POST-MORTEM AREA

BENISTANT David | DENNI Robin | FRANK Oscar | LIEBENGUTH Alessio | SCHMELTZ Théotime

1. **Mise en contexte**

Le projet AREA, intégré au cursus de troisième année du Programme Grande École à EPITECH, est notre deuxième projet technique de cette année, mais surtout notre premier projet web et mobile avec de telles attentes.

Ce projet vise à développer une application web, une application mobile, et une API. Cette application permettera aux utilisateurs de configurer des actions – réactions appelées AREA via une interface utilisateur graphique sur les services que nous avons décidé d’implémenter dans notre projet (Discord, Trello, Github, Asana, Google et Jira).

Le concept AREA repose sur l'automatisation de tâches inter-applications. Par exemple, une AREA pourrait se configurer de la manière suivante :

À la création d'une carte dans Trello, un message privé est envoyé sur Discord à l'utilisateur concerné, et simultanément, une carte est générée dans un tableau Jira.

L'objectif est d’effectuer ces opérations de manière fluide et automatisée, optimisant ainsi la productivité personnelle et professionnelle de l'utilisateur.

Par ailleurs, sachant que nous sommes 5 étudiants dans ce groupe, notre solution finale doit contenir au moins : 6 services et 15 actions/réactions.

1. **Choix des technologies utilisées**

Dans le cadre du développement du projet AREA, nous avons mis en place un benchmark sur les technologies potentielles que nous pourrions utiliser afin de mener à bien ce projet. Cette stratégie avait pour but de déterminer les langages les plus adaptés pour chaque composant de notre projet.

Concernant l'application web, nous avons mis en compétition React.js, Angular, Vue.js et SvelteKit. Les critères que nous jugions importants étaient ; la compétence collective du groupe dans chaque technologie, la performance de celle-ci, l'ampleur de la communauté d'utilisateurs (notamment sur des plateformes telles que Stack Overflow), la courbe d'apprentissage et l’espace occupé par les fichiers compilés. Le calcul des scores via un tableau Excel a révélé React.js comme le meilleur choix, en se distinguant par notre maîtrise collective, sa performance qui n’est pas abusive face à ses concurrents et sa vaste communauté.

Pour le développement de l'application mobile, SwiftUI, Kotlin, React Native et Flutter ont été comparés. Nous avons pris en compte la maîtrise de l'équipe, les performances de la technologie, la compatibilité multiplateforme, la taille de la communauté et la facilité d'apprentissage. React Native est ressorti comme le choix optimal, en raison de notre familiarité avec la technologie et de ses capacités multiplateformes.

Pour l'API nous avons réalisé une analyse des technologies suivantes : ExpressJs, ASP.NET Core, Python Flask et Phoenix (Elixir). Les critères de sélection se sont focalisés sur la maîtrise par l'équipe, les performances, l'aisance d'utilisation, la présence de la communauté et l'adéquation de la technologie avec la taille prévue du projet. ExpressJs s'est démarqué comme étant le plus conforme à nos exigences, offrant simplicité d'utilisation et adéquation avec les petits projets/micro services.

Enfin, pour la base de données, notre choix nous avons évalué MySQL, PostgreSQL, MongoDB, DynamoDB et Redis. Les critères déterminants étaient la maîtrise de l'équipe, les performances, la scalabilité, l'approche technologique (SQL ou NoSQL) et la communauté. DynamoDB s'est avéré être le candidat idéal, de nombreuses connaissances au sein de l'équipe, technologie NoSQL et intégration simplifiée avec AWS.

Le choix des technologies représente finalement une partie non négligeable pour la réalisation d’un tel projet, mais nous avons réussi à traverser cette étape à l’aide des connaissances de chaque membre du groupe mais aussi grâce à la volonté de chacun. Nous n’avons pas rencontré de difficulté au cours de cette étape, mais nous avons dû faire de nombreuses recherches afin de se fixer sur des statistiques réelles et non pas approximatives.

1. **Répartition des tâches**

Afin de ne décourager personne, nous avons laissé le choix à tout le monde de sa mission principale, laissant à chaque individu la liberté de choisir les responsabilités qui lui convenaient le mieux. Cette stratégie s'est avérée fructueuse, aboutissant naturellement à une distribution équitable des rôles.

L'équipe de développement web est composée d'Oscar FRANK et Alessio LIEBENGUTH. Le développement mobile est confié à Robin DENNI et Théotime SCHMELTZ. Puis pour finir, David BENISTANT prend en charge l'API et la base de données.

Cette étape qui n’est pas des moindres car c’est au cours de celle-ci que chaque membre du groupe est assigné à sa tâche et se doit de la faire au mieux possible afin de ne pas rendre les autres membres mécontents. De plus, nous sommes très satisfaits d’avoir passé cette étape rapidement, et ainsi, ne pas perdre de temps pour que tout le monde soit satisfait et content.

1. **Choix des services**

Comme évoqué précédemment, l'objectif de ce projet est de synchroniser différentes applications à l’aide d’actions et de réactions. Pour sélectionner les applications à intégrer, nous avons mis en place une réunion pour échanger sur les différentes propositions. Il était impératif que chaque membre du groupe contribue en intégrant au moins un service. Ce fonctionnement nous a permis de choisir individuellement les services à intégrer, contribuant ainsi à la qualité de notre solution finale.

Le choix des services s’est fait naturellement, nous avions tous une idée de service à intégrer dans notre solution, et nous pouvons aussi ajouter le fait que ces services ont une utilité à être liés donc une satisfaction supplémentaire est visible au sein du groupe.

1. **Maquettage – UI/UX**

Afin de respecter les normes de design et d’accessibilité, nous avons pris la décision de créer des maquettes web et mobile afin de s’assurer de conserver un design cohérent entre les deux plateformes, mais aussi en respectant les règles d’accessibilité comme la taille des boutons, l’utilisation (sans souris pour le web) ou encore certains contrastes de couleur à respecter.

La réalisation de maquette permet aussi aux équipes de développement web et mobile de savoir ou se diriger et ne pas réfléchir à un design pendant le développement et risquer de recommencer.

1. **Développement de notre solution**

Ayant chacun notre mission principale, le premier objectif à atteindre fut d’obtenir un POC. Chaque membre étant motivé par la tâche qui lui a été attribuée, nous avons pu assez rapidement avoir un premier échantillon avec une authentification fonctionnelle. Nous avons tous ainsi poursuit le développement de notre composant pour ainsi arriver sur la partie des actions et des réactions. Dans un premier temps, nous avons chacun développer ceci de notre côté, puis nous avons mis le tout en commun afin de lier le « front-end » avec le « back-end ». Une fois le tout lié, nous avons pu régler les derniers problèmes avant que notre produit soit fonctionnel.

Le développement de notre solution, première étape technique à laquelle nous sommes confrontés. Dans les grandes lignes cette étape s’est elle aussi très bien passée. Cependant, nous avons parfois manqué de connaissances sur les technologies qui ont été choisies, ce qui a entraîné un ralentissement dans le développement de notre solution. Par ailleurs, nous avons pris le temps de faire en sorte que notre solution soit accessible en respectant certaines règles comme la taille des boutons, la possibilité d’utilisation sans souris ou encore les « alt » d’images pris au sérieux.

1. **Mise en place des services**

Sachant que notre application était prête à accueillir les services que nous avons respectivement choisis, chaque membre du groupe peut désormais passer à l’intégration de son service. Un service se doit de disposer de sa propre authentification et de ses actions et/ou réactions.

Les services que nous avons décidé d’inclure à notre solution sont :

* **Discord** -> David BENISTANT,
* **Trello** -> Oscar FRANK,
* **Github** -> Alessio LIEBENGUTH,
* **Asana** -> Robin DENNI,
* **Google** -> Théotime SCHMELTZ,
* **Jira** -> Oscar FRANK & Robin DENNI.

Au cours de la mise en place des services choisis, nous avons rencontré quelques problèmes. Discord fut le premier service implémenté, puis nous avons mis en place le service Trello, la commence les problèmes. En effet, afin de récupérer les informations du webhook, ce dernier attendait un URL sur lequel il allait pouvoir renvoyer les différentes informations. Problème, nous développons en local… Un déploiement doit donc être mis en place afin que l’on puisse utiliser les webhook. Grâce à notre benchmark, nos technologies sont assez simples à déployer, ce qui nous a permis de certifier que notre choix de technologie était le bon. Tout une partie de GitHub actions a donc été mise en place afin que le déploiement se fasse automatiquement lorsqu’un « git push » est fait sur le dossier du projet.

Le second problème que quelques membres du groupe ont pu rencontrer est la complexité de certaines apis pour la connexion ou encore pour recevoir des actions ou envoyer des réactions.

Cependant, mise à part cela, l’intégration des services se déroule bien et assez rapidement.

1. **Documentation**

Pour notre projet, nous avons mis en place une documentation dite « utilisateur » ainsi qu’une documentation dite « technique ».

La documentation utilisateur reprend le tracé de la création d’un compte jusqu’à la mise en place d’AREA. Cette documentation est destinée aux futurs utilisateurs de notre solutions afin qu’ils bénéficient d’un accompagnement dans l’utilisation de celle-ci.

La documentation technique, quant à elle reprend en détail chaque composant de notre projet, à savoir web, mobile et l’API.

1. **Ce que l’on retient**

Suite à une discussion de groupe, nous avons remarqué que tout retient la même chose de cette expérience.

Tout d’abord, nous avons revu à quel point il est important pour une équipe de bien s’entendre afin de pouvoir échanger et questionner sans soucis.

Par ailleurs, nous avons pu constater à quel point utiliser des outils de gestion comme Jira, Trello ou encore Asana permet un bon suivi du projet, ce qui engendre une productivité beaucoup plus importante au sein du groupe.

Nous avons aussi remarqué qu’une répartition équitable des tâches permettait à tout le monde de travailler de manière égale sans que l’un s’ennuie pendant qu’un autre peine à avancer sur ses tâches.

Pour finir, nous avons pu relever que lorsque le projet est bien réparti, que chaque membre effectue son travail dans les temps et tient ses documents communs à jour, aucune tension n’est présente dans le groupe et le travail se fait dans une ambiance très agréable.