

Mise en pratique Angular, .Net et MongoDB

L'objectif de cet exercice est de valider les connaissances en .Net, découvrir MongoDB et consolider les connaissances Angular.

L'application

L'application permettra d'écrire, publier, gérer et consulter des actualités entre utilisateurs.

Un utilisateur est décrit par :

- un nom
- un prenom
- un profil appartenant à l'ensemble {**eleve**, **professeur**, **direction**}
- un login unique
- un mot de passe

Une actualité contient :

- un titre
- un ou plusieurs profils de diffusion
- une description
- un image
- une date de publication
- une date d'expiration
- une information sur le créateur de l'information

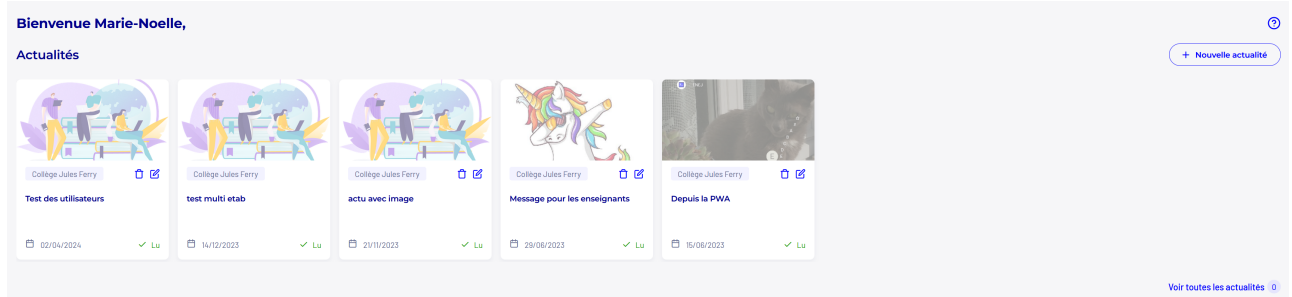
Les impressions d'écran sont extraites d'un logiciel disponible en production et données à titre indicatif pour l'aspect graphique. Plusieurs fonctionnalités et informations y sont présentes mais ne sont pas demandées dans cet exercice. On souhaite juste tester le cas nominal d'utilisation, il ne s'agit en aucun cas de réaliser une application prête pour la production.

L'application de type SPA, contiendra plusieurs écrans :

1. un premier écran de connexion (très simple)

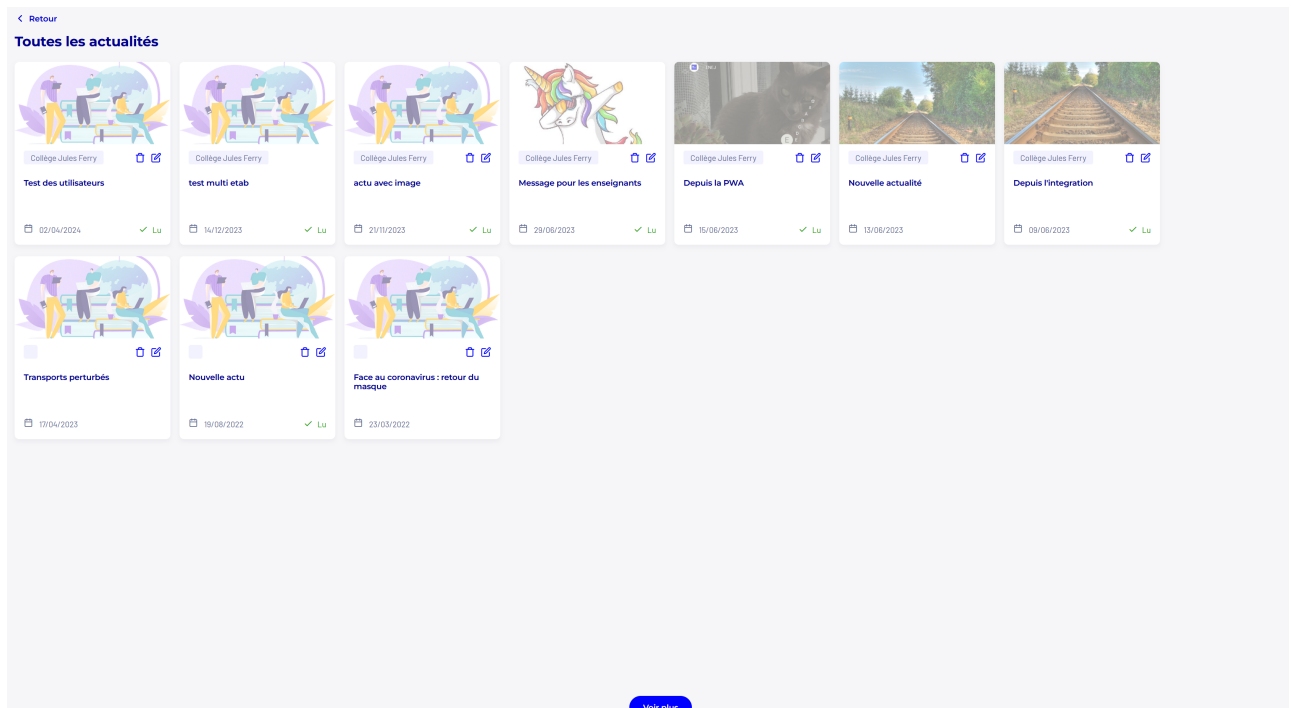
- l'utilisateur découvre l'application par ce premier écran
- il rentre son login et un mot de passe (qui peut être enregistré en base de données)
- il est alors redirigé vers son dashboard

2. un écran 'dashboard':



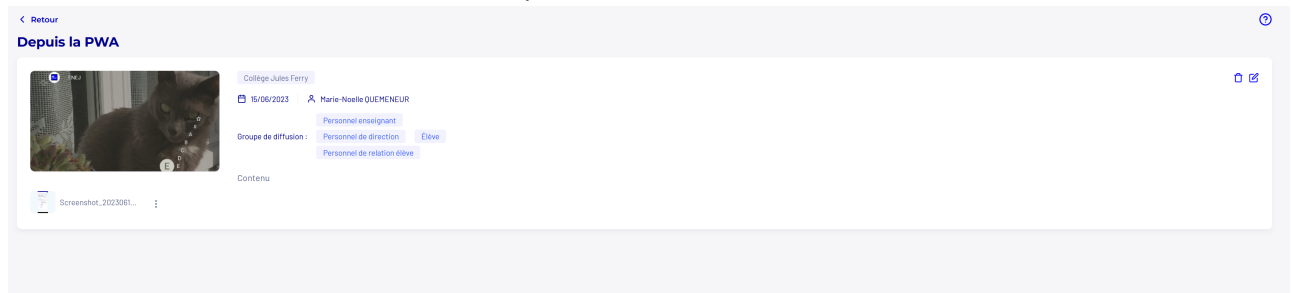
- chacun a accès à son dashboard
- les 5 premières actualités le concernant sont affichées sur ce dashboard
- il peut accéder à une écran pour voir toutes les actualités qui le concernent
- pour les profils de direction, un autre bouton est affiché permettant de créer une nouvelle actualité

3. un écran listant les actualités



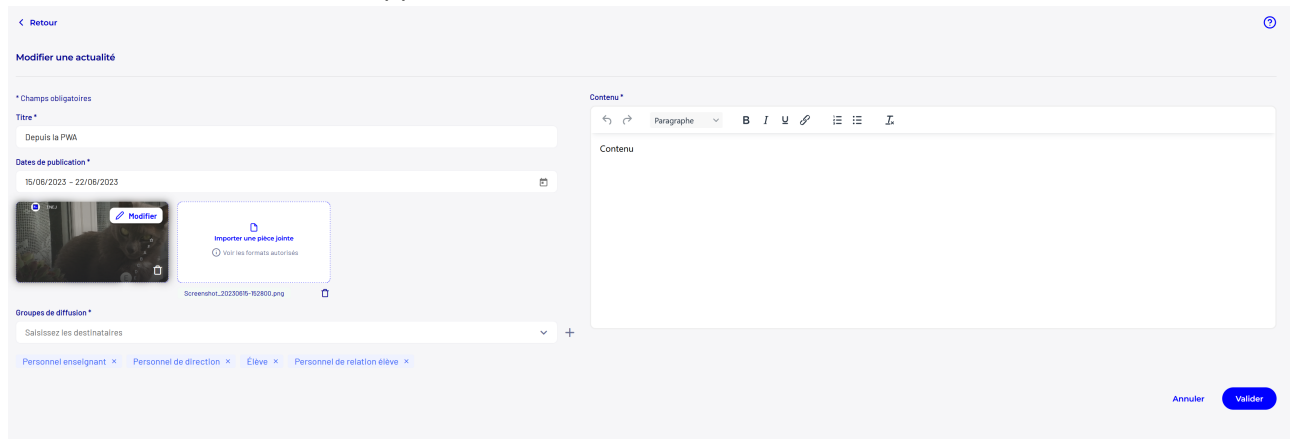
- chaque utilisateur peut voir les actualités filtrées selon son profil
- les 5 premières sont affichées, les 5 suivantes sont affichées au clic sur un bouton jusqu'à épuisement des actualités.
- les actualités sont ordonnées par ordre décroissant de date de publication

4. un écran de consultation d'une actualité après un clic sur la card associée



- tout le monde peut consulter les détails d'une actualité qui le concerne

5. un écran de modification et suppression des actualités



- seuls les profils de direction peuvent créer et modifier les actualités
- seul le créateur d'une actualité peut la supprimer

Scénarios attendus

- une personne de n'importe quel profil se connecte, retrouve son tableau de bord et les actions qui lui sont autorisées
- une personne de profil direction peut écrire une dizaine d'actualités avec n'importe quelle combinaison parmi les profils annoncés
- un professeur, élève ou directeur consulte les actualités dont il est destinataire
- les profils de direction modifient les actualités
- un créateur d'actualité supprime des actualités

Bonus

- les actualités peuvent être marquées comme lues par les utilisateurs
- les actualités peuvent contenir des pièces jointes
- la connexion se fait de manière sécurisée à l'aide d'un IDP. Le mot de passe n'est plus stocké en base de données.
- on suppose que la liste des actualités devient obsolète au bout de 10 minutes et doit être renouvelée

Performance attendue

- l'application doit afficher les actualités de l'utilisateur de manière transparente pour l'utilisateur.
- l'application doit effectuer un nombre de requêtes raisonnables pour la récupération des données
- l'application doit éviter les requêtes inutiles au back pendant la navigation

Rendus

- un schéma json des différentes structures de données en base de données
- le code source de l'application front
- le code source des apis
- une démonstration rapide

Indications

Générales

- tout choix d'architecture doit être justifié
- si un scénario ou un cas d'utilisation n'est pas clair, il est conseillé de demander des précisions dans la mesure du raisonnable.

Base de données

- la base de données s'appuiera sur MongoDB
- nous ferons attention aux structures des données et à l'architecture générale.

API

- les APIs s'appuieront sur le framework .NET
- nous ferons attention à l'architecture en générale, aux anti-patterns et à la lisibilité du code

Front

- l'aspect graphique ne doit pas correspondre à 100%, un effort sur l'interface sera tout de même apprécié
- l'utilisation de la bibliothèque Angular Material est fortement conseillée
- nous ferons attention au respect des paradigmes et l'utilisation des outils Angular