

MEYNADIER Renaud

**Développeur Web et Web Mobile.**

**Dossier projet**

**REMERCIEMENTS :**

Je voudrais remercier toutes les personnes sans qui ce projet n’aurait pas pu aboutir.

Mes remerciements vont donc à Human Booster, Pôle Emploi, Be-Ys Software ainsi que ses développeurs qui m’ont accueilli au sein de leur équipe.

Plus précisément l’équipe Human Booster qui a fourni les locaux, le matériel, la connexion à notre disposition.

A mes collègues de promotion de cette formation, avec qui on a quand même passé une bonne année tous ensemble.

A l’entreprise Be-Ys Software qui est vraiment stimulante, et est un vrai cadre de paradis pour l’évolution avec des professionnels.

Bien sûr à Pôle Emploi sans qui rien de tout cela ne serait possible sans leur investissement dans cette formation.

**Le projet doit couvrir les compétences suivantes :**

Le stage de Be-Ys Software couvre la partie front du projet :

CCP1 - Pour la partie « Développer la partie front-end d’une application web ou web mobile en intégrant les recommandations de sécurité »

* Maquetter une application
* Réaliser et développer une interface utilisateur web adaptable statique et dynamique
* Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

Mon projet annexe “fil rouge” couvre la partie back du projet :

CCP2 - Pour la partie « Développer la partie back-end d’une application web ou web mobile en intégrant les recommandations de sécurité »

* Développer les composants d’accès aux données
* Développer la partie back-end d’une application web ou web mobile
* Développer les composants d’accès aux données
* Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

**Les technologies abordées en formation sont les suivantes :**

**HTML - CSS - JS - TYPESCRIPT - ANGULAR - SYMFONY - PHP - SQL - DQL**

Précision :

Mon stage ne couvrant que la partie ‘’front” du métier de développeur, j’ai conçu en parallèle un site de e-commerce, permettant de compléter la partie fournie par le stage.

**SOMMAIRE**

**Résumé………………………………………………………………………………………………………………...Page 5**

**Présentation du dossier projet ……….…………………………………………………………………………...Page 6**

**PARTIE I - Le Stage à Be-Ys Software Almerys ………………………………..………………………………..Page 7**

**Résumé………………………………………………………………………………………………………………...Page 8**

**Résumé………………………………………………………………………………………………………………...Page 9**

**Résumé……………………………………………………………………………………………………………….Page 10**

**Résumé……………………………………………………………………………………………………………….Page 11**

**Résumé……………………………………………………………………………………………………………….Page 12**

**Résumé……………………………………………………………………………………………………………….Page 13**

**Résumé……………………………………………………………………………………………………………….Page 14**

**Résumé……………………………………………………………………………………………………………….Page 15**

**Résumé……………………………………………………………………………………………………………….Page 16**

**Résumé……………………………………………………………………………………………………………….Page 17**

**Résumé……………………………………………………………………………………………………………….Page 18**

**Résumé……………………………………………………………………………………………………………….Page 19**

**Résumé……………………………………………………………………………………………………………….Page 20**

**Résumé……………………………………………………………………………………………………………….Page 21**

**Résumé……………………………………………………………………………………………………………….Page 22**

**Résumé……………………………………………………………………………………………………………….Page 23**

**Résumé……………………………………………………………………………………………………………….Page 24**

**Résumé……………………………………………………………………………………………………………….Page 25**

**Résumé……………………………………………………………………………………………………………….Page 26**

**Résumé……………………………………………………………………………………………………………….Page 27**

**Résumé……………………………………………………………………………………………………………….Page 28**

**Résumé……………………………………………………………………………………………………………….Page 29**

**Résumé……………………………………………………………………………………………………………….Page 30**

**Résumé……………………………………………………………………………………………………………….Page 31**

**Résumé :**

Be-Almerys, est une entité de plus de 2000 collaborateurs dont 400 en France, qui propose des solutions et services pour le traitement de données, gestion de conformité, distribution de produits, délégation de gestion et la mise en œuvre des services différenciant. C’est un expert de haute technologie appliqué à l’assurance et à la santé. L’entreprise est composée de sous-entités :

Beys Trusted Solutions, Beys Pay, Beys Health Solutions, Beys Search, Beys Software et la Citadelle.

Ses sous-entités ont pour objectifs de :

* Sécuriser le bien le plus précieux de l’entreprise : ses données.
* Digitaliser les processus métiers de l’entreprise pour lui permettre de réduire ses coûts et d’améliorer ses ventes.
* Sécuriser juridiquement le client lors des prises d’engagement.
* Réduire et sécuriser les coûts d’archivage et de transfert de documents.

Ils ont différents clients : La Banque Postale (assurance), Generali, Mutuelle Générale, Swisslife…

J’ai donc fait un stage dans l’équipe ‘’front’’ au sein de Be-Ys Software France, Clermont-Ferrand. (Ils sont présents aussi à Paris, Toulouse, Annecy, au Luxembourg et à Tunis).

J’ai travaillé sur une application Angular IAM-FRONT, qui permet la gestion de rôles, de données pour les professionnels de la santé.

En ce moment l’application est refondue en version 2.0. Les tâches sont divisées en plusieurs Sprint de trois semaines, qui sont eux même divisés en plusieurs tickets. J’ai eu la chance de pouvoir prendre quelques tickets front, des correctifs de modal, de la correction de CSS et l’utilisation de leur librairie Angular.

J’ai eu l’opportunité de corriger quelques problèmes de traduction, de rendre l’interface plus intuitive et facile d’accès. J’ai eu recours à des appels d’API, et des dashboard pour gérer les comptes et identifiants.

Mes tickets sont ensuite analysés (pipeline gitlab), traités et validés (avec Jira), s’ils ont passé une première série de tests. J’ai utilisé MatterMost, Skype Entreprise et Outlook qui permettent de faire des retours très rapide.

Rappel : Le stage ne couvrant pas la partie ‘’back’’ j’ai conçu en parallèle un site de e-commerce : La Nimes’Alerie, que je vous présenterai juste après.

**PRÉSENTATION DU DOSSIER PROJET**

Dans cette partie je vais vous expliquer comment le cheminement de ce dossier va s’effectuer.

C’est en quatre parties que cette épreuve va se dérouler.

PARTIE I - Le Stage à Be-Ys Software Almerys

PARTIE II - Projet fil rouge : La Nimes’Alerie

CONCLUSION - (satisfactions, difficultés rencontrées, projet professionnel, finalité du projet, emploi)

ANNEXE Formation et capture d’écran ?

**PARTIE I - Le Stage à Be-Ys Software Almerys**

**• Cahier des charges (présentation du projet, des besoins de l’entreprise, la problématique, etc.)**

Be-Ys Software France, fait des interface front ‘’IAM’’ (Identity Access Management).

En ce moment l’équipe front que j’ai intégré fait une refonte totale de l’interface vers une deuxième version. Nous vendons des interfaces sécurisées à des clients sensible, protégeons leurs données. L’interface front doit être intuitive, claire, complète et facilement accessible.

**Note pour des recherches efficaces :**

Il est important de rechercher par mots-clés et je te conseille la structure suivante : Nom de la techno + élément technique OU erreur.

Ensuite, il est important d'identifier les points ci-dessous :

Identifier la technologie de travail en cours, que ce soit un langage, un framework ou un autre type de techno, il faut toujours commencer ta recherche par ce mot-clé

(Exemple : Javascript, Angular, CSS...)

Citer l'élément technique ou l'erreur affichée à l'écran.

* Cela peut être un concept propre à la techno si tu cherches à te documenter dessus

(Ex : BehaviourSubject en Angular)

* Ou bien l'erreur affichée sur ta console de la page

(Ex : Cannot read "length" of undefined)

Privilégier la doc officielle de la technologie ou StackOverflow.

Ce sont les premières ressources d'informations à privilégier, ne pas hésiter à détailler la recherche comme ceci :

Angular FormArray

Typescript cannot read "length" of undefined

Java NullPointerException

**• Spécifications techniques du projet (l’organisation de travail, environnement, outils, techno, etc.)**

Webstorm : Un IDE pour les langages Web, adéquat pour Angular. (VS Code également)

Firefox Developer : Une version pour développeur, avec des fonctions inédites.

Notepad : Pour la prise de note diverse, les journaux de bords, mémo et liste à faire.

Git Fork : Permet de visualiser ses branches en temps réel et les collaborateurs.

Keycloak : Logiciel Open source ouvert, permet l’authentification unique par identité.

Adobe XD : Aide à la construction de maquette pour le développement web.

Planit Poker : Aide à l’estimation des tâches pour les sprints à venir.

PowerShell : Permet d’exécuter des lignes de commandes, indispensable ici.

Chocolatey software : Gestionnaire de package en ligne de commande et installation.

Mattermost : C’est un open source de slack, il est similaire à Discord avec des salons.

Jira : Système de suivi de bug, gestion d’incident, prise et assignation de tickets.

Skype entreprise : Nécessaire pour les réunions journalières, remplacée par Teams.

Outlook Pro : Messagerie professionnelle pour gérer les retours client, tâches, erreur...

Ces outils sont utilisés quotidiennement pour les tâches effectuées à Be-Ys Software. Du simple Outlook jusqu’au Webstorm, cela devient la prise d’information jusqu’à son exécution.

**Une journée type à Be-Ys peut être schématisée ainsi :**

* Prise de poste à 9 heures.
* Vérification des messages sur Mattermost, des ‘’daily’’ (réunions à 10h45 jusqu’à 11h pour faire le point) avec Skype Entreprise, des e-mails automatique (pipeline gitlab acceptée, refusée, à modifier…), des retours des collègues en distanciel, avec Outlook Pro.
* Pendant la définition d’un nouveau sprint on utilise planitpoker qui aidera le PO et l’équipe à estimer la difficulté d’une tâche dans ledit sprint.
* Avec Jira, prise de nouvelles tâches à faire, de ticket, de commentaire à ajouter à l’équipe sur un travail à faire, réponse aux collègues qui veulent parfois des précisions sur votre travail.
* Prise de note pendant les réunions des fonctionnalités à faire avec bloc note (un cahier fonctionne aussi), utile aussi pour transmettre du code, faire des tâches…
* Parfois on doit utiliser PowerShell et Chocolatey Software pour installer certains logiciels (blocage strict de l’entreprise sur certains domaines), même Google Docs n’est pas toléré.
* On peut utiliser occasionnellement Adobe XD mais Be-Ys dispose d’un bureau à côté composé d’une équipe UX/UI.
* Webstorm et Firefox developer nous permettent le plus gros du travail, le développement web bien sûr. Avec l’aide aussi de Google et internet.
* Utilisation de keycloak pour la gestion d’identité.
* Gitfork je l’utilise personnellement, mais l’équipe utilise des commandes dans le terminal. (Git checkout -b ou -D, git status, git add -A, git commit -m, git push origin, « mais aussi » : git log, git fetch origin alpha, git cherry-pick)

**Quelques commandes propres au projet Angular IAM-FRONT :**

* Npm (node package manager) run start:int (intégration)
* Npm run lint (passe le code en revue à la recherche d’erreur)
* Npm run test (permet d’avoir une première version du code uniforme et propre, retour dans le navigateur avec ‘jasmine’, erreur à corriger pour pouvoir faire une merge request et vérifier les tests pipeline dans gitlab)
* Npm run watch (dans la librairie)
* Npm link dist/common (pour que NPM fasse un lien symbolique de ce répertoire vers sa base de dépendance)
* Npm link @beys-software/common (Permet de relier une au IAM FRONT qui recevra la dépendance)
* Npm run get:assets (Récupère les assets du projet angular)
* Npm install autocompleter (Dépendance d’auto-complétion)

**• Réalisations du candidat comportant les extraits de code les plus significatifs et en les argumentant, y compris pour la sécurité et le web mobile**

Ici je peux simplement montrer des bouts de code Angular hors contexte par capture d’écran. Les autres captures seront en annexe et concerneront le projet fil rouge.

**• Présentation du jeu d’essai élaboré par le candidat de la fonctionnalité la plus représentative (données en entrée, données attendues, données obtenues)**

**• Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité**

Ces deux rubriques sera plus étoffée avec le projet fil rouge. Be-Ys ne m’autorise pas à montrer leur projet en dehors de capture d’écran vérifiée. Je n’ai pas le droit par exemple de cloner leur projet sur l’ordinateur d’Human Booster. Après tout, c’est cohérent avec la politique de l’entreprise, les données sensibles restent à l’abri. Je montrerai plus en détail avec une démonstration du site de la Nimes’Alerie.

**• Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité**

**• Description d’une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de site anglophone**

**• Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment, accompagné de la traduction en français effectuée par le candidat sans traducteur automatique (environ 750 signes)**

Pour l’anecdote, j’ai même réussi à faire ce genre de recherche pour réparer un fichier qui refusait de se lancer dans un de mes jeux favoris Steam. Il s’agissait du jeu Oblivion (fait par Bethesda) qui plantait systématiquement à la suite d’un événement donné. Il a fallu rechercher la solution sur internet, rechercher le problème et apporter la correction. Bien sûr cela peut faire sourire mais, de mon point de vue, je ne trouve pas cela forcément très différent. Maintenant le jeu est jouable. Ce genre de problème arrive dans beaucoup de logiciels, jeux vidéo, applications, mais même sur tout support informatisé, ordinateur, téléphone, console de salons, etc…

**• Journal de bord, en entreprise Be-Ys Software France**

**Présentation :**

Ici j’ai noté le contenu essentiel et marquant de mes journées. Evidemment certaines journées sont plus mouvementées que d'autres. Parfois j’avais de la veille à faire, de l’apprentissage. J’ai repris à zéro Angular, et me rend compte qu’il est évidemment (?) plus poussé et costaud que ce qu’on a pu aborder pendant la formation. Le stage s’est fait en deux parties, cela a été judicieux, les premières semaines je faisais des corrections mineures, simples, j’apprenais à me servir de leurs multiples interfaces. J’ai même eu parfois pendant des temps plus calmes en fin de sprint, pu travailler sur mon fil rouge, et effectuer des exercices. De plus les appels aux endpoints d’une API, les collègues de BeYs savent faire, de même que des dashboard, j’ai pu en voir pendant mon stage, ce qui fut pratique pour visualiser le travail à mettre dans le projet annexe. Et puis pour donner suite au point de mi-stage, on a passé la seconde ! Ils m’ont donné un composant Angular entier, à fabriquer, dans un échantillon de librairie et de le relier à leur librairie centrale. Ce fut une tâche ardue, mais un de mes objectifs de stage à faire. J’ai aussi pris des captures d’écrans de certains tickets, évidemment tout est surveillé par l’entreprise qui ne m’autorise pas à sortir du code sensible, par ailleurs, il est hélas impossible de cloner leur projet à la maison, seuls les employés ont ce droit.

**Contenu du journal** **:**

Ce projet est majoritairement Front avec TypeScript & Angular. Participation à la refonte de IAM Front V1 vers IAM Front V2.

Journal de bord : Stage Be-Ys Software France 2 Mars - 29 Avril 2022.

**Date : Le 2 Mars 2022.**

Premier jour de stage, configuration de l'environnement de travail.

Prise de connaissance du projet IAM front.

Résolution d'un premier ticket permettant de naviguer dans l'application Angular (IAM front toujours).

Utilisation de JavaScript dans une page Angular pour changer dynamiquement des informations.

Appel de la librairie BeYs/common au projet IAM FRONT grâce à la commande ‘link’.

Le projet a été modifié, corrigé et poussé sur la branche alpha.

À la suite de cela, une merge request (MR) est faite, qui sera vérifiée et acceptée si les tests qualité sont tous cochés. Après, la correction sera définitivement appliquée.

Une capture d’écran a été effectuée.

**Le 3 Mars 2022.**

Aujourd’hui j’ai pu installer en parallèle le projet fil rouge pendant le temps de pause. Configuration symfony, amélioration de la navbar symfony.

A été assigné un ticket de correction CSS dans la librairie de material design d'Angular. Recherche fastidieuse et solution finalement trouvée.

Cette dernière fut vérifiée, et ajoutée au projet IAM Front.

(Début tranquillement dans le stage pour bien voir la globalité du projet.)

Nom du ticket : Ticket 3076 : Amélioration textes

Une capture d’écran a été effectuée.

**Le 4 Mars 2022.**

Nouveau ticket obtenu. Correction du problème : Lorsqu'un utilisateur n'a aucun offreur de service dans ses attributs, la page reste blanche.

Recherche du problème dans le projet Angular. Problème identifié ? Utilisation de la solution adéquate.

Apprentissage de la fonction .pipe en Observable, c’est une fonction rxjs qui permet de combiner deux fonctions en une, appelée le pipe.

Une capture d’écran a été effectuée.

**Le 7 Mars 2022.**

Ticket du 4 Mars terminé.

Exemple de rapport sur Mattermost :

MR FRONT correction du ticket 3098 : Lorsqu'un utilisateur n'a aucun offreur de service dans ses attributs la page reste blanche (fixing blank page when no services providers)

https://sources.devtools.local/gebeeiam/beiam/iam-front/-/merge\_requests/139

Correction faite dans le resolver.ts, puis le component.ts adéquat et le template html de ce dit component. (Utilisation de ngIf sur une longueur de tableau)

En clair : Si le rôle est de ‘USER’ nous verrons la page (avant correction la page ne chargeait pas), mais dans cette page aucun offreur de service ne sera visible, la page ne sera pas pour autant complètement blanche.

Si le rôle est admin, la page nous montrera les domaines. Si le rôle est ‘SUPER\_ADMIN’, la page nous montrera l’intégralité du contenu, domaine et offreurs. Correction soumise et envoyée sur la branche alpha.

Une capture d’écran a été effectuée.

**Le 8 Mars 2022.**

Une attaque de type DOS (Denial of Service) a été faite contre Almerys, comme l'indique le rapport bruteforce.

Correction de problème de pipeline et de suppression d'indentation. En effet le test de qualité est très strict, tout doit être indenté pareillement, la suppression de certains espaces n’est pas autorisée, le projet doit être uniforme sur toute ses pages.

Avancée rapide à côté sur le fil rouge avec l'ajout d'un petit footer. Amélioration du site projet fil rouge, étant donné qu'aucun autre ticket ne fut disponible aujourd'hui.

Une capture d’écran a été effectuée.

**Le 9 Mars 2022.**

Après chaque ticket traité, ne pas oublier de glisser le ticket dans les tâches effectuées dans le tableau de suivi JIRA.

Problème sur Gitlab depuis quelques jours. Avancée sur le business case pendant la résolution du problème interne. (Rien que je puisse faire hélas).

Le problème a été résolu, c'était un problème dans le framework angular de testing 'jasmine', il s'agissait d'un problème d'injection d'un service dans le composant testé, mais n'était pas présent dans un 'testbed'.

Il fallait que les conditions soient remplies pour les éléments testés. Nous avons donc modifié le composant, et donc on a pu modifier les conditions d'affichage.

On s'est assuré par la suite du bon déroulement des tests. Pour pouvoir pousser sur la branche alpha.

Une capture d’écran a été effectuée.

**Le 10 Mars 2022.**

Je n’ai pas eu de ticket aujourd'hui, j’ai donc pu améliorer mon business case. Des soucis à régler sur le site avec une barre de navigation irrégulière. Aussi, j’ai eu une proposition de démonstration d’Adobe XD avec l’équipe voisine de designer web.

Une capture d’écran a été effectuée.

**Le 11 Mars 2022.**

Des nouveaux tickets vont arriver pour les journées à venir. Amélioration du business case, page de base spatiale presque totalement terminée.

Faire une page de forum dans la page base terrestre à renommer en 'informations' ou 'communications' terrestres ?

Ticket 3107 obtenu et résolution du problème presque fini, un point et demi sur trois.

Reprise du travail lundi. Faire condition NGIF dans le template approprié selon le mode dans lequel nous sommes ''[required]="true"''

Une capture d’écran a été effectuée.

**Le 14 Mars 2022.**

Poursuite du ticket 3107 Corrections interface front.

Leçon sur les Angular reactive forms. Point du stage à faire avec un component lourd Angular à préparer en parallèle.

2eme point du ticket traité et corrigé.

Il s'agissait de trouver dans le projet angular le moyen de rendre le champ pour créer (ajouter/add) un utilisateur obligatoire, mais en mode édition rendre ce champ non obligatoire.Commande git notée avec link de librairie.

Prise de note faite sur les liens link de la librairie Beys au projet.

Une capture d’écran a été effectuée.

**Le 15 Mars 2022.**

Pas de nouveau ticket. Vérification des pipelines. Le sprint 57 se termine le 18 Mars 2023.

Nouvelles commandes git notées. Branche 3107 propre à présent.

Reactive forms en angular crée. Maintenant, à voir la différence avec un template driven forms.

Lancement des reactives forms, j'ai rassemblé mes anciens cours Angular d'Human Booster pour compléter.

Forum symfony à améliorer pour intégrer dans le business case.

Pas besoin de capture d'écran ici. (Journée d’apprentissage)

**Le 16 Mars 2022.**

Amélioration du forum pour l'intégration dans le fil rouge.

Nouveau ticket obtenu 3123 : [N2] Navigation sur les écrans de sélection de providers/domaines.

Une sidebar qui apparaît, persiste si l'on fait précédent dans le navigateur.

Correction du ticket terminée. Dans le ngOnInit, faire la fonction dans la condition d'affichage sur FALSE.

Une capture d’écran a été effectuée.

**Le 17 Mars 2022.**

Fin de sprint. Réunion habituelle avec rapport sur le ticket 3123.

Aucune assignation aujourd'hui, a pu réviser et continuer un peu le projet fil rouge.

La partie du forum a été refaite.

Il reste donc à faire : Page d'accueil - Page boutique - Dashboard - Responsive Pas besoin de capture d'écran ici. (Veille technologique, fil rouge)

**Le 18 Mars 2022.**

Mes tickets ont été vérifiés et validés. Confirmation sur le mail de l'entreprise. (Outlook pro)

Présentation de la fin du sprint.

Transition sur le front-v2. Démonstration IAM Gestion.

Contenu de la sprint review :

Ticket 2102 : Utilisation de l'API Almerys et Swagger UI. Attribution des rôles avec test sur le rôle admin.

Ticket 3051 : Ajouter ou modifier un attribut via l'API. (Action avec SUPER\_ADMIN/ ADMIN) Erreur 202 non désirée au lieu d'une 204. Retour et corrections sur le FRONT.

Ticket 1920 : En tant qu’ADMIN ou DOMAIN\_ADMIN, je dois pouvoir supprimer une application. Création d'une application rapide par API ‘Swagger’ UI. Suppression et retour 204.

Ticket 3106 : En tant qu'utilisateur connecté à IAM Gestion, je veux configurer les préférences du profil. Dark Mode, langages ajoutés.

Ticket 3075 : En tant qu'admin, je veux pouvoir filtrer les flux d'activité (traçabilité). Filtre et correction du flux historique, du plus récent au plus ancien.

Ticket 3070 : [GEBBEID] Update filebeat role for nuta deploiement.

Ticket 3107 : Corrections interface front. Amélioration des textes et interfaces IAM Front v2. Pour rendre l'application polyvalente.

Des rapports de tests ont été effectués. 16 sont en échec (51.61%) et 15 sont réussi (48.39%). Réparations des échecs de tests.

Assignation du ticket 3132 : Améliorations interface, deux points sur trois traités, poursuite lundi 21 Mars.

Une capture d’écran a été effectuée.

**Le 21 Mars 2022.**

Poursuite du ticket 3132.

Point 1 & 2 terminés.

Pour le point 3 du ticket, peut-être faire une fonction de switch dark mode qui ferait un RELOAD en dehors de l'init avec appel html ?

Des nouveaux points ont été ajouté =

4 La notification d'ajout d'un groupe => doit afficher "ajouter" et non "créé"

5 Contrôle des rôles : Alignement des boutons << < > >> entre les 2 input liste. Les aligner verticalement.

6 Contrôle des rôles : Les ascenseurs ne doivent pas s'afficher si pas scrollable. Dans les blocs de rôles. Chercher du côté de la librairie.

Définition du sprint 58. Les choses sont à corriger selon le retour des testeurs.

Petite correction d'un problème de navbar, étude sur la conversion d'une navbar parfaitement responsive.

Pour ce sprint 58 nous utilisons planitpoker, les votes portent sur la complexité des tâches à effectuer.

Une capture d’écran à été effectuée.

**Le 22 Mars 2022.**

Poursuite du ticket 3132.

Point 1, 2, 4, 5 & 6 terminés.

Pour corriger les points 5 et 6 de contrôle des rôles, il à fallu donner une hauteur fixe à des div parentes qui après par une fonction de calcul, ont permis de diviser par 4 les hauteurs pour répartir les 4 boutons de contrôle. << < > >>

En modifiant donc la CSS et des div parentes on a pu afficher correctement les contrôleurs. Avec de plus une propriété CSS overflow, désactivation des ascenceurs sur les boîtes de contrôle.

A côté de cela, la barre de navigation du fil rouge a été problématique mais à finalement été corrigée. Tout le business case est donc maintenant responsive.

Le module front est donc validé normalement. Il reste à faire la boutique, l'accueil et le dashboard.

Réunion à 16 heures pour les modifications de la librairie beys software. A chaque sprint les modifications seront faites.

Capture d'écran faite.

**Le 23 Mars 2022.**

Léger souci sur le ticket 3076 qui a été corrigé.

Ticket 3132 avec le DarkMode à poursuivre. Tous les autres points sont corrigés.

Apparition d'un problème avec le ticket 3107. Résoudre le tout et pousser(push) le correctif dans une seule branche.

Ticket 3132 terminé. Il s'agissait d'un problème de DarkMode avec la librairie personnalisée de Be-Ys Software. Cette librairie est un composant enfant de Iam Front. <beys-navbar></beys-navbar> en est l'exemple. Le html est appelé dans un autre template html par ces balises.

Pour solutionner ce problème épineux il a fallu d'abord :

- Définir les couleurs dans une fonction ''init'', qui seront les couleurs des modes light/dark.

- Faire un 'output' dans le composant (le navbar.component.ts dans la libs beys), ensuite le brancher sur le html. (Navbar.component.html)

- Faire des fonctions, une dans le ''ui.service.ts'', une dans le ''main.component.ts'', appel du ui.service et du setColor dans l'initData.

- Raccorder le tout dans le main.component.html.

IAM FRONT (html principal) -> BEYS LIB (librairie html indépendante) -> darkmode - output/emit -> main.ts IAMFRONT enableDarkMode -> uIservice 'isDarkMode' -> Dashboard

'isDarkMode.subscribe'

Ce petit circuit permet de vérifier dans quel état nous sommes, et par un simple boolean 'true false', de changer les couleurs des éléments sur la couleur du reste du site.

Une capture d’écran a été effectuée.

**Le 24 Mars 2022.**

Aujourd’hui j’ai pu débuter la page d'accueil du business case. Puis aussi faire le début de la page résumé du projet Human Booster.

Le ticket 3076 et le ticket 3132 sont finalement terminés et mis à jour.

La recherche du ticket 3107 et son bug généré montre qu'il est plus complexe qu'il n'y paraît. Peut-être en poursuite le vendredi 25 avec de l'aide.

Problème de Pipeline (Merge Request qui a un mauvais test de qualité), il faudra rétablir ça au plus vite.

Pas besoin de capture d'écran aujourd'hui.

**Le 25 Mars 2022.**

A l'arrivée, correction directe du Pipeline qui semble fonctionner. Il suffisait de supprimer les fichiers et imports non utilisés. Ensuite faire un ng lint + ng rest, puis le git status, git add -A, git commit et git pull origin 'nom de ticket'.

En attente de nouveau ticket. Si temps libre, page résumée à faire aujourd'hui, et avancer sur la page accueil du Business case. Correction d'un léger souci de traduction pour donner suite à la découverte.

Nouveau ticket 3138 : Changer l'affichage du bouton "retirer" en mode edit.

Une capture d’écran a été effectuée.

**Le 28 Mars 2022.**

Fiche de résumé complétée et terminée. Je l'ai soumise à Human Booster et le retour fut positif. J'ai donc adopté la correction et retourné le tout.

J'ai mis ma correction de la librairie sur la branche alpha et mon pipeline à été validé.

Maintenant la merge est en attente d'acceptation, les tests sont en cours.

Demain si c'est validé il faudra mettre à jour la dépendance dans le package.json. Et puis faire un git add commit push depuis le front.

Ainsi sera clôturé le ticket 3138.

**Le 29 Mars 2022.**

Toujours en attente de la MR, elle est bonne sur les tests. Mais elle doit être vérifiée et validée. Une fois cela fait la MR front sera à faire, et puis la petite routine habituelle.

J'ai fait un peu de veille et commencé à faire la page boutique sur le fil rouge, la gestion du panier est délicate cependant.

**Le 30 Mars 2022.**

Assignation d'un ticket IAM-3156 : [N1] [Ajout d'une identité] Titre erroné.

Ce ticket avait été repéré et réparé il y'a quelque temps. Il sera donc corrigé avec la MR du ticket précédent le 3138.

La MR est validée, ticket 3138 et 3156 corrigés et mis à jour.

Enfin, dans le fil rouge, la première version de la boutique est faite avec son panier, il ne me reste qu'à la peaufiner.

**Le 31 Mars 2022.**

Création d'un component dashboard angular. Exercice de connexion Angular avec fichier d'exemple pour le business case. Amélioration de la boutique sur le business case. A utiliser avec des librairies JS pour faire des graphiques et données de mois.

**Le 1 Avril 2022.**

Aujourd'hui fut une journée calme, j'ai refait des exercices Angular pour comprendre les endpoints et modifié le business case.

La page boutique est presque finie à 100%.

La page de contact a été bien modifiée et l'utilisateur peut maintenant modifier ses informations. De nouvelles ressources sont disponibles, il est aussi possible de choisir une autre adresse et un pays.

La page d'admin avec la liste des USER est prête, il faudra la faire avec le dashboard angular.

**Le 4 Avril 2022.**

Continuer si possible le projet fil rouge. Avec mot de passe hashé à faire. Crud boutique. Gérer la facturation. Recentrer image + favicon. Dashboard.

Aujourd'hui à 14h30 réunion avec Charlotte. J'ai avancé sur le business case et terminé les hashages de mot de passe.

Ce week-end j'ai pu recentrer les images, faire le crud de la boutique. Il me reste la facturation dashboard et favicon. Tout le reste est amélioré, le contenu de certaines choses ou modifications requiert d'être connecté pour pouvoir procéder à l'action.

**Le 5 Avril 2022.**

Hier soir j'ai pu faire les gestions de rôle symfony. Cela fonctionne et me permet de masquer des zones ou des pages pour le simple utilisateur. Bloquer l’accès à certaines pages et options. Seul l'ADMIN peut faire ce que bon lui semble sur le site.

Aujourd'hui nouveau ticket : intégration d'un 'Stepper'.

Création du stepper qui sera un composant générique dans la libs de beys-software. On exporte des données de la beys-software-lib-sample vers beys-software-common-lib.

Cela permettra d'utiliser <beys-stepper></beys-stepper> qui invoquera ledit stepper.

Découverte du PowerPoint à rendre pour le 25 Avril, sera faite après le dossier projet, entre le 15 et 25.

**Le 6 Avril 2022.**

Poursuite du projet de faire un composant entier dans une librairie Angular.

Le stepper est finalement affiché en mode DUR. Il faut le rendre dynamique, avec des statuts. C'est du Angular pur. Input/output. Avec des NGIF NGFOR NGCONTAINER.

Poursuite de ce projet le lendemain.

**Le 7 Avril 2022.**

Aujourd'hui révision Angular. Création de petits projets. Une mini boutique dans un premier temps et un prototypage de dashboard afin de monter en compétences.

Le JIRA a été mis à jour de mes heures effectuées.

Création d'une mini boutique en Angular.

Nouvel exercice Angular qui simule une connexion à une base de donnée / API, poursuite le 8 Avril.

**8 Avril 2022.**

Début de réflexion sur le dossier projet d'Human Booster. Une petite moitié déjà effectuée.

Sprint Review 58 aujourd'hui.

Ticket d'amélioration d'interface passée en revue.

Ticket 3107 - Ticket 3138 - Ticket 3028 - Ticket 2964

Poursuite de la montée en compétence d'Angular. Des nouveaux exercices et parcours de documentation pour concevoir le stepper.

**11 Avril 2022.**

Ce week-end j’ai réussi à intégrer sur le projet fil rouge de nouvelles fonctionnalités. Une barre de recherche fonctionnelle, un bouton précédent, une boutique affichée par ordre alphabétique.

Poursuite de la veille sur Angular avec simulation d’appel à une API grâce à un tutoriel. On a un sprint refinement cet après-midi, pour définir les nouvelles tâches à venir, avec les tickets. Découverte d’un nouveau projet angular à faire en tutoriel, un dashboard qui permet de comprendre l’utilisation de charts, et à appliquer sur le fil rouge plus tard.

**PARTIE II - Projet fil rouge : La Nimes’Alerie**

Dans cette page j'ai établi les tâches à faire pour le site de la Nimes'Alerie.

Ceci est conçu pour gérer le temps et prioriser les tâches à effectuer.

Certaines pages ont été faites de mon propre chef pour construire un site plus étoffé et amusant à parcourir.

J'ai constaté qu'avec ce projet, on a sans cesse de nouvelles idées d'implantation, d'amélioration, je pense que ce ne sera jamais parfait.

On peut parcourir ce site en mode lecture, certaines rubriques ne seront accessibles que par compte, et certaines que si vous êtes l'administrateur de ce site.

Utilisation de technologies : HTML, CSS, JAVASCRIPT, SYMFONY, TWIG, PHP, MYSQL & DQL

Le site de la Nimes’Alerie a été conçu en parallèle, pendant le temps libre, en toute opportunité. Ci-dessous voici les grands points que j’ai pensé à lister et à finir qui sont parfaitement fonctionnels sur mon site.

Je me suis inspiré de ce que j’ai vu en stage pour hiérarchiser les tâches, et les ‘’qualifier’’ avec le « : effectué ».

- Création du projet Symfony de la Nimes’Alerie : effectué

- Fabrication de la base de données répondant aux entités du site : effectué

- Conception d'une top bar restante en haut de l'écran : effectué

- Implantation d'une barre complémentaire à cette top bar qui sera elle aussi fixée en haut : effectué

- Conception d'un footer pour encadrer le contenu du site : effectué

- Importation d'un favicon faisant une identité de site : effectué

- Création d'un formulaire de connexion utilisateur : effectué

- Création d'un formulaire de contact Nimes'Alerie : effectué

- Elaboration d'un CRUD dans les articles de rédaction blog, seul un 'ADMIN' pourra effacer du contenu) /ajout /post /edit /delete : effectué

- Création d'une page bonus cachée dans la barre de navigation : effectué

- Fabrication d'une page space base html/css type classique : effectué

- Création d'un forum grâce à un CRUD : effectué

- Alignement parfait des contenus avec la top bar et le footer : effectué

- Faire en sorte que le site de la Nimes'Alerie soit responsive et ajouter des corrections mineures : effectué

- Développement d'une page d'accueil faisant office de résumée sur le site : effectué

- Conception de la boutique, d'une page détail produit, d'un panier avec un calcul hors taxe (HT) et toute taxe (TTC) : effectué

- Permettre à l'utilisateur de modifier ses informations clients (e-mail, nom/prénom, adresse etc.…) : effectué

- Développement du hashage de mot de passe pour sécuriser la connexion utilisateur (client) : effectué

- Création de vrais produits en BDD (avec la boutique) : effectué

- Gestion de différent rôle propre au site de la Nimes'Alerie (gestionnaire = ADMIN, utilisateur = USER) : effectué

- Résolution d'un problème d'affichage dans la boutique et aussi dans la gestion du panier : effectué

- Possibilité pour le ["ROLE\_ADMIN"] d'ajouter, voir, éditer ou supprimer des produits de la boutique : effectué

- Possibilité pour le ["ROLE\_ADMIN"] d'accéder par la page 'La Nimes'Team' à la gestion des utilisateurs par un CRUD : effectué

- Une barre de recherche de produit fonctionnelle dans le header : effectué

- Mise en place de la sécurité d’authentification anti brute-force : effectué (A l’aide de composer require symfony/rate-limiter)

- Ajout d’un bouton précédent interne (dans certaines pages) et partout dynamiquement (à côté du bouton ‘’revenir en haut’’) : effectué

Il me reste à faire au moment où j’écris ce dossier :

* Un dashboard récupérant des statistiques sur les commandes clients.
* Une facturation qui s’effectue quand je valide un panier.

J’ai encore l’impression que le site peut être peaufiné, amélioré, avec toujours plus de nouvelles implémentations. D’autre part, je souhaite compléter ces deux points avec mon projet annexe, le fil rouge Human Booster. Ce sera dans la description du site de la Nimes’Alerie.

**DESCRIPTION DU SITE DE LA NIMES’ALERIE**

**• Présentation du jeu d’essai élaboré par le candidat de la fonctionnalité la plus représentative (données en entrée, données attendues, données obtenues)**

N’ayant pas eu l’autorisation de vous montrer IAM\_FRONT, je vais vous présenter le site de la Nimes’Alerie. Les fonctions de la boutique, les différentes pages, ce site est assez complet, reprend beaucoup de choses vus en formation est de plus responsive. Ci-dessous une description des rubriques.

**Présentation du fil rouge : La Nimes’Alerie.**

J'ai commencé par étudier la demande du client, un site boutique utilisant deux framework : Angular pour le côté Dashboarding et Symfony pour pratiquement tout le reste.

Angular acheminera donc une connexion à une API, pour faire un Dashboard. Symfony lui servira à afficher des informations dans des vues (twig) les controller respectifs. Le site est découpé en plusieurs sections qui seront reliées par une barre de navigation, et le tout sera fermée par un footer avec des informations génériques.

Nous avons donc :

* La barre de navigation avec les rubriques, les liens ou aller, un compteur de prochain lancement. Une zone de recherche est incluse dedans.
* Le footer lui vous présentera les moyens de paiement, les coordonnées, la bannière human booster.
* Des boutons magiques en bas à droite pour vous aider à naviguer, précédent, et retour en haut de la page.
* Une page de connexion avec un formulaire à remplir, le client renseigne ses coordonnées, son adresse, nom, prénom, email, etc...
* Une page d’accueil qui reprendra des éléments récents. Cette page est la page d’arrivée pour l’utilisateur, elle présente brièvement le site, ses fonctions.
* La Nimes’Team, un formulaire de contact, et une section qui permet d’éditer vos informations de compte, et d’aller sur la page ADMIN si vous avez les accréditations.
* Une page de boutique avec les différents articles, détails, photos, et panier. Avec cette page vous pouvez procéder à vos achats. En ajoutant des produits, les supprimant, modifier un contenu du panier, et calcul de la TVA inclus !
* La page Article (de rédaction), ici vous retrouverez facilement des sujets de rédaction fait par la communauté, des informations, sur la vie du site.
* La base terrestre, qui est un forum communautaire entre utilisateur de la Nimes’alerie, grâce à cela, vous pouvez parler, débattre, créer des sujets passionnants.
* La pension spatiale permettant d'envoyer vos animaux dans l'espace (en toute sécurité bien sûr).
* Une page de bonus à été faite, elle est cachée dans le header, et permet de jouer à un mini jeu en JavaScript pur, ce ne fut pas simple à intégrer.
* La page admin est protégée par une accréditation. Et si vous tentez d’y accéder par une url, cela ne sera tout simplement pas possible.
* La sécurité anti brute force est aussi présente, ainsi que le hashage des mots de passe. Seuls les admins pourront modifier certaines choses comme l’édition de la boutique.

Pourquoi avoir fait ce site ?

**• Réalisations du candidat comportant les extraits de code les plus significatifs et en les argumentant, y compris pour la sécurité et le web mobile**

**• Bilan de la formation l’apprentissage.**

**• Les captures d’écran.**

Les captures d’écran seront celles du projet fil rouge ici.