

Compte Rendu du Projet AREA

Première Défense

Cette première défense avait pour objectif d'évaluer la qualité de notre phase de recherche, notre capacité à comparer différentes technologies via des POC, ainsi que la cohérence globale de nos choix pour construire l'architecture du projet AREA.

Nous avons également été évalués sur notre organisation, notre méthodologie de travail et notre capacité à planifier correctement le développement à venir.

Récapitulatif de l'Évaluation

Nous avons présenté deux comparatifs distincts pour le choix des technologies frontend. Pour le Web, nous avons évalué React, Next.js, Blazor WebAssembly et Flutter Web selon des critères tels que le poids, la performance, la documentation, le déploiement et la cohérence de l'interface utilisateur. Notre choix s'est porté sur Next.js, en raison de sa cohérence entre les versions mobile et web, de son hot reload, de son excellente documentation et de la simplicité de son déploiement. Pour le Mobile, nous avons comparé React Native, Flutter et Kotlin en fonction de la portabilité, de l'écosystème, de la communauté et de la rapidité de développement. Flutter a été retenu pour ses performances cross-platform, son hot reload et sa communauté active.

Un comparatif a été réalisé entre Go, Python et NestJS en fonction de la performance, de la consommation des ressources, de l'écosystème et de la facilité de déploiement. Le choix final s'est orienté vers Go, pour sa légèreté, sa rapidité et son efficacité. Une remarque subsiste : il aurait été préférable de comparer uniquement des langages ou des frameworks entre eux, afin d'éviter les biais méthodologiques.

Aucun point n'a été abordé concernant la sécurité, ce qui constitue un manquement important. Ce sujet devra être traité lors de la prochaine défense, en incluant notamment l'authentification, la gestion des droits, le stockage sécurisé des données et la sécurité des API.

Un comparatif a été mené entre PostgreSQL, MongoDB et Redis, en tenant compte de la latence, du support JSON et de la cohérence avec les besoins du projet. Le choix s'est porté sur PostgreSQL comme solution principale, complétée par Redis pour des cas d'usage spécifiques.

L'équipe a adopté une méthodologie Agile avec des sprints de 7 jours, ce qui a permis une organisation claire et efficace.

L'outil Linear a été validé pour la gestion des tâches. Cependant, deux problèmes ont été identifiés : les tickets ne sont pas suffisamment détaillés, et il manque encore des issues nécessaires au MVP. De plus, la visualisation temporelle est insuffisante pour anticiper correctement les deadlines.

Voies d'Amélioration

Certaines issues essentielles pour la prochaine défense manquaient. Il est nécessaire de lister toutes les tâches à réaliser pour les 3 prochaines semaines, en s'assurant qu'aucune étape ne soit oublié

Ensuite il nous faut rendre nos descriptions d'issues plus précises. Les descriptions actuelles sont trop vagues. Pour chaque issue, il faut :

- Décrire clairement ce qui doit être fait et comment le réaliser.
- Ajouter des supports visuels (maquettes Figma, diagrammes Mermaid, schémas d'API, etc.).
- Définir des critères d'acceptation pour valider la tâche.

Ensuite il nous faut intégré un planning réaliste avec :

- Des deadlines claires pour chaque tâche.
- Une timeline ou roadmap pour visualiser l'avancement.
- Une estimation du temps nécessaire pour chaque issue.

Un point a été perdu car ce sujet n'a pas été abordé. Pour la prochaine défense, il faudra inclure :

- La gestion des tokens et l'OAuth2.
- La sécurisation du backend (protection des API, gestion des droits).
- Le stockage sécurisé des données utilisateurs.

Conclusion

Suite à cette première défense, notre priorité est de renforcer notre organisation pour préparer efficacement la prochaine étape.

Nos objectifs d'ici 3 semaines sont :

- Créer et compléter toutes les issues nécessaires, avec descriptions détaillées et supports visuels.
- Établir une roadmap claire, incluant un découpage temporel réaliste.
- Intégrer la dimension sécurité dans notre architecture.
- Commencer la réalisation du MVP, avec au moins une AREA fonctionnelle pour la prochaine défense.

Ce retour nous donne une direction claire et nous permet de nous organiser de manière plus rigoureuse pour la suite du projet.