

Akiba Town

Dossier de Projet Professionnel



**François Niang - Ibrahim Sylla - Hugo Chabert -
Thibault Kine**

2022/2023

CDA

Table des matières

Compétences du référentiel couvertes par le projet.....	2
Résumé du projet et de son but.....	3
Arborescence du site.....	4
Conception du projet.....	5
Choix des techniques.....	5
Mise en place de l'environnement.....	8
Réalisation.....	10
Charte graphique.....	10
Cahier des charges.....	10
Maquette.....	10
Conception de la Base de Données.....	10
Répartition des tâches.....	11
Développement du projet.....	12
Front-End.....	12
Spécifications fonctionnelles.....	12
Spécifications techniques.....	13
Back-End.....	13
Création d'API et Récupération d'API externes.....	13
Sécurité.....	13
Back Office.....	14
Web Application.....	14
Organisation Hiérarchique.....	15
Annexes.....	17

Compétences du référentiel couvertes par le projet

1. Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité :

Maquetter une application

Développer une interface utilisateur de type desktop

Développer des composants d'accès aux données

Développer la partie front-end d'une interface utilisateur web

Développer la partie back-end d'une interface utilisateur web

2. Concevoir et développer la persistance des données en intégrant les recommandations de sécurité :

Concevoir une base de données

Mettre en place une base de données

Développer des composants dans le langage d'une base de données

3. Concevoir et développer une application multicouches répartie en intégrant les recommandations de sécurité :

Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement

Concevoir une application

Développer des composants métier

Construire une application organisée en couches

Développer une application mobile

Préparer et exécuter les plans de tests d'une application

Préparer et exécuter le déploiement d'une application

Résumé du projet et de son but

Dans le cadre de la formation de développeur et concepteur d'application, et la présentation de fin d'année, le projet regroupe toutes les compétences que j'ai pu acquérir durant mon cursus. Le projet est un projet Forum développé sous forme d'application mobile. Ce projet semblait très complet aussi bien dans la conception que dans le développement. En effet, lors de ce projet, la partie conception fut la plus importante dans le sens où mon équipe et moi avons constaté tous les nouveaux paramètres que nous devons prendre en compte pour aboutir à un résultat convenable.

Akiba Town est une application de Forum (questions/réponses) principalement concentrée sur l'univers japonais, plus précisément le manga. Le but de l'application est, comme son nom et le slogan le disent, de retrouver le célèbre quartier d'Akiba au Japon directement sur son téléphone. Par le biais de cette application, les utilisateurs pourront créer divers sujets classés par catégorie de manga pour débattre, donner leur avis et plus généralement interagir avec la communauté de l'application.

Nous avons décidé avec mon équipe de réaliser ce projet car il n'existe pas encore d'application réunissant cette communauté autour du manga et de l'univers japonais.

L'application sera réalisée en français, et cette dernière devra être accessible sur mobile en temps qu'application et une partie accessible sur d'autres appareils tels que Tablette, ou Desktop selon certains critères (Modérateur ou Administrateur).

Cette application ciblera une communauté de passionnés de l'univers Japonais, ses utilisateurs seront acceptés selon un critère qui sera vérifié lors de la création du compte : l'âge. En effet, certaines œuvres et sujets de l'application pourront être catégorisés Public averti. Cet outil devra être totalement contrôlé par un administrateur.

Arborescence du site

En ce qui concerne l'arborescence de l'application, elle se décline comme ceci (voir annexe) :

- Landing page
- Page de connexion
- Page d'inscription
- Page d'accueil
- Page Mon Profil
- Page Profil Utilisateur
- Page Equipes
- Page Rechercher
- Page Thèmes
- Page Forum
- Page Détails de Sujet
- Page Création de Sujet
- Page Activités

Une partie back-office est également prévue afin de permettre la gestion du site :

- Page Gestion des utilisateurs (Administrateur)
- Page Gestion des Sujets (Administrateur)
- Page Statistiques (Administrateur)
- Page Gestion des Commentaires (Administrateur)
- Page Gestion des Sujets (Modérateur)
- Page Gestion des commentaires (Modérateur)

Conception du projet

Compte tenu de la taille du projet et de la complexité de ce dernier dûe aux nouvelles techniques acquises lors de mon apprentissage, la phase de conception est un élément décisif au bon déroulement du projet. Avoir un “fil rouge” est le meilleur moyen pour réussir et mener à bien le projet et obtenir un résultat convenable et fonctionnel.

Lors de cette phase de conceptualisation nous avons, mon groupe et moi, “brainstormé” sur toutes les fonctionnalités intéressantes que l’application pourrait avoir et comment les mettre en places, comment générer dynamiquement les catégories par lesquelles nous allions classer les sujets, et allions nous, pour ça, utiliser des outils externes comme une API répertoriant une liste de tous les mangas existant.

Nous avons aussi fait un travail de documentation avant de nous lancer pour bien comprendre et avoir un aperçu global sur les technos que nous allions utiliser pour le développement de notre application.

Choix des techniques

La première partie de la phase de conception a été la décision des technos et outils que nous allions utiliser pour réaliser notre application. Dans un premier temps, nous nous sommes concentrés sur les éléments plutôt externes (outils de conceptions) et par la suite nous nous sommes attaqués à la partie plus interne (environnement de travail, technos utilisées, versionning).

Pour la partie externe nous avons décidés, lors de notre réunion de groupe, de nous servir de :

- **Figma** : Qui est un outil de conception d'interface utilisateur (UI) et d'expérience utilisateur (UX) en ligne. Il permet aux designers de créer des maquettes, des prototypes et des designs interactifs en temps réel, et de collaborer avec d'autres membres de l'équipe en temps réel. Figma est également connu pour ses fonctionnalités de collaboration en temps réel, ce qui signifie que plusieurs membres de l'équipe peuvent travailler sur un même projet simultanément, ce qui facilite la collaboration et la communication. Nous avons réalisé grâce à cette dernière les maquettes de notre application et avons réussi à la rendre en quelque

sorte interactive grâce aux nombreux avantages de Figma.

- **LucidChart** : Qui est une application de création de diagrammes en ligne, utilisée pour créer des organigrammes, des diagrammes de flux, des diagrammes UML, des diagrammes de réseau, des diagrammes de processus métier, des maquettes de conception de sites web et d'autres types de diagrammes et de graphiques. Lucidchart est utilisée par des professionnels de différents domaines tels que la technologie, le marketing, la finance, la santé, l'éducation et bien d'autres. L'application offre une variété de fonctionnalités, notamment la possibilité de collaborer en temps réel avec des collègues, de partager des diagrammes avec des tiers, d'importer des données à partir de feuilles de calcul et d'autres sources de données, et d'intégrer des diagrammes dans d'autres applications et services en ligne. Lucidchart est conçu pour être facile à utiliser, avec une interface utilisateur intuitive et une large gamme de modèles prédéfinis et de formes personnalisables pour créer rapidement des diagrammes professionnels. L'application est accessible via un navigateur web ou une application de bureau, et offre également une application mobile pour iOS et Android. Dans notre cas nous avons utilisée cette dernière pour la conception de nos MCD et MLD

Pour la partie plus interne du projet nous avons décidé de nous concentrer autour de :

- **Github** : Qui est une plateforme de développement de logiciels qui permet aux développeurs de stocker, de gérer et de partager leur code source avec d'autres développeurs. Il utilise le système de contrôle de version Git, qui permet de suivre les modifications apportées au code source au fil du temps. Les développeurs peuvent utiliser Github pour collaborer sur des projets, partager des bibliothèques de code, contribuer à des projets open source, signaler des problèmes et proposer des modifications de code. Github offre également des fonctionnalités de suivi des problèmes, de gestion de projet et de documentation, ce qui en fait une plateforme complète pour le développement en tous genres. Nous avons utilisé cette plateforme pour le versionning de nos codes.
- **Github Desktop** : GitHub Desktop est une application de bureau pour les développeurs qui facilite l'utilisation de la plateforme de développement collaboratif GitHub. Elle fournit une interface graphique pour accéder aux fonctionnalités de Git, un système de contrôle de version de logiciels largement utilisé dans l'industrie du développement de logiciels. Avec GitHub Desktop, les développeurs peuvent cloner des dépôts de code source à partir de GitHub, créer des branches, effectuer des modifications de code, committer des changements, créer des demandes de pull, synchroniser les branches avec les dépôts distants et

effectuer d'autres opérations Git courantes, le tout à partir d'une interface graphique simple et conviviale. En somme, GitHub Desktop est un outil de gestion de version de code qui facilite la collaboration et la gestion de projets de développement de logiciels en équipe.

- **React Native** pour la partie Front-End : Qui est un framework open source de développement d'applications mobiles qui permet de créer des applications mobiles multiplateformes pour iOS et Android en utilisant JavaScript et React, une bibliothèque JavaScript pour la création d'interfaces utilisateur. React Native permet aux développeurs de créer des applications mobiles avec des performances élevées et des fonctionnalités natives, tout en utilisant des technologies web familières telles que JavaScript, HTML et CSS. Le framework utilise une approche de développement hybride, qui combine des éléments natifs de chaque plateforme mobile avec une logique de code JavaScript partagée entre les plateformes. Les applications React Native peuvent être développées à partir d'une seule base de code, ce qui réduit considérablement le temps et les coûts de développement. Les développeurs peuvent également utiliser des bibliothèques et des modules tiers pour ajouter des fonctionnalités supplémentaires à leurs applications. React Native est devenu très populaire dans le développement d'applications mobiles en raison de sa facilité d'utilisation, de sa grande communauté de développeurs, de sa prise en charge de la réutilisation de code et de sa capacité à fournir des performances natives pour les applications mobiles multiplateformes.
- **Node Js** : Node.js est un environnement d'exécution de code JavaScript côté serveur, open source et multiplateforme, qui permet d'exécuter des applications JavaScript en dehors d'un navigateur web. Node.js utilise le moteur JavaScript V8 de Google, le même moteur qui est utilisé par le navigateur web Google Chrome, pour exécuter le code JavaScript. Node.js est souvent utilisé pour la création de serveurs web, de services web, d'API, de microservices, de chatbots et d'autres applications côté serveur. Il est également utilisé pour la construction d'applications de bureau et de jeux vidéo. Node.js permet aux développeurs de créer des applications évolutives et hautement performantes avec une consommation de ressources minimale. Node.js est particulièrement adapté aux applications en temps réel qui nécessitent des mises à jour rapides et des connexions permanentes avec le client, comme les applications de chat, les jeux multijoueurs, les systèmes de streaming en direct, etc. Node.js possède également une riche collection de modules et de packages open source, disponibles via le gestionnaire de packages npm, qui facilite le développement et l'intégration de

nombreuses fonctionnalités dans les applications Node.js. Dans notre cas nous l'utiliserons pour la création d'une API pour le Back-end de notre application.

- **Sequelize** : qui est une bibliothèque open source pour Node.js qui facilite l'interaction avec des bases de données relationnelles telles que MySQL, PostgreSQL, SQLite et Microsoft SQL Server. Sequelize fournit un ORM (Object-Relational Mapping), qui permet aux développeurs de manipuler des données relationnelles en utilisant des objets JavaScript plutôt que du SQL brut. Avec Sequelize, les développeurs peuvent définir des modèles de données en utilisant des classes JavaScript, qui sont ensuite mappées à des tables dans la base de données. Les développeurs peuvent utiliser ces modèles pour effectuer des opérations de création, de lecture, de mise à jour et de suppression (CRUD) sur les données, ainsi que pour gérer les relations entre les tables de la base de données. Sequelize fournit également des fonctionnalités avancées telles que la validation des données, le support des transactions, le contrôle des conflits de concurrence et la migration de schéma de base de données. En somme, Sequelize facilite la manipulation de bases de données relationnelles en utilisant JavaScript et fournit une interface intuitive pour l'interaction avec la base de données, réduisant ainsi le temps et les efforts nécessaires pour développer des applications Node.js avec des bases de données relationnelles.

Mise en place de l'environnement

Après nous être mis d'accord sur tous les outils et technos que nous allions utiliser nous avons décidé de mettre en place l'environnement de travail pour ne pas perdre plus de temps étant donné que le projet était plutôt imposant. Pour se faire nous avons décidé de travailler sur l'IDE **Visual Studio Code**, qui nous a permis de mettre en place un environnement de travail convenable et facile à comprendre.

Par la suite nous avons dû réaliser les téléchargement nécessaires au bon fonctionnement de l'application c'est à dire :

- **Node JS** : que nous avons téléchargé de cette manière :
 1. Aller sur le site officiel de Node.js à l'adresse <https://nodejs.org/>
 2. Cliquer sur le bouton "Download" situé sur la page d'accueil.
 3. Sur la page de téléchargement, les options pour télécharger la version actuelle de Node.js ou la version LTS (Long Term Support) stable seront affichées. Utiliser la version LTS stable pour une

utilisation en production.

4. Sélectionner le système d'exploitation (Windows, macOS ou Linux) et cliquer sur le bouton de téléchargement correspondant.
5. Une fois le téléchargement terminé, installer Node.js en suivant les instructions d'installation affichées.

Une fois l'installation terminée, nous pouvons vérifier si Node.js est correctement installé en ouvrant une fenêtre de terminal (ou de ligne de commande) et en tapant "node -v". Cela affichera la version de Node.js installée sur le système.

- Pour télécharger React Native, nous pouvons suivre les étapes suivantes :
 - Tout d'abord, nous devons nous assurer que nous avons Node.js installé.
 - Ensuite, nous pouvons ouvrir une fenêtre de terminal (ou de ligne de commande) et installer Expo CLI en utilisant la commande suivante :
 - **npm install -g expo-cli**

Cette commande installe Expo CLI de manière globale, ce qui nous permet de créer et de gérer des projets React Native.

- Maintenant que nous avons installé Expo CLI, nous pouvons créer un nouveau projet React Native en utilisant la commande suivante :
- **expo init Akiba Town**
- Nous pouvons ensuite sélectionner le template de projet que nous souhaitons utiliser en suivant les instructions à l'écran.
- Enfin, nous pouvons ouvrir un navigateur web et accéder à l'URL affichée dans la fenêtre de terminal pour accéder à l'interface de développement d'Expo. Nous pouvons maintenant éditer le code source de notre projet React Native et voir les changements en temps réel dans notre application.
- Ces étapes nous permettront de télécharger React Native et de commencer à développer des applications mobiles multiplateformes en utilisant JavaScript et React.

Réalisation

Charte graphique

Pour la conception de ce projet, lors de la réunion de brainstorming nous avons décidé de réaliser une charte pour définir l'identité visuelle de l'application et tout ce qui tourne autour. Cette charte graphique comprend (voir annexe) :

- Logo
- Couleurs
- Polices (Taille et style)
- Icônes

Cahier des charges

Le cahier des charges du projet nous a été fourni par l'école qui nous a guidé dans les attentes de l'application, dans un sujet où toutes les fonctionnalités du site sont détaillées. (voir annexe)

Maquette

Lors de la conception de l'application, la maquette fut une étape très importante. Grâce à **Figma**, nous avons pu réaliser une maquette dynamique et interactive permettant de proposer un prototype sur lequel nous pouvons naviguer comme sur l'application finale.

Conception de la Base de Données

La conception de la base de données est une partie cruciale de la conception. Nous l'avons réalisée à l'aide de la méthode **MERISE**, consistant à modéliser, analyser et concevoir les systèmes d'information en se concentrant sur les aspects structuraux et fonctionnels.

Nous avons donc pour cela réalisé un **Modèle Conceptuel de Données (MCD)** et un **Modèle Logique de Données (MLD)** qui résume la structure et toutes les spécificités de notre base de données. (Voir annexe)

Répartition des tâches

Pour ce projet nous avons décidé de séparer le projet en 3 parties (Voir annexe):

- **Front-end** : Dans cette partie on travaille le design et l’affichage de l’application pour ensuite permettre d’avoir un aperçu sur notre travail lorsque l’on travaille sur la partie backend
- **Back-End** : Dans cette partie nous développons la partie API et Base de données de l’application, création de compte, connexion, création de post, réaction aux post etc...
- **Gestion** : Dans cette dernière partie nous regroupons toutes les interactions que nous avons avec les versions du projet, c'est-à-dire toutes les actions via Git et Github.

Développement du projet

Front-End

Comme résumé précédemment, la partie frontend nous sert ici de repère visuel, nous devons donc nous assurer de reproduire le plus fidèlement possible la maquette.

Spécifications fonctionnelles

- **Création de compte et identification** : Comptant une multitude d'utilisateurs, ils seront en possibilité de créer un compte s'ils n'en ont pas et par la suite se connecter pour accéder à l'application pouvant ensuite réagir à des posts, créer des post, consulter et modifier leurs profils.
- **Création d'un Post** : Le but de cette application est de créer des post pour permettre au reste des utilisateurs d'y réagir. Un espace y est consacré, on y retrouve un formulaire contenant:
 - un Input **Titre du sujet** dans lequel l'utilisateur devra renseigner le nom qu'il donnera au post
 - un Select **Choix du manga** qui permettra de renseigner le manga sur lequel se porte le post pour faciliter la recherche par la suite
 - Et enfin un Textarea **Description du sujet**, qui résume brièvement le sujet. L'utilisateur sera libre de poser ses questions ou exprimer son avis.
- **Réaction au post** : Rappelons le, le but de l'application est de réunir une communauté basée sur l'amour du manga et de l'univers japonais, il serait donc logique de pouvoir réagir aux différents posts pour donner son avis et débattre avec le reste de la communauté. Dans un espace dédié au post, une rubrique commentaire est disponible, contenant un input permettant d'écrire un commentaire. Dans le cas où l'utilisateur apprécie le post, il peut **aimer le post**.
- **Modification des informations personnelles** : Après s'être connecté, l'utilisateur a accès à l'onglet Mon profil où il pourra par la suite avoir un aperçu de ses informations personnelles et s'il le souhaite les modifier à sa guise. En plus de cela, il aura un récapitulatif de ses posts et de ses commentaires dans sa section profil.

Spécifications techniques

Etant donné que ce projet est notre première application, nous avons dû nous adapter pour pouvoir acquérir de nouvelles compétences techniques.

- **La Navigation** : Pour pouvoir se déplacer parmi les pages de notre application, nous avons besoin d'une des librairie les plus utilisée en React Native: la librairie React Navigation. Pourquoi avoir utilisé cette librairie ? Car comme dit juste avant, c'est une des plus utilisée en React Native ce qui veut donc dire qu'elle est beaucoup documentée ce qui est quelque chose qui nous a beaucoup été utile quand nous avons un problème. En plus de ça, cette librairie est, quand on comprend son fonctionnement, très simple d'utilisation.
- **Liaison Front/Back** : La liaison entre le Front-End et le Back-End, à été faite avec Sequelize qui, comme nous l'avons expliqué plus haut, est une bibliothèque open-source de Node.js qui nous permet d'interagir avec les données du Back-End, pour ensuite transmettre ces données au Front-End
- **Expo** : Pendant la création de notre projet, nous avons besoin de quelque chose pour pouvoir voir l'avancement de l'application mobile directement sur téléphone. Nous avons donc utilisé Expo qui non seulement nous permet de tester l'application instantanément sur nos téléphones, mais aussi de faciliter l'installation et la configuration initiale du projet.

Back-End

Notre partie Back-End est primordiale, car c'est elle qui nous permet de faire fonctionner l'application.

Création d'API et Récupération d'API externes

- **Qu'est-ce qu'une API ?** : Avant tout, rappelons ce qu'est une API. Une API est une interface logicielle qui permet de connecter un logiciel ou un service à un autre afin d'échanger des données.
- **Son utilité dans notre projet** : Créer une API nous donne plus de flexibilité et nous aide à personnaliser notre application tout en permettant l'intégration de solutions tierces.
- **Quelle est notre API ?** : Nous avons créé une API très simple, elle nous permet de connecter notre projet à une base de données que nous avons créée au début du

projet.

- **API Externe** : Nous avons besoin d'une base de données regroupant énormément de noms d'animés. Nous avons donc trouvé sur internet une API public qui a donc été utilisée pour notre application.

Sécurité

- **JWT** : Pour sécuriser notre application, nous avons utilisé la technologie JWT (JSON Web Token) qui nous permet d'échanger avec sécurité des jetons avec l'utilisateur.
- **Rôles** : L'utilisation de rôles dans les sites web et applications mobiles, est quelque chose de très commun et même primordial car cela nous permet de créer une sorte d'architecture entre les utilisateurs. Et plus l'utilisateur est de confiance, plus il aura un rôle important, ce qui lui permettra d'accéder à certaines pages inaccessibles par un utilisateur lambda.

Back Office

Web Application

Pour la partie Back office de l'application, nous avons réalisé une WebApp pour la gestion de l'application.

le choix de la WebApp pour la partie back office est un parti proposant de nombreux avantages :

- **Accessibilité** : Une WebApp permet aux utilisateurs d'accéder au Back Office depuis n'importe quel navigateur web, sans nécessiter l'installation d'un logiciel spécifique sur leur ordinateur. Cela permet aux administrateurs et aux gestionnaires de se connecter et de gérer les données et les fonctionnalités du système à partir de n'importe quel appareil disposant d'une connexion Internet.
- **Interface utilisateur conviviale** : Avec une WebApp, vous pouvez concevoir une interface utilisateur intuitive et conviviale pour faciliter la gestion des tâches dans le BackOffice. Vous pouvez utiliser des frameworks de développement web modernes pour créer des interfaces réactives et interactives, offrant ainsi une

expérience utilisateur agréable.

- Mises à jour centralisées : En utilisant une WebApp, vous pouvez mettre à jour et déployer de nouvelles fonctionnalités ou des corrections de bugs de manière centralisée. Les utilisateurs n'ont pas besoin de télécharger et d'installer des mises à jour manuellement, car les modifications sont immédiatement accessibles via le navigateur.
- Sécurité : Une WebApp permet de centraliser la gestion de la sécurité. Vous pouvez mettre en place des mécanismes d'authentification et de contrôle d'accès pour protéger les données sensibles du Back Office. De plus, les mises à jour de sécurité peuvent être appliquées rapidement et facilement pour garantir la protection du système.
- Intégration facilitée : Une WebApp facilite l'intégration avec d'autres systèmes et services. Vous pouvez utiliser des API (interfaces de programmation d'applications) pour communiquer avec d'autres applications, échanger des données et automatiser des processus.
- Évolutivité : En développant une WebApp, vous avez la flexibilité nécessaire pour faire évoluer votre BackOffice en fonction des besoins changeants de votre entreprise. Vous pouvez ajouter de nouvelles fonctionnalités et adapter l'application aux exigences spécifiques de votre organisation.

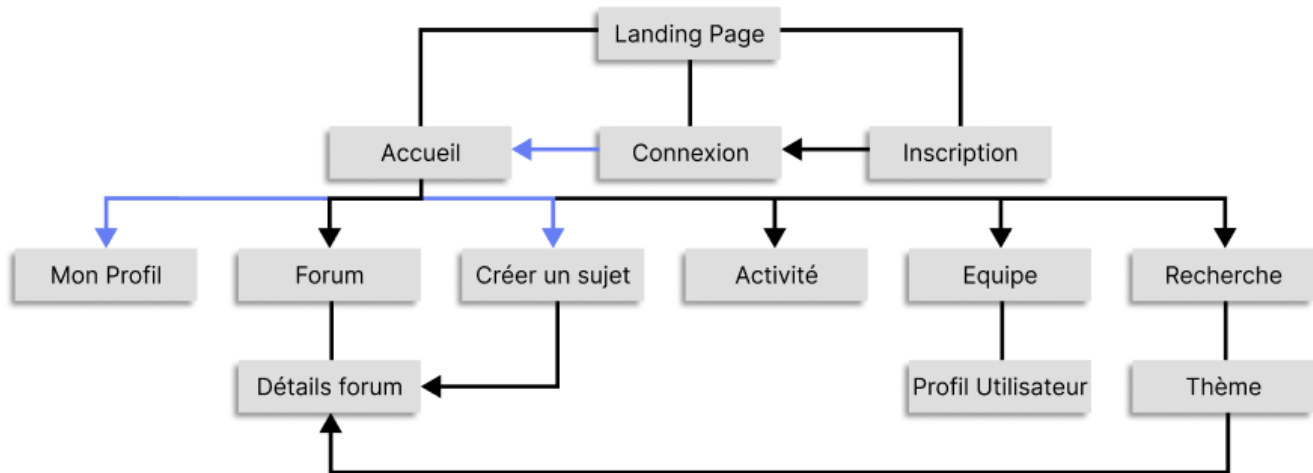
En résumé, une WebApp pour la partie Back Office offre une accessibilité accrue, une interface utilisateur conviviale, une gestion centralisée des mises à jour, une sécurité renforcée, une intégration facile avec d'autres systèmes et une évolutivité pour répondre aux besoins de votre entreprise.

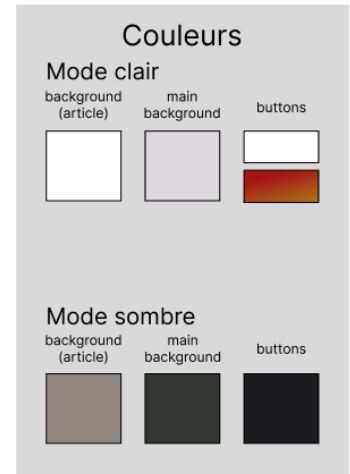
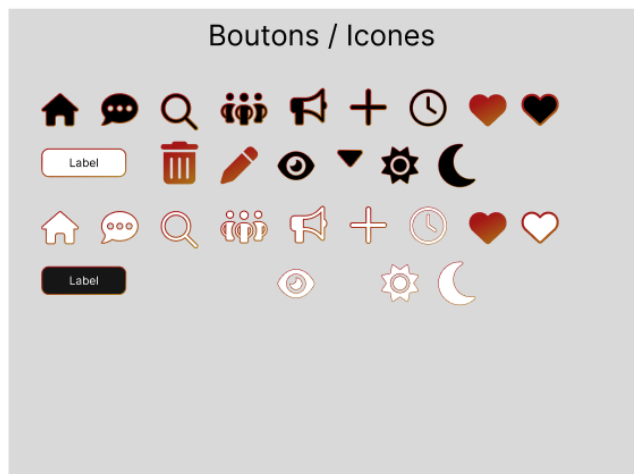
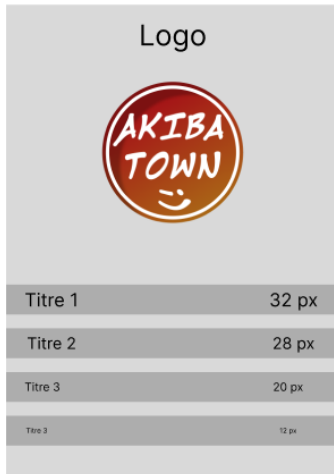
Organisation Hiérarchique

Dans notre projet, nous avons décidé de différencier les utilisateurs :

- **Les Utilisateurs non connectés** : Utilisateur lambda pouvant naviguer librement sur le site sans possibilité d'interactions (like, commentaire, création de post, visualisation de son profil...)
- **Les Utilisateurs connectés** : Utilisateur ayant un accès totale sur l'application avec la possibilité d'interagir avec le reste de la communauté (like, commentaire, création de post, visualisation de son profil...). Pour cela il faut qu'il dispose évidemment d'un compte existant, dans le cas contraire il devra s'en créer un.
- **Les Modérateurs** : Utilisateur ayant accès total sur l'application et la possibilité d'interagir, possibilité d'accéder partiellement à la WebApp réservée au Back Office, les modérateurs ont uniquement un pouvoir sur les posts. Ils ont accès à un historique des posts, pouvant les supprimer, supprimer des commentaires selon un motif bien défini.
- **Les Administrateurs** : Utilisateur ayant accès à l'intégralité de l'application et la possibilité d'interaction. Accès Total de la WebApp permettant ainsi la gestion entière de l'application via un historique de posts pouvant les supprimer, supprimer des commentaires selon un motif bien défini mais aussi un historique d'utilisateurs permettant la modification, la suppression ou la visualisation des utilisateurs ayant créé un compte. Mais aussi une partie statistiques chiffrant l'application depuis sa création.

ARCHITECTURE





❖ Text + label



❖ Input + label



❖ Entete



❖ Input + label



❖ bout...



❖ NAVBAR



❖ iPhoneHeader

9:41





Application Forum

Job 00 : Introduction

Dans le cadre de votre année, vous allez devoir réaliser une application mobile en React Native. Dans ce sujet, le but est de réaliser une app de chat. Il vous sera possible de vous créer un profil, d'y modifier des informations, et d'envoyer des messages sur un channel public, commun.

Un panel admin sera demandé afin de gérer les users, les sujets du forum, et les commentaires aux sujets.

Pour ce projet, il vous est recommandé d'utiliser API Platform pour le back. Node JS est cependant autorisé.

Ce projet a pour but de remplir la plus grande partie des compétences demandées pour le titre de Concepteur Développeur d'Application. Aussi n'oubliez pas ces points importants :

- La gestion de projet est une compétence importante du référentiel. Pensez à utiliser des outils tels que Trello, à faire des diagrammes de Gantt, et à bien utiliser Git et github de façon professionnelle
- La conception est aussi une partie importante. Vous vous devrez donc de rendre des maquettes professionnelles.

Maquette

Avant de vous lancer dans le dev de votre application, il est important de maquetter ses différentes pages, et de faire une modélisation de votre Base de Données avec un **MCD** et un **MLD**.

Dans un fichier **Maquette.pdf** et à l'aide d'un outil comme Figma ou Adobe XD, réalisez une maquette des différentes pages de votre App.

Dans un autre fichier **BDD.pdf**, réalisez le MCD et MLD de votre BDD. Ce seront les modèles les plus aboutis possibles. Si des changements ont lieu durant le développement de votre application, vous devrez mettre à jour ce fichier (il serait intéressant de faire une version Old-BDD.pdf, représentant votre MCD et MLD avant le dev, puis une version BDD.pdf représentant vos deux modèles à leur version finale, après le dev).

API

Pour ce projet, le choix de techno pour votre API est Symfony ou Node JS.

Cette API comportera à minima les tables suivantes :

- Users
- Rôles
- Articles
- Commentaires

Votre API va être composée à minima de plusieurs routes:

- Une route qui permet à un utilisateur de s'enregistrer en complétant les champs prenom, nom, email et mot de passe.
- Une route qui permet à l'utilisateur de se connecter
- Une route qui retourne tous les articles
- Une route qui retourne tous les commentaires des articles
- Une route qui permet de poster un article
- Une route qui permet de commenter un article
- Une route qui permet de modifier ses informations (uniquement celles de l'utilisateur connecté)
- Les routes qui permettent de supprimer, modifier un user
- Une route qui permet de supprimer un message de n'importe qui

En vert : ce sont les routes ouvertes à tous, qui ne requièrent pas d'authentification

En orange : ce sont les routes ouvertes uniquement aux utilisateurs authentifiés, et à l'admin

En rouge : ce sont les routes ouvertes uniquement à l'admin

Chaque route doit avoir son jeu de test via Postman.

Il serait idéal que vos fonctionnalités de votre code soient aussi testées.

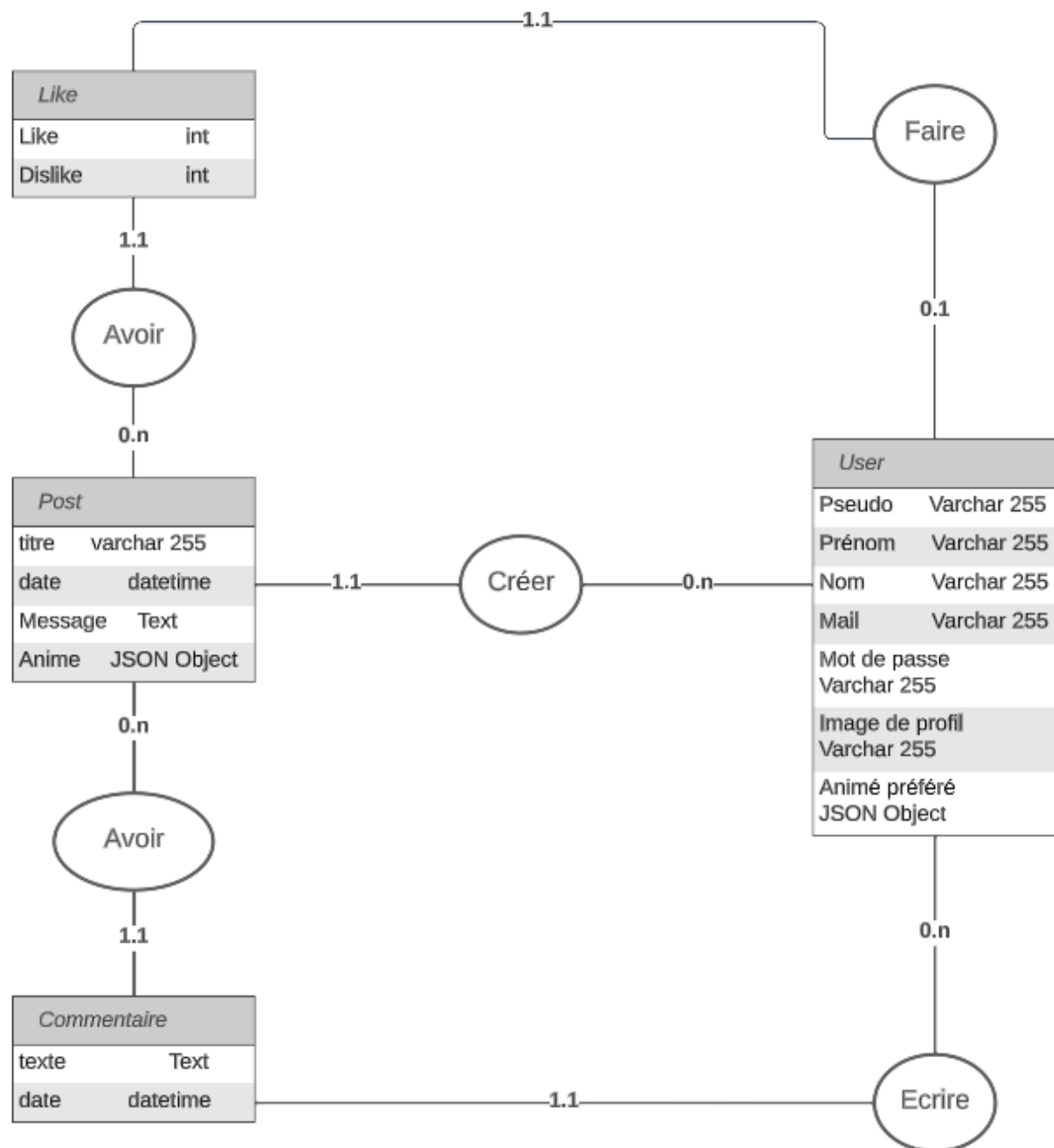
App Mobile

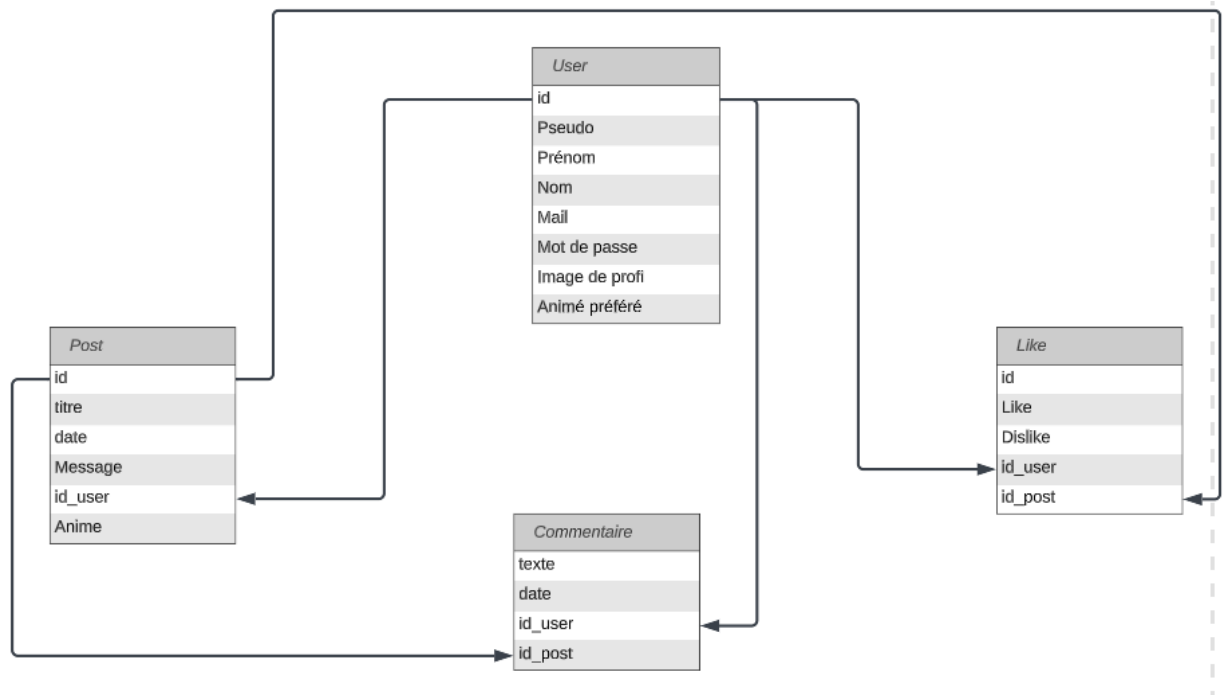
Maintenant que votre API est prête, vous allez à présent développer votre App Mobile en React Native. Votre app devra être composée à minima des différentes pages :

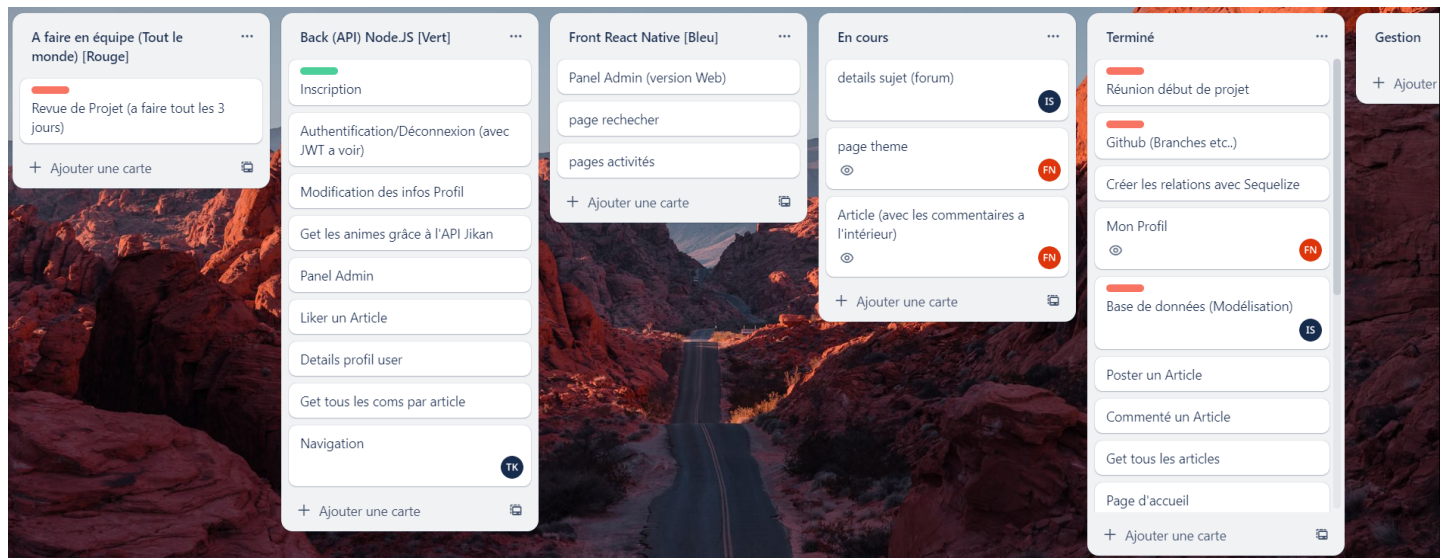
- Inscription
- Connexion
- Forum (liste des articles)
- Article (article avec commentaires associés)
- Profil (accès aux infos du user et possibilité de les modifier)

Rendu

Le projet est à rendre sur github <https://github.com/prenom-nom/app-mobile-forum>







DROITS

