



Challenge S2 - 5IWJ - 2024

Coachs

Matières	Coachs
Golang	Matteo DELANDHUY
Flutter	Raphaël KIFFER

Présentation du projet

Le but de ce projet est de réaliser des applications **Flutter** mobile et web s'interfaçant avec une API **Golang** pour **gérer des groupes de personnes autour d'une thématique commune (école, concert, voyage, sport, événement...)**. Il sera nécessaire de trouver un sujet correspondant aux fonctionnalités principales décrites ci-après et de le faire valider par les coachs lors de la première journée.

Le projet est à réaliser en groupe de 3 ou 4 étudiants.

C'est à vous d'établir la thématique et les fonctionnalités présentes dans le projet sous forme de mini-spécifications.

Fonctionnalités principales (15 pts)

- **Golang**
 - API REST
 - Système d'authentification (au minimum via login / mot de passe), lors de l'inscription envoyer un email de confirmation avec un code valable 15min qui sera stocké dans le Redis
 - Gestion multi-rôles (utilisateur simple, administrateur...), vérification des droits et filtrage des résultats lors des requêtes en fonction du rôle de l'utilisateur connecté
 - Interface Swagger UI pour interagir avec l'API
 - API CRUD sur les éléments fonctionnels de l'application
 - Pagination des requêtes *GET all* par rapport aux paramètres envoyés par le frontend
 - API Web Socket fournissant au moins une fonctionnalité temps-réel
 - Configuration modifiable (via fichier de configuration, variables d'environnement, base de données...)
 - Logs formatés consultables

- Base de données relationnelle avec jeu de données conséquent représentatif des cas d'usages de l'application
- Au moins une route API doit utiliser un service tiers qui sera consommé via une requête
- Gérer le *batch request* (plusieurs requêtes en une)
- **Flutter**
 - Application mobile (iOS & Android)
 - Écrans différents en fonction des rôles de l'utilisateur
 - Envoi de notifications push (Android uniquement)
 - Utilisation d'une technologie pseudo temps réel
 - Intégration du multilingue
 - Gestion *offline* d'une fonctionnalité de l'application
 - Token d'authentification unique de l'app pour faire des requêtes à l'API
 - Application web
 - Panel d'administration cohérent pour contrôler les différentes entités du projet
 - Affichage des événements survenant au sein de l'application pour consultation

Fonctionnalités bonus (5 pts)

- Système d'authentification OAuth
- Interface de consultation des logs (dans l'application ou via outil tierce KIBANA, Grafana...) avec graphiques et stats pertinentes (fréquentations, erreurs...)
- Intégration de fonctionnalités tierces comme un carte Google Maps avec de la géolocalisation
- Scalabilité horizontale
 - possibilité de déployer plusieurs instances de l'API derrière un load-balancer afin de supporter une fréquentation forte
 - tests de montée en charge
- ...

Contraintes

- **Golang**
 - Utilisation de Go 1.22.1
 - Dockerisation du backend
 - Base de données PostgreSQL avec utilisation de l'ORM Golang
 - Redis pour gestion des données éphémères avec TTL
 - API stateless, avec pagination des résultats

- Configuration modifiable (fichier de configuration, variables d'environnement, base de données...)
- Tests
 - tests unitaires sur les règles de gestion de l'application
 - tests d'intégration sur les différentes routes de l'API
 - tests des cas d'erreur
- Erreurs
 - pas de "panic" dans le code, les erreurs doivent être correctement gérées
 - renvoi des erreurs fonctionnelles par l'API et affichage du détail des erreurs techniques dans les logs
- Sécurité
 - API hébergée en HTTPS avec certificat SSL
 - Respect des normes de développement standard (OWASP...)
- Clean code
 - Le code doit être découpé en fichiers et dossiers cohérents et respecter les règles du clean code et SOLID (découpage, noms des variables et fonctions...)
 - Code formaté et indenté avec Gofmt
 - Fonctions et contenu de fonctions commentés si besoin
- **Flutter**
 - Utilisation de plusieurs émulateurs ou terminaux pendant la soutenance pour faire les démonstrations nécessaires en fonction des différents profils
 - L'application devra être visuellement responsive en fonction de la taille du téléphone / navigateur
 - Le code rendu doit être propre et indenté correctement via les outils de base mis à disposition

Validation du sujet

Le présent cahier des charges sert de base à la réflexion de mini spécifications concernant le projet que vous allez réaliser. Ces mini spécifications seront à valider lors de la première journée de coaching pour établir un sujet solide et une roadmap pour le reste des séances.

Rendu

- **Dépôt(s) GitHub**
 - **▲ Tous les membres du projet doivent commit sur le dépôt** (commits signés via une clef GPG)
 - `README.md`

- NOM Prénom de chaque étudiant avec compte GitHub associé
 - Fonctionnalités du projet et répartition du travail
 - Toute information semblant nécessaire...
 - **Mise en production**
 - Pour valider, votre projet doit être impérativement mis en production pour le jour de la soutenance et ce durant au moins 2 semaines. Vous pouvez utiliser un VPS (Hetzner, DigitalOcean) ou par exemple des services comme Fly.io
 - **Golang**
 - Dépôt Git contenant le code du backend
 - Fichier `README.md` détaillant
 - La listes des binaires et les fonctionnalités fournies
 - Les procédure de compilation et d'exécution du ou des binaires
 - Les procédures de lancement des tests unitaires
 - Un exemple de configuration pour lancement sur poste de travail local
 - Archive au format **ZIP sur MyGES**
 - Éléments tiers nécessaires (configuration, certificats, variables d'environnement...)
 - **Flutter**
 - Dépôt Git contenant le code de l'application Flutter
 - Archive au format **ZIP sur MyGES**
 - Éléments tiers nécessaires (certificats, variables d'environnement...)
 - Application Android de production sous forme d'APK signé et installable
-

Notation matière & Challenge

Matières

- **Contrôle continu** : réalisé pendant le cours
- **Examen** : analyse technique sur le rendu du challenge

Challenge

- **Contrôle continu** : suivi de projet sur les différentes dates de coaching
- **Examen** : soutenance du challenge avec rendu global

Barème sur la note de suivi de projet du challenge

Pour chaque session de coaching, le barème ci-dessous est à remplir pour chaque équipe par les coachs. La note de suivi est une note de groupe.

Compétence à évaluer	Échec (0 pt)	Médiocre (1 pt)	Passable (2 pts)	Bien (3 pts)	Excellent (4 pts)
Présentation du projet et de son avancement					
Qualité du travail produit					
Respect du niveau d'avancement prévu					
Organisation du groupe					
Cohésion du groupe					

Organisation des soutenances

Un ordre de passage sera défini pour soutenir votre projet. Vous aurez 30 minutes par groupe, à répartir selon le schéma suivant :

- 20 minutes de présentation / démonstration
- 10 minutes de questions / réponses

Pour les 20 premières minutes, vous pouvez suivre la ligne directrice suivante :

- Présentation rapide du concept et de l'équipe (à l'oral, 30 secondes)
- Présentation des grandes fonctionnalités du projet (2 minutes max.)
- Démonstration de votre projet à travers les différents parcours utilisateurs