base de Données du projet et Hugo s'est occupé de la partie Front du projet.

De plus, nous avons utilisé des technologies récentes, fiables et connues pour développer ces applications. Le fait d'utiliser des Framework (Symfony / React / MUI) ainsi que des patterns comme le MVC nous permettent de développer plus simplement le projet. Ils nous permettent également de faciliter l'ajout de fonctionnalités et de le sécuriser. Nous avons mis l'accent sur la sécurité des applications car c'est une partie importante dans une application demandée par l'État.

ANNEXES

ANNEXE:

Liste des fonctionnalités qui ont été développées sur la partie Symfony :

- Connexion
- Déconnexion
- Inscription
- Ajout de ressources
- Mise en favoris
- Fil d'actualité

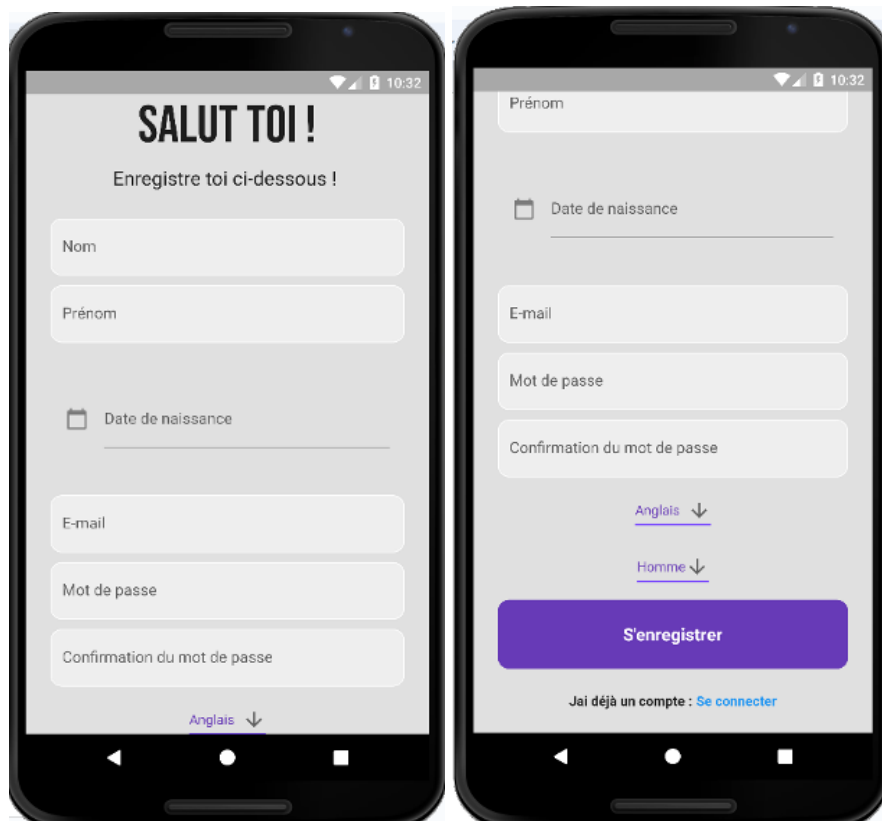
SCREEN DE L'APPLICATION MOBILE :

Quelques écrans de la partie mobile réalisés :

- Connexion



- Inscription



- Déconnexion



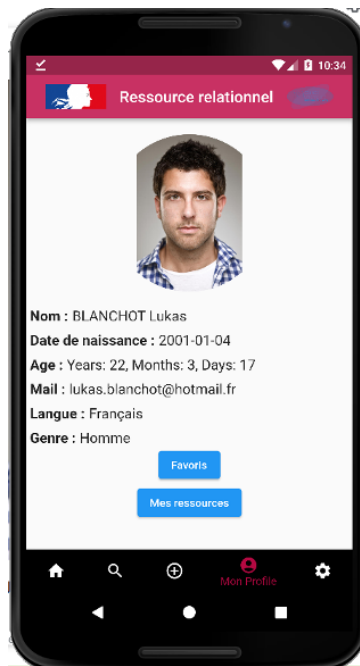
- Fil d'actualité



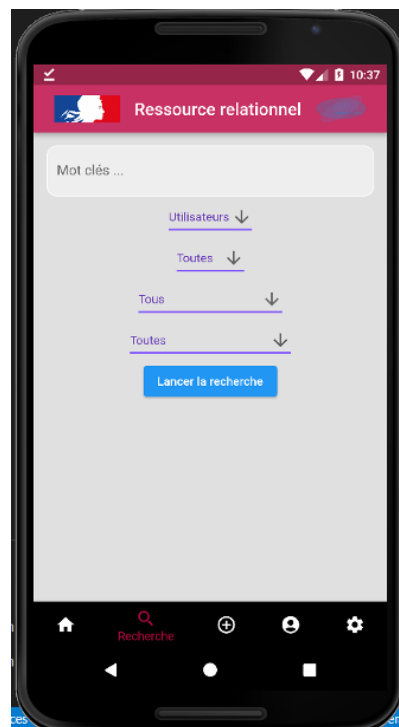
- Ajout de ressources



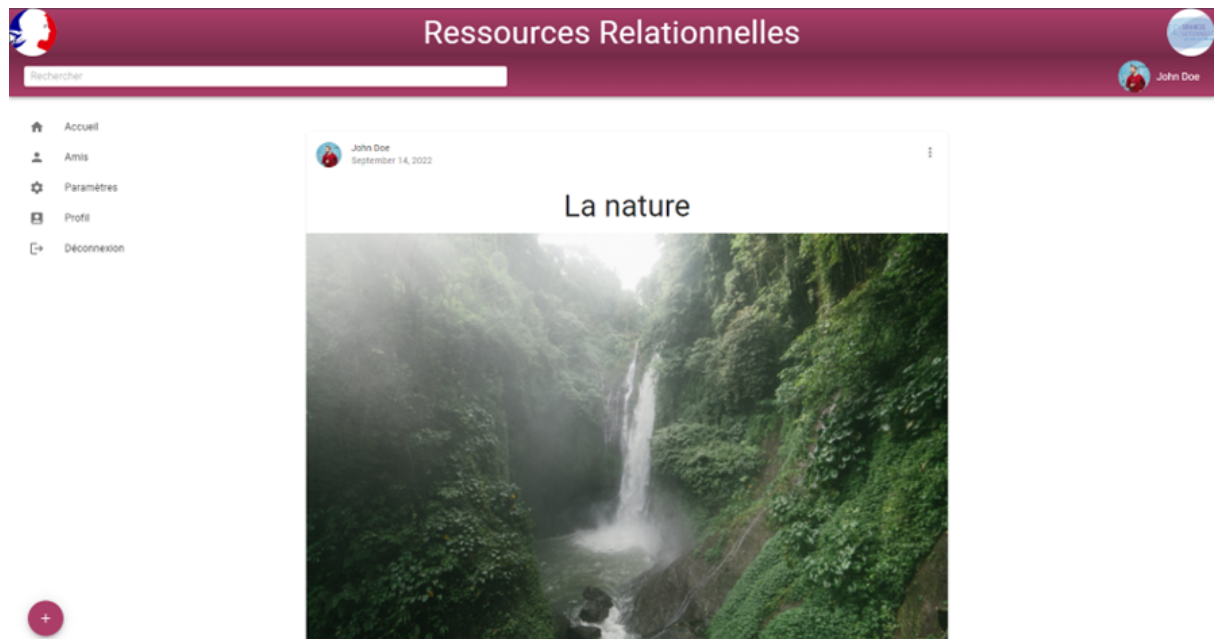
- Mon profil



- Recherche



Premier Rendu de l'application Web (Version Ordinateur) :



Premier Rendu de l'application Web (Version Mobile) :

