PTS4

Rapport

Groupe 22

"Social Network"

Auteurs :
Morgan NEHDI
Antoine LANCELOT

Client :
IUT Laval Département MMI et TC

Adresse :
52 Rue des Docteurs Calmette et Guérin
53000 Laval

Enseignants à l'initiative du projet :
- Sebastien.George@univ-lemans.fr

IUT Laval
Le Mans Université
Département Informatique

SOMMAIRE

I. Mise en place du projet

1. Contexte et sujet

Dans le cadre de notre projet tutoré, nous avons fait le choix de créer une application type réseau social. En effet, un réseau social est composé d'une multitude de fonctionnalités différentes. On peut retrouver l'authentification, la publication, la gestion des fichiers, la gestion du temps réel, les services en tâche de fond...Ce type d'application nous permet d'étendre nos connaissances et d'approfondir l'apprentissage de nouveaux langages/conceptions.

2. Choix des technologies

Durant plusieurs années, nous effectuons des veilles technologiques quotidiennes. Cela nous permet de découvrir des nouvelles technologies, langages et conceptions.

Pour ce projet nous avons utilisé :

- Flutter
- Ferry
- Neo4j
- NodeJS
- Apollo Graphql
- Minio
- NGINX
- PM2
- Lottie

L'ensemble des ces technologies sera abordé par la suite.

3. Environnement de travail

Afin de garantir le bon déroulement du projet, nous avons utilisé 3 outils majeurs. On retrouve tout d'abord Discord : outil que l'on connaît depuis maintenant quelque temps et qui nous sert de moyen de communication lorsque nous travaillons sur le projet à distance. Le deuxième outil que nous avons utilisé est Github : il nous à permis d'échanger rapidement nos codes et à les fusionner pour travailler sur le projet de façon optimale et rapide. Afin de visualiser le rendu de notre application finale, nous avons utilisé Figma qui nous permet de concevoir des maquettes et de prototyper nos idées.

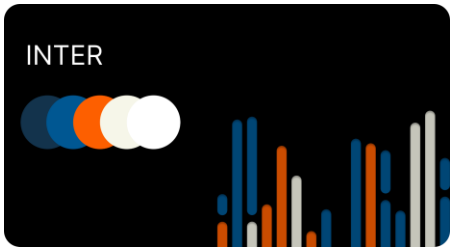
Nous avons également mis en place un serveur hébergé dans un data center (OnetSolutions situé à Bordeaux). Car suite à l'expérimentation de la solution d'hébergement proposée par l'IUT ne fut pas concluante (proxy bloquant).

II. Maquettes UI/UX

1. Charte graphique

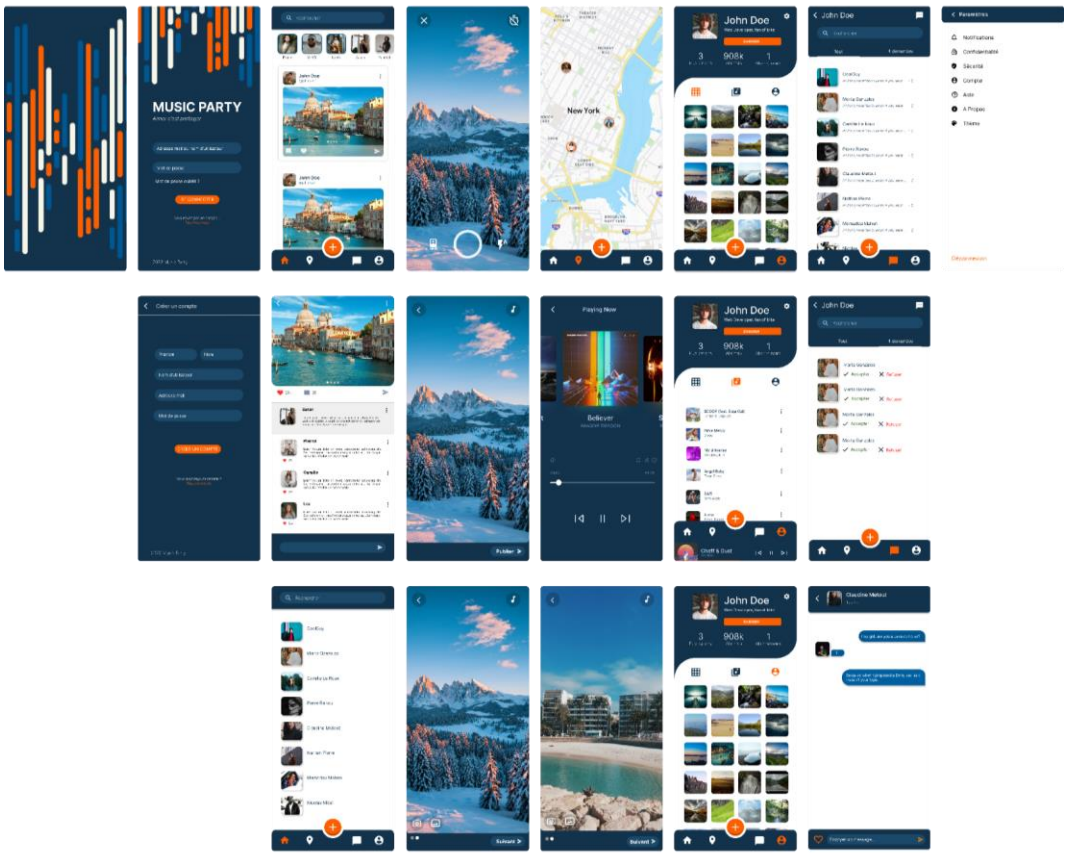
Nous avons commencé par créer une charte graphique, avec une police (Inter), une palette de couleur et un pattern qui peut être utilisé un peu partout pour donner une identité à l'application.

Nous utilisons les icons natifs de Flutter : Material Design icons créé par google.

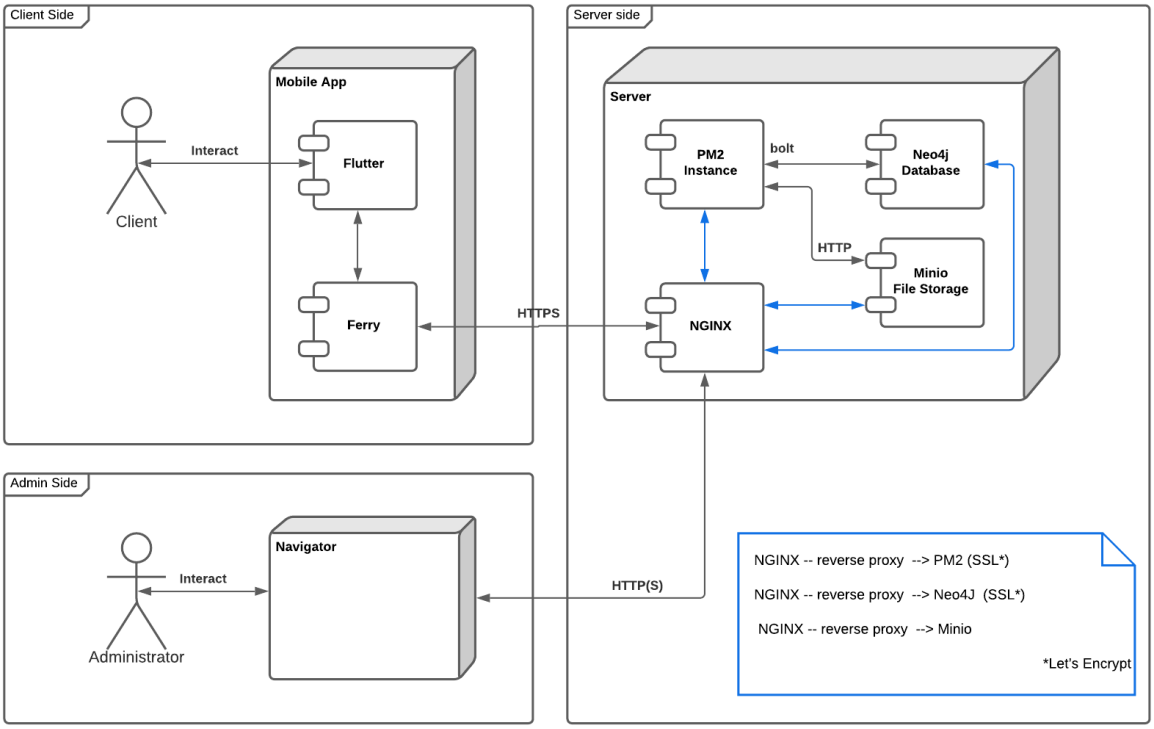


2. Figma

Afin de collaborer sur le design, nous avons utilisé l'application Figma. De plus, celle-ci permet de créer des composants réutilisables, des interactions et prototypes.



III. Architecture



3. Client

a. Dart

Dart est un langage de programmation interprété de haut niveau qui permet de créer des applications mobiles, serveur, de bureau et Web. Dart est assez facile à appréhender, moderne, fonctionnel, flexible et compétitif. L'écosystème est simple, comprendre les terminologies, les outils et les SDK appropriés pour le langage est facile. C'est un langage de programmation orientée-objet comme Swift ou Kotlin.

b. Flutter

Flutter est une technologie récente, développée par Google permettant de créer des applications multiplateformes pour Android, IOS ou encore Web. Il utilise le langage de programmation Dart.

Le premier avantage de Flutter est qu'il permet de réduire le temps de développement. Au lieu de créer deux développement natifs en parallèle, la technologie permet de développer 2 versions d'une application mobile (Android et iOS) en utilisant le même code source.

Un autre avantage est que la compilation est extrêmement rapide. En effet, l'application se recharge automatiquement quand le code est modifié de manière presque instantanée (hot reload).

c. Ferry

Ferry est un ensemble de librairies permettant d'utiliser des requêtes en graphql directement depuis flutter. Cette librairie permet de convertir automatiquement les fichiers requêtes de type .graphql en objets Dart.

De plus, la solution offre un widget dédié, le widget "Operation" permet d'encapsuler d'autres widgets afin d'exécuter une opération avant durant le rendu des widgets.

d. Lottie

Lottie est une solution open source développée par Airbnb, qui permet d'exporter des animations complexes After Effect et ainsi de pouvoir les utiliser nativement via une librairie Flutter.

En effet, aujourd'hui les animations représentent l'une des tendances cruciales de l'interface utilisateur d'une application moderne. Elles permettent de créer un véritable lien entre l'utilisateur et l'interface.

2. Serveur

a. NodeJS

Node.js est une plateforme logicielle libre en JavaScript, orientée vers les applications réseau évènementielles hautement concurrentes qui doivent pouvoir monter en charge. De plus, il est facile à apprendre, rapide, évolutif et possède une communauté active.

NodeJS possède un gestionnaire de paquet assez fourni, permettant d'importer des librairies facilement.

b. Apollo Graphql

GraphQL est un langage de requête et un environnement d'exécution côté serveur. Utilisé à la place de REST, GraphQL permet aux développeurs de créer des requêtes récupérant l'intégralité des données nécessaires à l'aide d'un seul appel à l'API. (retour en JSON)

Apollo Graphql est une librairie javascript, permettant d'exposer un "endpoint" unique, une interface de requêtage et de définir la structure de l'API.

Exemple de requête d'un utilisateur, de ses amis et de ses postes :

```
query fetchData {
  users {
    id, email, firstName, lastName, email, image,
    friends {
      id, firstName, lastName, image,
    },
    posts {
      id, description, createdAt
    }
  }
}
```

c. Neo4j

Neo4j est un système de base de données orientée graphe open source développé en Java.

Nous avons fait le choix d'utiliser ce type de base de données, car un réseau social peut être vu comme un graphe, avec des connexions entre les données.

L'avantage d'utiliser ce type de base de données est le fait qu'il est possible de parcourir le graph de manière très rapide. On peut donc proposer aux utilisateurs des recommandations de publication, utilisateurs et musiques en temps réel en quelques secondes en une seule requête.

Neo4j dispose de son propre langage de requête orienté graphe, il s'agit de Cypher. Le langage se veut simple et efficace dans la formulation des requêtes d'interrogation et de mise à jour de Neo4j.

Exemple de requête et de rendu :

```
$ MATCH (n) RETURN n;
```

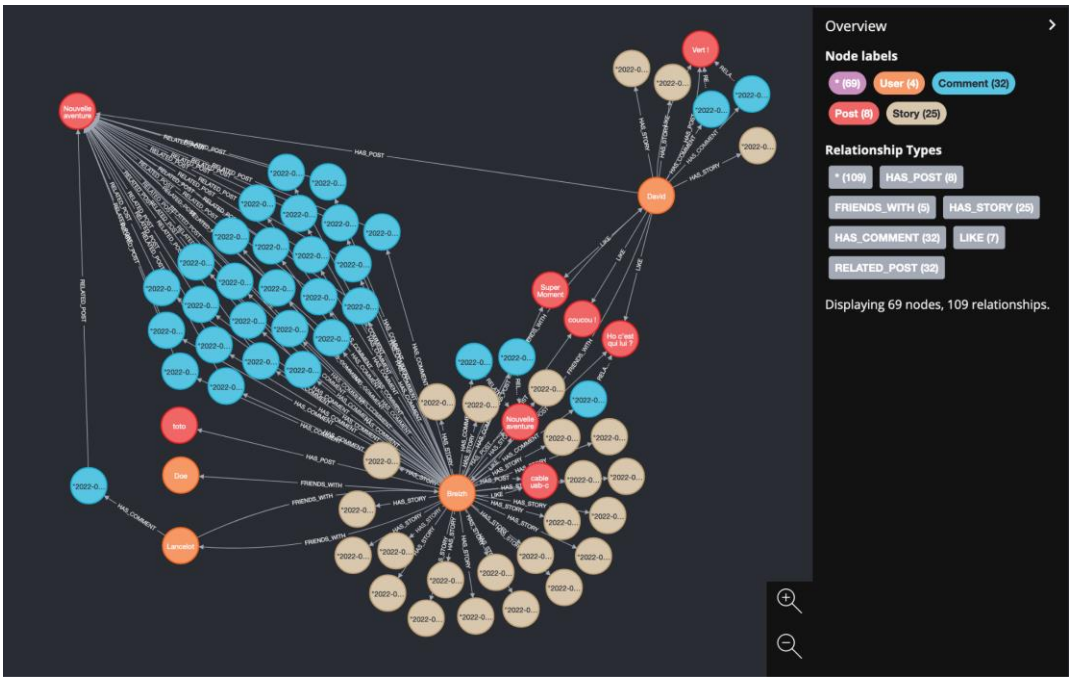
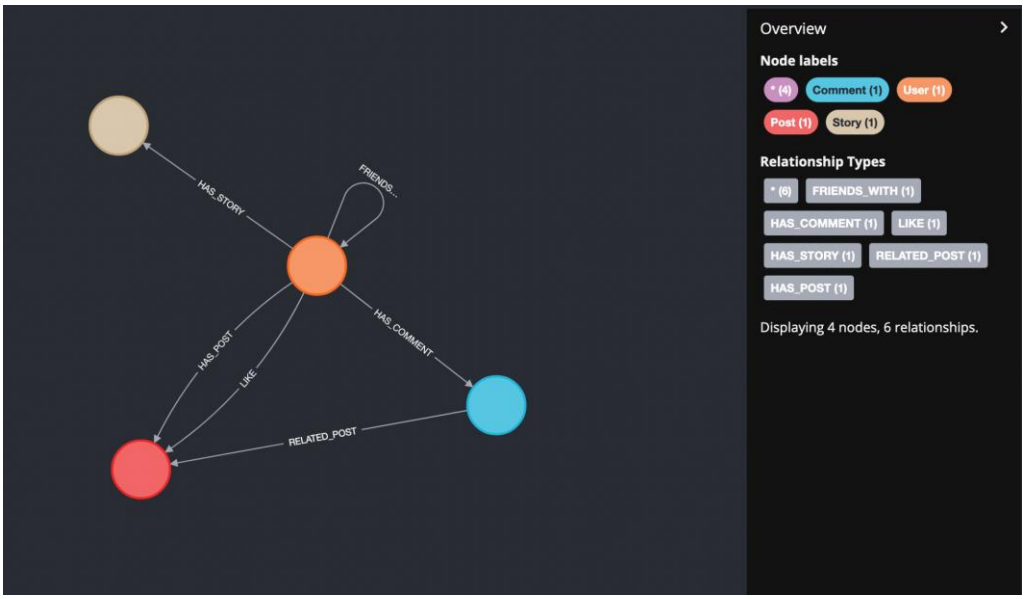


Schéma de la base de donnée :

```
$ CALL db.schema.visualization();
```



d. Minio

Minio est un logiciel de stockage d'objets hautes performances auto-hébergé compatible avec l'API AWS S3.

L'utilisation d'un stockage d'objets distribués permet de faire face à de grosses quantités de données.

Les atouts d'un tel mécanisme sont :

- Haute performance (lecture/écriture jusqu'à 170 GO / s)
- Évolutif (Utilisable en mode clustering)
- Utilisation de l'API AWS S3, pour une utilisation dans le cloud juste en changeant la configuration (url, public key, private key)
- Notification d'évènement (envoyer des données sur Kafka, MQTT, …)
- Cryptage multiple pris en charge, y compris AES-CBC, AES-256-GCM, ChaCha20
- Gestion d'identités
- Réplication des données et reprise après sinistre

e. PM2

PM2 est un logiciel open source, permettant de lancer des script NodeJS en production sous forme de service/micro-service. Il permet donc de lancer, démarrer (à chaud sans coupure) et arrêter des instances. Mais il comprend une gestion des logs, le monitorage en temps réel des ressources, la réplication d'application et la répartition de charges (load balancer).

f. NGINX

Dans la mise en place de notre infrastructure, nous avons mis en place la solution NGINX. Il s'agit d'un logiciel libre de serveur Web ainsi qu'un proxy inversé (reverse proxy) pour les sites ayant besoin d'un fort trafic.

Configuration: **(possibilité de demander des accès aux dashboard)**

- PM2 en proxy inversé (support SSL Let's Encrypt)
  - https://cloud-549579146.onetsolutions.network/
- Neo4J en proxy inversé sur le port (SSL)
  - https://cloud-549579146.onetsolutions.network/neo4j
- Minio en proxy inversé
  - http://cloud-549579146.onetsolutions.network/minio

L’utilisation d’un proxy favorise la sécurisation de l’infrastructure.

IV. Les aspects Gestion/Organisation

Afin de faciliter le développement de l'application, nous avons tout d’abord réalisé les interfaces de l’application à l’aide de Figma. Figma est un éditeur de graphiques vectoriels et un outil de prototypage. Une fois, les interfaces réalisées, il fallait désormais développer les fonctionnalités en elle-même.

Ensuite, nous avons réparti les tâches, Antoine a commencé à intégrer les maquettes (côté client) pendant que Morgan mettait en place la machine et l’API (côté serveur).

Puis nous avons commencé à intégrer petit à petit la logique à notre application, en commençant par la connexion et l’inscription, puis le menu, les vues principales et annexes.

Durant ce projet, nous étions que deux, et les centres d'intérêt et compétences de chacun ont déterminé notre organisation.

V. Application

Durant ce projet tutoré, nous avons avancé sur les différents aspects constituant l’application finale.

Nous avons travaillé sur une multitude d’aspect comme :

- La mise en place un serveur avec différentes application
- Les interface graphiques
- L’authentification
- L’utilisation de l’API Spotify
- Mise en place de requête de Graphql
- Les animations
- L'expérience utilisateur
- Les services en arrière plan
- La gestion des fichiers
- L’utilisation de l’API MapBox (Carte interactive)
- L’utilisation des appareil photo / stockage en cache
- …

Le projet projet de ce semestre étant très court, il y a pleins de fonctionnalités qui n'ont pas été faites, c’est pourquoi il y a plusieurs aspects qui peuvent être améliorés ou ajoutés :

- Meilleur gestion du cache
- Refactor du code
- Mise en place de testes
- Utilisation d’une autre base de données pour les stories/messages…
- Mise en place de filtres photos

- Optimisation des médias envoyé sur le serveur pour réduire le poid des images (et du chargement)
- Mise en place de la messagerie
- Mise en place de notifications
- Personnalisation des paramètres de comptes
- …

VI. Bilan et Conclusion

Globalement, nous avons apprécié ce projet. Réaliser une application de type réseau social en si peu de temps est un défi ambitieux. A ce jour, l'application peut donc encore évoluer et de nouvelles fonctionnalités peuvent être ajoutées. Nous sommes tout de même fiers du résultat malgré la frustration engendrée face à l'émergence d'idées que nous n'avons pas pu concrétiser. Durant ce projet, il n'y a pas eu de difficultés particulières, les différentes phases du projets furent fluides et notre duo efficace.

Durant ce projet, nous avons mis l'accent sur l'expérience utilisateur (UX) et l'interface utilisateur (UI). Nous nous sommes efforcés de suivre le design d'origine, c'est pourquoi notre l'application est le reflet exact de nos interfaces.

Pour terminer, ce projet nous à permis d'apprendre de multitudes de technologies, méconnues au début de ce semestre. Avant de commencer ce projet, nous avions comme optique d'utiliser uniquement des technologies que nous ne maîtrisons pas. Plongés dans la peau d'autodidactes, nous nous sommes documentés et avons concrétisé ce projet en totale autonomie.