HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

TWEB - MAC
Paul Ntawuruhunga & Nastaran Fatemi
Miguel Santamaria & Gary Marigliano

Projet - Réseau Social

Objectifs

Le but de ce projet est de mettre en pratique la matière étudiée durant les cours TWEB et MAC afin de déployer une application web complète. L'objectif est de créer une application sociale permettant l'échange de ressources (photos, vidéos, musiques, ou autre) entre différents acteurs (qui seront en principe des utilisateurs) reliés grâce au principe d'abonnements/followers (voire des amis ou autre, selon le projet). Les ressources seront tout d'abord postées par un utilisateur via un formulaire, puis seront affichées dans la rubrique "Mon Mur" des utilisateurs qui le suivent. Par exemple, un utilisateur connecté verra apparaître dans son mur les recettes de cuisines postées par son ami Michel auquel il est abonné.

Les étudiants qui suivent le cours de **MAC** cette année développeront aussi des fonctionnalités liées à de l'analytique, à des recherches avancées, à des annotations (par exemple des hashtags), et à des notifications (liées à des réactions et à des actions utilisateurs).

Tout d'abord, pas de panique ! Le but n'est pas de développer un produit complet tel que Facebook ou Instagram : vous imaginerez d'abord un concept, puis développerez un MVP (minimum viable product) en vous focalisant sur les fonctionnalités essentielles. Attention cependant au fait que ce projet n'est pas non plus à prendre à la légère, car il reste bien fourni. Un conseil : visez petit, mais fonctionnel.

Au niveau de la technique, l'application sera composée de trois tiers : un back-end, un front-end, et une API pour les relier :

1. Le **back-end** s'occupera de stocker, de gérer (**create/read** et update/delete facultatifs) et d'analyser (data analytics, en MAC) les ressources partagées par les utilisateurs. Pour ce faire, vous utiliserez une base de données (dont les exigences dépendront de vos cours respectifs) adaptée au type d'information à partager.
2. Une **API** (REST ou équivalent) permet d'exploiter les fonctionnalités offertes par le back-end.
3. Le **front-end** consomme l'API et réagit aux évènements des utilisateurs (par exemple s'abonner à un utilisateur, rechercher une information, publier une ressource).

Au terme de ce projet en **TWEB**, vous aurez aussi l'occasion de faire face aux réalités du développement d'un produit en ligne. Nous utiliserons une méthode appelée "User Testing" qui a pour but d'évaluer l'ergonomie d'une interface en observant la manière dont des utilisateurs réels s'en servent. C'est la raison pour laquelle, l'expérience utilisateur, la qualité du code et les meilleures pratiques concernant l'utilisation des technologies et framework modernes seront les sujets principaux tout au long de cette deuxième partie du semestre.

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

TWEB - MAC
Paul Ntawuruhunga & Nastaran Fatemi
Miguel Santamaria & Gary Marigliano

Règles

- Le projet sera réalisé par groupes de 3. Prenez en considération le fait que certains étudiants suivent un cours, mais pas l'autre ; dès lors, il est possible d'avoir un groupe de 4, avec deux personnes qui travaillent à 50% (l'une dans TWEB, l'autre dans MAC). Attention cependant à l'organisation que cela engendre !
- L'application contiendra un front-end, un back-end, et une base de données (pour tout le monde), ainsi que des fonctionnalités avancées liées aux données pour les IL (en MAC).
- Dans le cas où vous n'auriez que MAC, deux possibilités s'offrent à vous : soit vous décidez de vous greffer dans des groupes qui suivent aussi TWEB, soit vous décidez de faire un projet à part entière. Dans ce dernier cas, nous vous demandons de rapidement venir nous voir avec une idée de projet idéalement pas trop complexe, et nous vous fournirons un front-end et une architecture de projet minimaux.
- Une documentation complète de l'application est demandée (choix des technologies, API, requêtes et données, déploiement, utilisation, auto-évaluation, tests, etc.). Le choix de la technologie (MarkDown, Office, LaTeX) est libre.
- Votre code sera hébergé sur un dépôt git.
- Le projet est séparé en 4 phases qui sont décrites plus bas.
- Deadline finale pour le rendu : **20.01.2019**. Des présentations auront lieu la dernière semaine.
- La note du projet a une pondération de 70% sur la note finale des laboratoires.
- Le projet est d'une grande ampleur, et il sera impossible de le développer totalement dans les dernières semaines, organisez-vous donc pour travailler à intervalles régulières.
- Comme toujours, la créativité et la qualité de l'application sont récompensées. Sentez-vous libres d'explorer des horizons diversifiés !

Fonctionnalités TWEB

Vous trouverez dans ce chapitre les fonctionnalités minimales requises pour l'évaluation de TWEB.

Authentification et profil utilisateur

Votre application gèrera l'inscription et la connexion d'un utilisateur, ainsi que l'accès à sa page de profil.

Partage de ressources (rubrique "Mon Mur")

Votre application permet de partager **au minimum un type de donnée**. C'est à vous de définir le type de donnée selon le concept de votre l'application : photos, vidéos, messages, articles, positions géographique, ou autre. Fondamentalement, votre mur s'occupera de récupérer les dernières ressources publiées par les abonnements de l'utilisateur à intervalles régulières, et les affichera.

API

Votre API doit être développée à l'aide de REST ou de GraphQL et doit fournir au minimum des endpoints permettant de gérer les fonctionnalités décrites plus haut :

- au login et à l'authentification
- à l'accès et à la modification de comptes utilisateurs / des acteurs
- à la création et à l'accès aux ressources ; attention à bien réfléchir aux différents cas possibles selon votre projet ("je veux accéder à toutes les publications de mes abonnements par ordre chronologique" ; "je veux accéder aux publications qui ont été postées par tel ou tel acteur" ; etc.)

Critères d'expérience utilisateur

L'application devra être conforme à un ensemble de bonnes pratiques en terme d'expérience utilisateur dans le but d'être ergonomique et accessible à tous. Une bonne organisation de l'interface ainsi que l'implémentation des concepts ci-dessous aidera les utilisateurs qui testeront votre projet à s'y retrouver.

- Onboarding: dans le but de former l'utilisateur et de lui souhaiter la bienvenue, l'application donne un aperçu des fonctionnalités principales et présente les menus de bases.
- UI States: l'application indique clairement l'état des différentes zones de l'interface (lorsqu'un chargement est en cours, lorsqu'une erreur est survenue ou lorsqu'une liste est tout simplement vide, par exemple).
- Feedback: l'application réagit immédiatement aux actions de l'utilisateur en effectuant des changements visuels sur l'interface. Ces changements se déclinent sous différentes formes en fonction de la situation (transitions, curseur de la souris, bulle de notification, etc.).

Évaluations TWEB

Les étudiants seront notés sur la base des critères suivants :

- Les fonctionnalités demandées ont été implémentées.
- L'application est accessible publiquement (sans crash).
- L'application est suffisamment ergonomique pour être utilisée par un utilisateur lambda.
- Il est possible de tester l'application localement.
- Qualité et robustesse du code

Bonus

- Bonus UX: les feedbacks sur l'expérience des utilisateurs ayant testé votre application web vous permettront d'obtenir un bonus.
- Bonus effort: tout effort fourni et documenté sur un aspect de l'application est récompensé d'un bonus, en particulier ce qui concerne l'UX.

Fonctionnalités MAC

Vous trouverez dans ce chapitre les fonctionnalités minimales requises pour l'évaluation de MAC.

Choix de technologies NoSQL

L'application doit utiliser une des technologies suivantes: mongoDB, CouchDB, Dynamo, Arango, Neo4J, Orient, ou Elastic Search.
Une même technologie peut être choisie par deux groupes au maximum.

Gestion d'annotations ou Notations (Rating, like/dislike)

Ici, nous vous demandons de faire le choix entre **au moins l'une de ces deux fonctionnalités** :

- **Gestion d'annotations** : lors du partage de ressources (e.g. un post), une prise en charge d'annotations est demandée. Un exemple d'annotations est l'utilisation de hashtags (e.g. "Je me réjouis de ce test de maths #ironie") ou encore l'utilisation de mentions (e.g. "Et toi @alice, tu pars aussi en vacances ?"). Ces annotations vont vous permettre d'effectuer des requêtes avancées expliquées dans les points suivants.
- **Notations** : l'application intègre un système de notation qu'il vous est libre de choisir. Par exemple, l'utilisation de like/dislike ou de rating (e.g. 5 étoiles). On gardera en tête que l'on pourra effectuer des recherches ou d'autres fonctionnalités analytiques sur ces notations.

Recherche avancée

Une série de fonctionnalités de recherche adaptées à votre application, comme par exemple :

- Rechercher les gens avec des intérêts/profils similaires.
- Rechercher les médias selon différents critères (e.g. titre, description).
- Recherche de publications portant une annotation (e.g. lister les posts avec le hashtag #ironie)
- Etc.

Proposez également à vos utilisateurs des filtres et autres tris vis-à-vis de ces recherches avancées.

Notifications (Flux d'activités)

En complément du "mur" qui contient les dernières publications/derniers partages auxquels on est abonné, un système de flux d'activité doit être implémenté. Le flux d'actualité permet aux utilisateurs de rester informés des sujets qui les intéressent. Concrètement, un utilisateur verra afficher des notifications du genre "Alice est désormais abonnée à vos publications" ou "Bob a mis à jour sa photo de profil", "Michel a aimé votre publication". Votre application intègre cette fonctionnalité dans la manière qui convient le mieux selon le concept choisi.

API

En plus des fonctionnalités de l'API listées dans la partie TWEB, votre API doit fournir au minimum des endpoints permettant de gérer les fonctionnalités décrites ci-dessus :

- Récupérer les notifications d'un utilisateur donné
- Actions sur les annotations / notations
- Effectuer des recherches et récupérer les résultats de ces dernières
- Récupérer les informations contenues dans les fonctionnalités analytiques

Évaluations MAC

- Modélisation des données
  - Bons choix de structures de données lors de la modélisation (structure de documents, graphes, etc)
  - Bonne utilisation des mécanismes spécifiques offerts par la base NoSQL de votre choix
  - Le modèle de données doit tenir compte des besoins de requêtes et doit favoriser la réalisation efficace des requêtes les plus courantes.
- Qualité des requêtes
  - Utilisation des opérateurs appropriés selon la base NoSQL choisie
  - Performance des requêtes
  - Indexation: bons choix des structures d'indexation selon les besoins

Pour chaque partie pensez à documenter et à justifier vos choix.

Bonus

- L'analyse de la montée en charge, évaluer les goulets d'étranglement et proposer des mitigations.
- Mise en place de sharding/réplication
- Bonus effort : tout effort fourni et documenté sur un aspect de l'application est récompensé d'un bonus.

Phases

Le développement de cette application est divisé en plusieurs phases

- **Phase 1** (1 semaine - pour le 28-29 novembre)
  Imaginer le concept de l'application sociale et définir la fonctionnalité la plus importante à implémenter.
  Choix des technologies :
  - Front-end (React, Vue.js, Angular, ou autre)
  - Back-end (Express ou autre)
  - Définir la méthode de communication entre l'application front-end et le back-end (REST API, GraphQL, ou autre)
  - Choix des technologies utilisées pour la persistance
  - Choix des services d'hébergement de l'application
- **Phase 2** (2 semaines - pour le 12-13 décembre)
  Développement des pages de login et de bienvenue, initialisation de la base de données, et premier déploiement du tout.
- **Phase 3** (pour le 20 janvier)
  Implémentation
- **Phase 4** (durant la semaine du 14 janvier)
  User testing