PS7
Rapport

AL-SD1

Julien N'Diaye

Axe AL : Langage Dédié
Axe SD : Tableau de Bord

Année 2020-2021

Sommaire

I - Mise en contexte

Grâce à la loi *Elan*, le maire de PolyVille a décidé de profiter de l'ORT (Opération de Revitalisation de Territoire) afin de dynamiser son centre-ville déserté depuis la construction d'un grand centre commercial à la sortie de la ville. Il veut doter les commerçants, les centres culturels et les habitants d'outils numériques de nouvelle génération. Ainsi, *PolyVilleActive* est un outil à destination des différents acteurs de la ville qui a pour objectif de les mettre en relation. L'application a pour but principal d'informer des activités, des promotions, des évènements, des places de stationnement disponibles ou encore du taux de fréquentation des différents lieux de la ville.

A travers ce projet conséquent, notre équipe a travaillé sur le développement d'une solution permettant à des animateurs non informaticiens de développer leurs propres interactions en fonction des positions de personnes ou d'autres événements de plus haut niveau. Nous avons donc mis l'accent sur le développement d'un langage dédié qui permet de réaliser de telles tâches.

De plus, afin de mesurer l'impact du système *PolyVilleActive*, il serait souhaitable de disposer d'un certain nombre d'indicateurs tels que le succès d'une opération, le taux de fréquentation d'un événement, le niveau de satisfaction des usagers, à destination principalement de la mairie de PolyVille, mais également à destination des commerçants, des centres culturels et des habitants.
Notre équipe a donc également travaillé sur la recherche et le développement d'indicateurs pertinents pour les différents acteurs de la ville. La source de ces données a également été pensée et développée.

II - Personas

Nous allons dans cette partie vous présenter les différents personas de *PolyvilleActive*.

Ainsi, nous en avons identifié quatre, ayant chacun des besoins précis, liés à nos axes de conception.

Persona n°1 : Habitante

Marine, 35 ans, est une habitante de PolyVille. Elle se rend régulièrement dans les différents commerces de la ville pour y effectuer toutes sortes d'achats.

Elle aimerait avoir accès à des informations utiles sur ses commerces favoris afin d'éviter les heures de fréquentation importante ou encore connaître les dernières promotions disponibles. De plus, il lui arrive régulièrement de ne pas trouver de place de parking disponible lorsqu'elle se rend en ville, ce qui lui fait perdre un temps considérable.

Persona n°2 : Commerçant

Vincent, 29 ans, est un jeune commerçant de PolyVille. Il possède une boutique de vêtements et souhaiterait développer son activité. Ainsi, avoir une vision globale des statistiques importantes de son commerce en temps réel lui serait d'une grande aide afin d'adapter sa stratégie de vente. De plus, il aimerait avoir la possibilité de créer des promotions et d'informer ses clients en les notifiant.

Persona n°3 : Employée de la mairie

Christine est une employée fidèle de la mairie de PolyVille. Elle y travaille depuis plus de 20 ans et est chargée du contrôle de l'organisation d'évènements culturels au sein de la ville.

Cependant, elle trouve que la procédure de validation des événements est assez contraignante et fastidieuse à cause des nombreuses vérifications à effectuer avant d'autoriser un événement. Ainsi, elle souhaiterait avoir un outil lui permettant d'automatiser la validation des évènements proposés par les différents acteurs de PolyVille. Cela lui permettrait de gagner un temps précieux et de contrôler pleinement la vie culturelle de la ville.

Persona n°4 : Acteur de la vie culturelle

Léo, 43 ans, est acteur de la vie culturelle de PolyVille. Il est musicien et souhaite proposer à la mairie d'organiser régulièrement des concerts ou des festivals en ville.
Cependant, il ne connaît pas les règles d'interdiction propres à chaque évènement. Il aimerait proposer directement via une interface ses événements sans avoir à se déplacer jusqu'à la mairie. De plus, il aimerait avoir des informations sur les événements qu'il a organisés afin d'adapter les prochains et ainsi attirer plus de personnes.

III - Notre solution

Comme indiqué précédemment, nous avons choisi les axes "*Langage Dédié*" en Architecture Logicielle et "*Tableau de bord*" en Sciences des Données.

Une fois nos personas identifiés de manière précise et les premières *User Stories* établies, nous avons pu commencer à réfléchir aux technologies que nous allions utiliser et implémenter afin de mener à bien notre projet.

1/ Choix des technologies

Les premières interrogations qui se plaçaient face à nous furent les suivantes: Quelle forme donner à notre solution ? Quelle interface choisir pour répondre au mieux à nos problématiques ? Site web ? Application mobile ? Ou encore une simple CLI ?

Au vu de notre axe Tableau de Bord, il était évident que nous allions manipuler des données, et donc devoir les afficher pour les utilisateurs de notre produit. Une interface en ligne de commande n'était donc pas adaptée à nos besoins. De plus, n'ayant pas choisi des axes orientés IHM, il n'était donc pas crucial d'avoir une application mobile pour démontrer notre solution fonctionnelle. Ainsi, nous avons fait le choix de développer un site web afin d'interagir avec nos utilisateurs.

Une fois le choix de l'interface fait, il a donc fallu choisir parmi les différents framework web accessibles en ligne. Quel était le plus approprié à nos besoins et le plus à notre portée ?

Après une discussion entre nous et une légère hésitation entre ***Angular*** et ***VueJS***, nous avons opté pour *Angular,* qui nous a semblé être le plus adapté à notre situation. En effet, ce choix nous a paru être le plus cohérent au vu de notre expérience sur le framework grâce à un précédent projet (PS6), contrairement à Vue JS qui était globalement moins bien maîtrisé par l'équipe.

La gestion de notre backend a également été un sujet de réflexion et il s'est avéré comme pour le choix d'***Angular***, que l'utilisation de ***NodeJS*** combiné à ***ExpressJS*** et ***JOI*** était la plus adaptée et la plus simple à implémenter.

Les données étant stockées dans des fichiers ***Mock*** (naturellement pour tout projet réalisé en local) de la même manière qu'un précédent projet, qui était une solution convenable pour le développement d'un produit minimum viable. A terme, si nous avions à développer de manière complète l'application et à la mettre en production, les données seraient hébergées sur un serveur distant.

Un autre choix technologique important que nous avons dû effectuer a été au sujet de notre dataset de départ. En effet, afin de résoudre la problématique de notre axe SD, nous avons dû établir un dataset de départ. N'ayant jamais réalisé cela auparavant, nos recherches ont convergé vers trois solutions possibles.

La première fut d'utiliser des OpenDataset déjà existants. Ce sont des bases de données libres de droit de certaines organisations avec des informations cohérentes dans un contexte spécifique et ainsi de les adapter à nos besoins.

La deuxième fut la possibilité d'écrire des scripts nous permettant de générer les données dont nous aurions besoin.

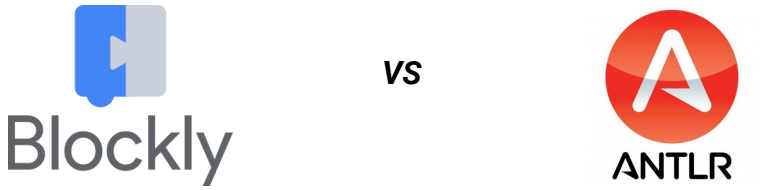
La dernière solution s'est avérée être la meilleure des trois. Elle consistait à utiliser *Datagenerator,* une application en ligne permettant de générer des données cohérentes en spécifiant les modèles de notre base de données et en choisissant des types spécifiques de données à générer. Nous avons donc opté pour celle-ci car c'était la plus simple et la plus adaptée à nos besoins. De plus, elle nous a fait gagner un temps considérable.

Ainsi, mis à part quelques légères hésitations, les réponses aux questions précédentes nous ont paru globalement assez évidentes à l'aide de notre expérience.

En revanche, le choix de la technologie nous permettant d'implémenter un langage dédié au sein de notre projet fut une réflexion complexe qui a mis un certain temps avant d'être résolue. En effet, aucun membre de notre équipe n'avait implémenté de langage dédié et cette notion, qui s'avérait assez vague pour nous tous. Il a donc fallu effectuer de nombreuses recherches afin de mieux assimiler la notion de "Langage Dédié", avant de faire un choix sur la technologie adaptée à notre problème. Ces recherches nous ont permis d'avoir une idée beaucoup plus précise de comment l'implémenter de manière cohérente au sein de notre site web.

Une fois la notion assimilée, nous avons pu débuter chacun de notre côté un benchmark des technologies adaptées à l'implémentation d'un langage dédié au cœur de notre projet.

Parmi plusieurs solutions, nous en avons retenu seulement deux qui semblaient les plus pertinentes au regard de nos besoins : **Blockly** et **ANTLR**.



Après une analyse plus approfondie des deux technologies, nous avons très vite identifié **Blockly** comme LA solution à nos besoins. En effet, la bibliothèque logicielle Blockly créée par Google possède une large documentation et est implémentable avec Angular, ce qui nous a également confirmé que le choix de notre framework était le bon. De plus, son utilisation est très intuitive puisqu'elle est similaire au langage de programmation *Scratch* qui permet, à l'aide de blocs d'instruction déjà définis, d'apprendre à des débutants les bases de la programmation.

A contrario, **ANTLR** permet principalement de générer des analyseurs lexicaux et est le plus souvent utilisé avec le langage JAVA, ce qui ne correspond pas vraiment avec la vision que nous avions de notre langage dédié.

Enfin, la possibilité de créer nos propres blocs d'instruction avec la bibliothèque Blockly à vraiment été le critère final qui nous a poussé à choisir cette technologie.

2/ Problématiques de nos axes

a - Axe Sciences des Données

La problématique majeure de notre axe Sciences des Données a été de trouver les indicateurs qui seraient pertinents et utiles pour nos personas. Ainsi, au cours du premier sprint, il nous a semblé cohérent de cibler la relation habitant-commerçant au vu du nombre de données intéressantes qui peuvent en découler. Afin de sélectionner les indicateurs pertinents, chaque membre de l'équipe a effectué des recherches sur le sujet, puis nous avons mis en commun nos idées au cours d'un brainstorming. Celui-ci nous a permis de garder uniquement les indicateurs cohérents, originaux et pertinents. En voici un résumé général, ainsi qu'une explication de leur utilité :

RÔLE	INDICATEUR	INTÉRÊT
Commerçant	Heures de fréquentation	Connaître les horaires de forte affluence
	Notation	Adapter la qualité de son service et de ses produits en fonction de sa notation
	Nombre de personnes près du commerce en temps réel	Connaître l'affluence en temps réel
	Pourcentage d'achats en fonction de l'âge	Adapter sa stratégie de vente
	Pourcentage d'achats en fonction du sexe	Adapter sa stratégie de vente
	Nombre de visites avant achat	Évaluer l'efficacité du commerce
	Taux de réachat	Évaluer la fidélité client
	Taux de conversion du commerce	Évaluer le ratio visiteur/acheteur
Habitant	Heures de fréquentation	Connaître les heures à éviter pour effectuer ses achats
	Notes de chaque commerce	Connaître les commerces de bonne qualité
	Promotions les plus appréciées par les consommateurs	Identifier les promotions qui valent le coup

Une fois les indicateurs identifiés, nous avons dû réfléchir à la façon dont ils allaient être récupérés et manipulés.

Dans un premier temps, nous avons axé notre réflexion sur les différents moyens permettant de récupérer les données qui serviront d'indicateur. Nous en avons conclu que le moyen le plus simple pour faire cela était de créer différents simulateurs au sein de notre site web. Le premier étant un simulateur de position qui permet de simuler le déplacement des habitants dans les différents commerces de la ville. A chaque déplacement, les données de localisation stockées dans notre backend sont mises à jour et elles nous permettent de savoir à chaque instant où se trouve chaque habitant. Ces données sont donc cruciales puisqu'elles nous permettent d'établir de nombreux indicateurs tel que la fréquentation.

Nous avons également créé pour chaque habitant et chaque magasin, une interface permettant d'indiquer quels achats ont été effectués. Grâce à cette interface, nous avons pu en déduire un nombre conséquent d'indicateurs important pour le commerçant.

Enfin, nous avons également offert la possibilité aux habitants de noter les commerces afin de modifier leur note.

Ainsi, le backend ne stocke que les données primaires, c'est dans notre front-end que nous les manipulons pour créer les indicateurs. Cette solution a été choisie car les données changent rapidement, il ne nous fallait pas surcharger notre backend de données inutiles. Cela implique, en revanche, que le front-end réalise toutes les opérations sur les données.

Dans un second temps, nous nous sommes concentrés sur la représentation optimale des données. Même si cette problématique est plutôt orientée IHM, il nous a paru important de réfléchir à cela afin de mettre en valeur nos indicateurs. Ainsi, l'utilisation de Chart Module dans Angular a joué un rôle décisif pour résoudre cette problématique.

b - Axe Architecture Logicielle

Du côté de l'axe Architecture Logicielle, la problématique majeure était de réfléchir à un langage dédié permettant à nos personas de créer des interactions avec le système de façon simple. Cette tâche a été la plus dure de toute au cours de notre projet et nous reviendrons en détail, dans la partie suivante, sur les problèmes auxquels nous avons été confrontés.

Dès lors que l'idée d'un langage dédié cohérent à nos besoins fut trouvée, nous avons pu commencer à l'implémenter à travers la technologie Blockly. L'idée est axée sur la relation employé de mairie-acteur de vie culturelle. Elle consiste à permettre aux employés de mairie d'établir des règles d'interdiction sur la création d'évènements dans la ville. En d'autres termes, notre langage dédié permet de définir un ensemble de règles cohérentes entre elles qui empêcheront les acteurs de vie culturelle de créer des événements selon certains critères spécifiques. Un exemple pour l'illustrer serait la possibilité pour un employé, de créer une règle qui interdit l'organisation de concert dans la ville si le public visé est majoritairement très jeunes **ET** si l'heure de fin de l'événement dépasse les 2h du matin.

Le point fort de notre langage est sa capacité à détecter lors de la création d'une nouvelle règle, si cette dernière est syntaxiquement correcte, si elle n'existe pas déjà ou si elle entre en conflit avec une autre règle déjà créée.

Au final, nous sommes plutôt fiers du résultat final du langage que nous avons développé dans le temps imparti et compte tenu des difficultés auxquelles nous avons été confrontés. Nous n'avions aucune expérience sur ce sujet, et la solution que nous avons développée nous semble totalement répondre à la problématique de l'axe. En effet, le langage est à la fois intuitif, fonctionnel et permet aux employés de mairie de créer un ensemble de règles diversifiées et cohérentes entre elles.

3/ Difficultés rencontrées

La principale difficulté rencontrée au cours de ce projet a été l'assimilation de la notion de langage dédié. En effet, nous avons pris du temps pour comprendre son utilité auprès des utilisateurs et en quoi il est différent d'un formulaire classique.

Au début du projet, par manque de communication, nous sommes partis sur le développement d'une idée de langage dédié qui se rapprochait en réalité plus d'un formulaire. Cette erreur nous a fait perdre un temps considérable. C'est pourquoi nous avons dû redoubler d'efforts afin de rattraper notre retard et d'implémenter une nouvelle solution cohérente avec cet axe. Après de nombreuses discussions en équipe et une réflexion importante, nous avons finalement trouvé une solution définissant un réel langage dédié, à destination des employés de la mairie de PolyVille, et réalisable dans le temps qu'il nous restait.

Seule problématique, la solution n'est pas vraiment liée à la partie de l'axe SD que nous avions déjà bien avancée. En effet, notre implémentation de l'axe SD est plus orientée vers la relation habitant-commerçant, tandis que notre axe AL est lui destiné à la relation employé de mairie/acteur de vie culturelle. Compte tenu du timing, nous avons pris la décision, après une longue discussion avec notre client, de réaliser tout de même cette solution au détriment idéalement d'un mariage des axes. Nous avons en effet privilégié la livraison d'une solution cohérente pour chacun de nos axes, plutôt que de risquer une livraison d'une solution non fonctionnelle.

4/ Perspectives d’améliorations

Comme expliqué dans la partie précédente, le mariage des axes n’est pas optimal, c’est pourquoi parmi les user stories à réaliser au cours du sprint 4, nous en avons défini plusieurs dans le but de rendre notre produit plus homogène. Elles visent, pour la plupart, à développer des indicateurs utiles portant sur les différents évènements organisés dans PolyVille et destinés aux employés de mairie et aux acteurs. Ces user stories permettraient donc de faire un lien entre la création/validation des évènements et la visualisation d'indicateurs sur ceux-ci, tel que le nombre de participant, la satisfaction des participants ou encore leur tranche d’âge.

Une autre fonction que nous aurions pu implémenter avec plus de temps, aurait été de pouvoir offrir aux commerçants, la possibilité de contrôler leur propre tableau de bord de données à l’aide d’un langage dédié. Autrement dit, afficher des données cohérentes sur le tableau de bord vis-à-vis des règles établies par le commerçant par l’intermédiaire d’un langage dédié.

IV - Notre innovation

Notre innovation met en place un système de parking assez avancé qui permet d'obtenir un bon compromis entre places payantes et gratuites.

Nous stockons chaque place de parking d'un magasin dans les fichiers mocks des commerçants du serveur. Fichiers qui pourront être mis à jour lorsque cela sera nécessaire. Nous avons opté pour ce choix, car étant donné l'appartenance des places aux commerçants, il était plus simple et plus cohérent d'implémenter notre solution de cette façon.
Pour les habitants ou les visiteurs qui veulent s'y garer, une interface visuelle est mise à leur disposition pour qu'ils aient un plan assez clair de la disposition des places. Ce plan des places utilise les packages d'extension *bootstrap* d'angular.

De plus, une fois la place de parking choisie, c'est-à-dire si quelqu'un l'a réservée ou bien si quelqu'un y stationne son véhicule, l'information est envoyée au serveur. Ainsi, la place change de couleur visuellement, ce qui indique qu'elle n'est plus disponible.

Sans oublier que lorsqu'une personne a effectué une réservation et qu'elle effectue un achat dans le magasin concerné, son stationnement sera gratuit.

Les commerçants ont le choix d'appliquer la réduction qu'ils veulent sur la place de parking.

En regardant de plus près, notre innovation ne nous a pas posé de gros problèmes. En effet, l'architecture de notre projet aurait pu se révéler incompatible avec celle de notre innovation, mais la bonne conception de notre application nous a épargné de ce risque.

Enfin, lorsque l'utilisateur quitte la place, le système le détecte automatiquement et l'information que la place est dès à présent disponible est envoyée au serveur pour mettre à jour la base de données.

V - Qualité de notre projet

1/ Test

Pour les tests de notre application, nous avons procédé de deux manières différentes.

Tout d'abord, nous avons fait des tests informatisés. Pour se faire, nous avons utilisé les technologies Karma et Jasmine. Les tests que nous avons faits sur les composants de notre application web avaient pour objectif de vérifier la bonne adéquation de nos composants. Ceci pour s'assurer qu'ils ne comportaient pas d'erreurs intrinsèques à leur fonctionnement.

Face à court délai, nous avons décidé de ne pas consacrer trop de temps à l'apprentissage de ces nouvelles technologies que sont Karma et Jasmine. C'est pourquoi nous avons choisi de faire des tests manuels. Chaque jour, une personne était désignée pour tester une configuration et s'assurer de son bon fonctionnement. Si nos tests, informatisés ou non, détectaient une erreur, une correction était appliquée aussitôt.

Pour notre backend, nous avons utilisé Postman pour faire des vérifications manuelles sur la gestion des données de notre serveur. Compte tenu du caractère "simple" et immuable de notre serveur, l'utilisation de Postman en tant qu'outil vérificateur de celui-ci nous a paru une bonne solution.

2/ Documentation

La documentation est, tout comme les tests, un élément important dans un projet informatique. Elle l'est encore davantage dans la réalisation d'un langage dédié.

C'est pourquoi, il nous a paru évident de fournir une documentation détaillée de l'utilisation de notre langage dédié, à destination de ceux qui vont l'utiliser. Elle permet de décrire en détail ce que chaque bloc d'instruction permet de faire, comment le langage dédié s'utilise généralement mais également de rappeler les règles à respecter pour ne pas rencontrer d'erreur.

De plus, nous avons jugé important de réaliser une documentation de notre code, pour les prochains développeurs qui effectueront le sprint 4.

3/ Gestion agile du projet

La gestion de notre projet de manière agile nous a permis d'identifier de manière précise le travail à effectuer mais également de maintenir une bonne organisation sur la durée. Les fonctionnalités de GitHub ont été d'une grande aide dans la gestion et l'organisation du projet et de son **backlog** puisque dès le départ nous avons pu représenter chaque **user story** par une *issue* et indiquer ses critères d'acceptation en description.

De plus, chaque **sprint** a été défini par une *milestone* dans GitHub à laquelle nous avons assigné les issues à réaliser pendant celui-ci. La deadline indiquée sur le milestone ainsi que son pourcentage de complétion nous ont permis à chaque instant d'avoir une vision globale de l'avancement du sprint, et ainsi de prendre des décisions en conséquence.

Les *labels* de GitHub nous ont également été d'une grande utilité dans l'organisation et la visualisation des user stories les plus importantes à réaliser au cours du projet. En effet, nous les avons principalement utilisés pour chaque user story afin d'indiquer leur degré de priorité (établi à l'aide de la méthode **MoSCow**), la difficulté de leur réalisation (grâce au **T-shirt sizing**) ainsi que l'axe de développement auquel elles appartiennent.

Nous avons également utilisé la méthode **Kanban** à travers un *GitHub Project*, qui nous a permis d'avoir une vision précise de l'avancement du sprint en cours et . En effet, chaque membre de l'équipe pouvait savoir en quelques secondes quelles étaient les issues à réaliser, celles en cours de réalisation ou encore celles finalisées.

Enfin, à chaque fin de sprint, nous avons fait le choix de produire une *release* sur la livraison afin de garder les différentes versions stables de notre produit.

VI - Vision critique

1/ Limites

Notre solution comporte certaines limites qui découlent du fait de la nature de notre application.
D'une part, une partie des indicateurs qu'elle offre aux commerçants se basent sur le bon vouloir et l'honnêteté des gens. En effet, quand un utilisateur notifie l'application de son achat, nous ne pouvons pas vérifier l'information et sommes obligés de le croire.

D'autre part, le nombre important d'indicateurs que nous mettons à dispositions des commerçants à tendance à enlever un peu d'intuitivité à l'application. Plus il y a d'indicateurs, plus la page est dense et plus il est difficile de s'y repérer.

À cela s'ajoute l'impossibilité de récupérer le compte potentiellement perdu d'un utilisateur. Nous n'avons pas fait de système de récupération de compte. Or, étant donné la nécessité pour un utilisateur de d'identifier, l'application lui sera inutile s'il perd son ID.

2/ Si c'était à refaire

Pour ce qui est du choix des technologies utilisées, nous avons effectué dès le départ une bonne analyse de nos besoins et des technologies adaptées à ceux-ci. Si c'était à refaire, nous ferions donc les mêmes choix technologiques.

Si nous devions refaire notre projet, nous nous pencherions un peu plus sur les problèmes utilisateurs. Par exemple, faire en sorte que les promotions de la ville soient accessibles sans forcément se connecter pour ne pas bloquer un visiteur ou un habitant qui aurait malencontreusement perdu son id de connexion.

Dû à des contraintes de temps, notre solution n'a pas totalement marié ses axes de développement. En effet, l'axe "Tableau de bord" et l'axe "Langage Dédié" ont été insuffisamment mis en relation. Nous nous sommes beaucoup focalisés sur la relation habitant/commerçant alors que le langage dédié intervient plus pour les employés de mairie et les acteurs de vie culturel.

Enfin, nous créerons une map de la ville pour que les habitants aient une idée plus précise de la localisation d'un magasin car l'unique affichage de sa longitude et sa latitude peut être compliqué à comprendre.

VII - Organisation du travail

Chaque début de semaine , l'équipe organisait une petite réunion pour déterminer les tâches à effectuer pendant la semaine. Cette réunion de début de semaine permettait aussi de s’assurer que les tests manuels et informatisés passaient sans soucis.

De plus, chaque jour, nous travaillions sur une issue que nous prenions sur github. Nous organisions également une réunion lorsque une issue nous paraissait déséquilibrée ou non adaptée.

À la fin de chaque jour, nous déposions sur un channel slack ce que nous avons fait, ce que nous faisons et ce qu’il reste à faire. Par la même occasion, nous nous assurions que personne n’était bloqué et par conséquent que tout le monde pouvait continuer son travail.

Le week-end nous permettait de faire une conversation vidéo d’une durée d’une demi-journée pour revenir sur ce que nous avons fait durant la semaine et sur les potentiels problèmes que nous avons réglés.

Enfin, pour bien avancer sur nos axes, nous nous sommes répartis les tâches de façon à répartir les axes par binôme de travail.

Toutefois, nos multiples réunions pendant la semaine nous ont permis de ne pas diviser l’équipe et de tous nous tenir informés, cela même lorsque nous ne travaillions pas sur les mêmes axes.

Conclusion

Ce projet était très enrichissant et nous a donné un aperçu du métier d'Ingénieur en Architecture Logicielle ainsi qu'en Science des Données.
Nous avons appris tout au long de ce projet, l'importance de la bonne répartition des tâches et de la mise en place de qualité qui se fait dès le début par l'intermédiaire de tests.

De plus, la prise d'initiative et le travail en équipe ont été capitaux dans la réalisation de ce projet et ce sont des aspects essentiels qui nous serviront dans notre futur métier d'ingénieur.

Puisque les temps impartis à la réalisation du projet furent brefs, il nous a fallu faire preuve de flexibilité et de persévérance pour s'y adapter. C'est pourquoi nous avons fait certaines concessions pour pouvoir livrer notre projet à temps avec de la valeur.

Enfin, nous nous sommes tous investis dans ce projet et nous avons travaillé avec grande rigueur. Même si des différences de niveau était notable, l'implication des membres de l'équipe ps7-al-sd1 fut globalement la même pour tout le monde.