



Formation : Analyste développeur

Projet Angular

Le bateau de Thibault

Nicolas LEJEUNE
Gabriel LEFFAD
Matthieu POIRIER

SOMMAIRE

Contexte du projet.....	2
- Présentation de l'équipe.....	2
- Expression du besoin.....	3
- Solution proposée.....	4
Points techniques.....	5
- Choix des technologies.....	5
- Gestion de projet.....	6
- Nos ressources.....	6
- Outils et déroulement du projet.....	6
- Architecture logicielle.....	9
- Vues.....	11
Conclusion.....	13

Contexte du projet

Présentation de l'équipe



Nicolas LEJEUNE

Titulaire d'une licence en informatique de l'université Paris Diderot, je suis particulièrement spécialisé en C, JAVA, et Python.

Gabriel LEFFAD

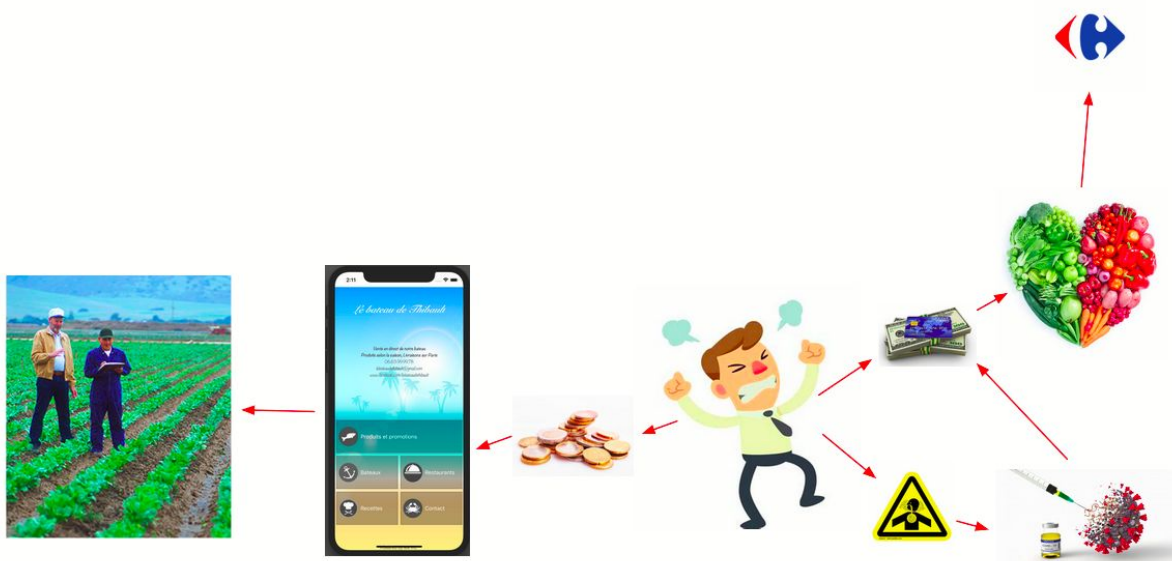
Titulaire d'une licence en informatique de l'université Paris Diderot, je suis particulièrement spécialisé en C, JAVA, et Python.



Matthieu POIRIER

Titulaire d'une licence en Informatique, je suis particulièrement spécialisé en PHP/ Symfony.

Expression du besoin



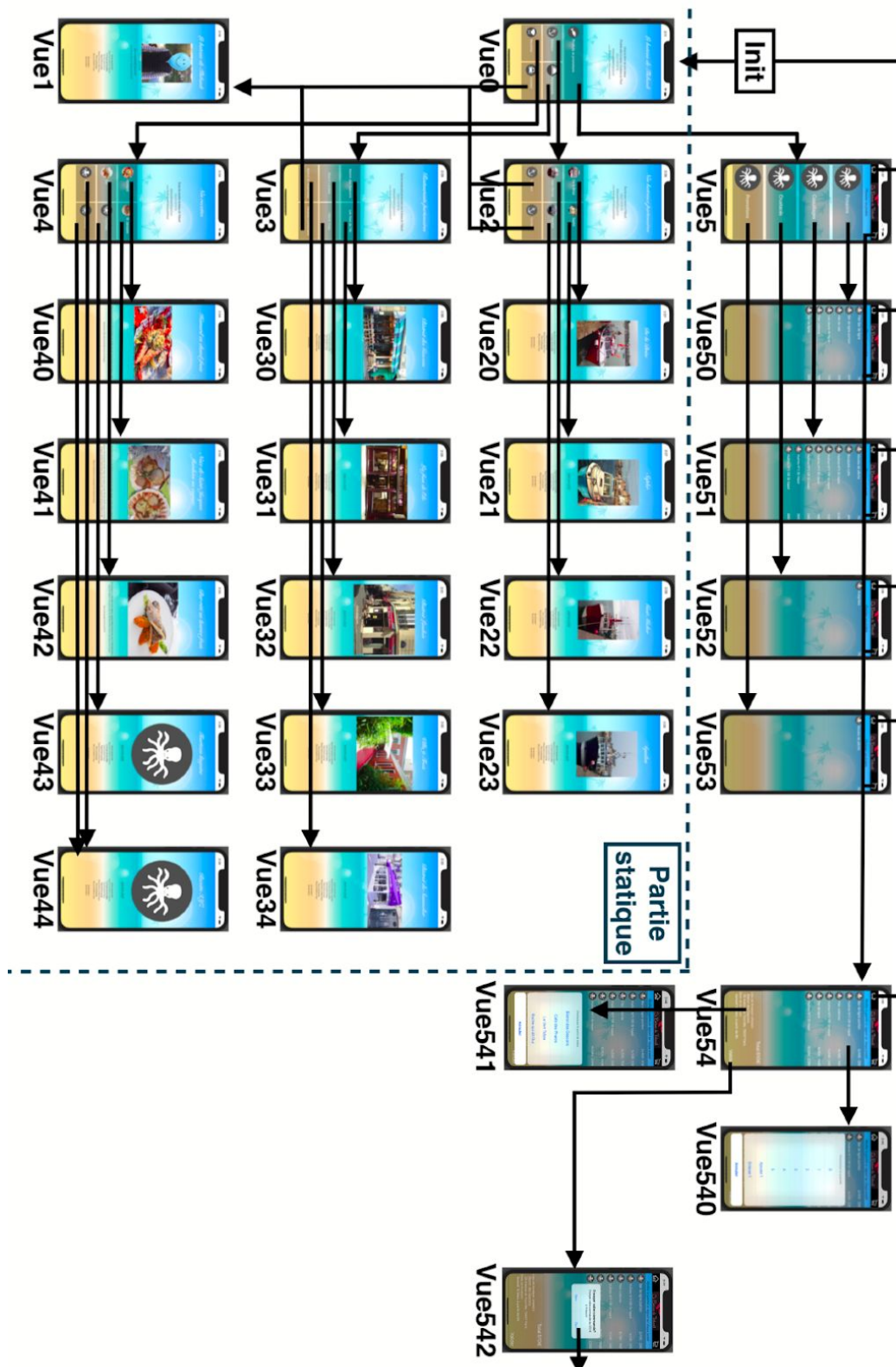
Les consommateurs font aujourd'hui face à différentes problématiques : des prix élevés, des produits industriels dans les supermarchés remplis de conservateurs, et de l'autre côté, des agriculteurs qui reçoivent une infime partie de la valeur finale de la production.

L'application "Le bateau de Thibault" permet de relier directement le client en circuit court au producteur en court-circuitant les revendeurs. Ainsi, le client obtient une traçabilité du produit, une meilleure qualité, le tout à un coût souvent réduit. Et de l'autre côté, le producteur augmente ses marges ce qui lui permet de mieux vivre son activité. De plus, en période de coronavirus, le client s'expose à un risque de contamination en allant au supermarché, notamment les personnes à risque, obligées de faire leurs courses. Cette application peut donc permettre aux personnes à risque de limiter leur risque d'infection.

Thibault a décidé de développer une refonte d'une application de prise de rendez-vous commercial pour une entreprise de type AMAP. Un utilisateur est un client d'un circuit ultra-court de produits issus de la pêche en Normandie (du producteur directement au consommateur).

Solution proposée

Nous proposons donc une implémentation de l'application suivant ces planches représentant l'interface utilisateur :



Points techniques

Choix des technologies



Pour ce projet, nous avons utilisé Angular afin de réaliser cette application. Développé par Google, Angular est un Framework open source écrit en JavaScript qui permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « Single Page Applications » : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action.

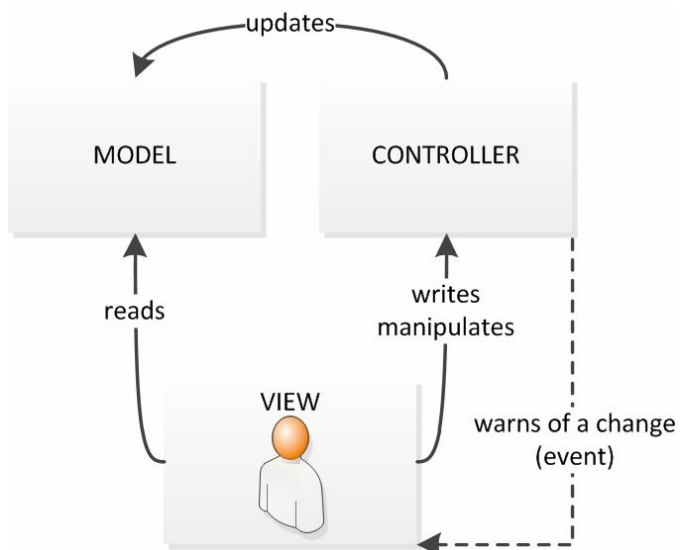
Le Framework est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités. Un type d'architecture qui a largement fait ses preuves et qui permet une forte maintenabilité et une amélioration du travail collaboratif.

Angular est une réécriture complète de AngularJS, framework construit par la même équipe.

Modèle-vue-contrôleur ou MVC est un motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

Un modèle (Model) contient les données à afficher. Une vue (View) contient la présentation de l'interface graphique. Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur. Ce motif est utilisé par de nombreux frameworks pour

applications web tels que Ruby on Rails, Grails, ASP.NET MVC, Spring, Struts, Symfony, Apache Tapestry, Laravel, ou AngularJS.

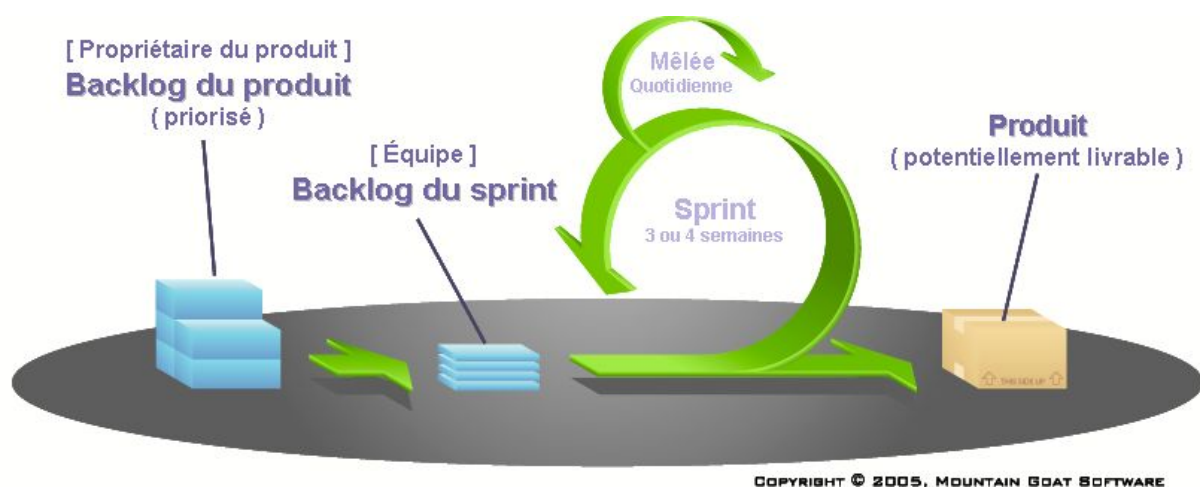


Gestion de projet

Nos ressources

Dû à un projet en React en parallèle avec des deadline très rapprochées, et les premières journées dédiées à la découverte, nous disposons de $3,5\text{JH} \times 3 = 10,5\text{JH}$ et développeurs sur trois ayant aucune expérience en développement Web.

Outils et déroulement du projet



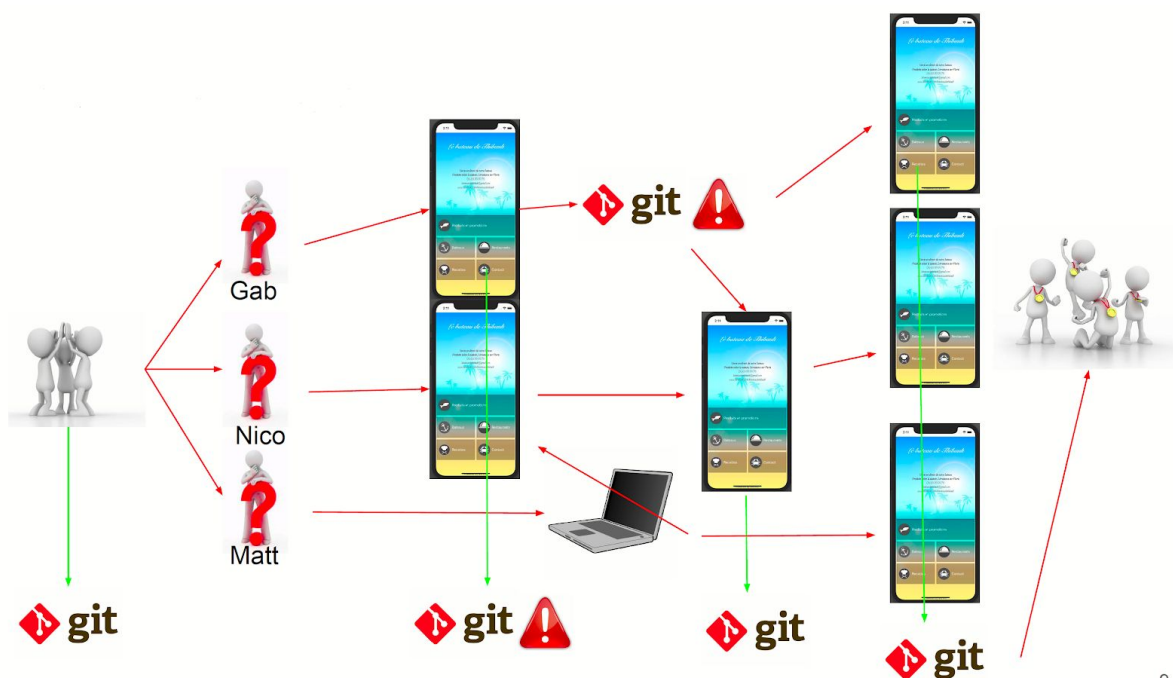
Nous avons opté pour une organisation inspirée du framework Scrum. Il s'agit d'un framework ou cadre de développement de produits logiciels complexes. Il est défini par ses créateurs comme un « cadre de travail holistique itératif qui se concentre sur les buts communs en livrant de manière productive et créative des produits de la plus grande valeur possible ». Scrum est considéré comme un groupe de pratiques répondant pour la plupart aux préconisations du Manifeste agile.

Scrum s'appuie sur le découpage d'un projet en « boîtes de temps », nommées sprints (« pointes de vitesse »). Les sprints peuvent durer entre quelques heures et un mois (avec un sprint médian à deux semaines). Chaque sprint commence par une estimation suivie d'une planification opérationnelle. Le sprint se termine par une démonstration de ce qui a été achevé. Avant de démarrer un nouveau sprint, l'équipe réalise une rétrospective. Cette technique analyse le déroulement du sprint achevé, afin d'améliorer ses pratiques. Le flux de travail de

l'équipe de développement est facilitée par son auto-organisation, il n'y aura donc pas de gestionnaire de projet.

La création de frameworks de développement logiciels hybrides couplant Scrum et d'autres frameworks est commune puisque Scrum ne couvre pas le cycle de développement de produit. Par exemple, on pourra utiliser des pratiques issues de l'extreme programming, de la phase de construction structurée de la méthode RAD, ou un ensemble de pratiques de qualité du logiciel issues du vécu de l'équipe projet.

Au fil du projet, nous avons décidé de nous munir d'un repository Git, et d'un Discord, nous permettant de fluidifier le travail à plusieurs.



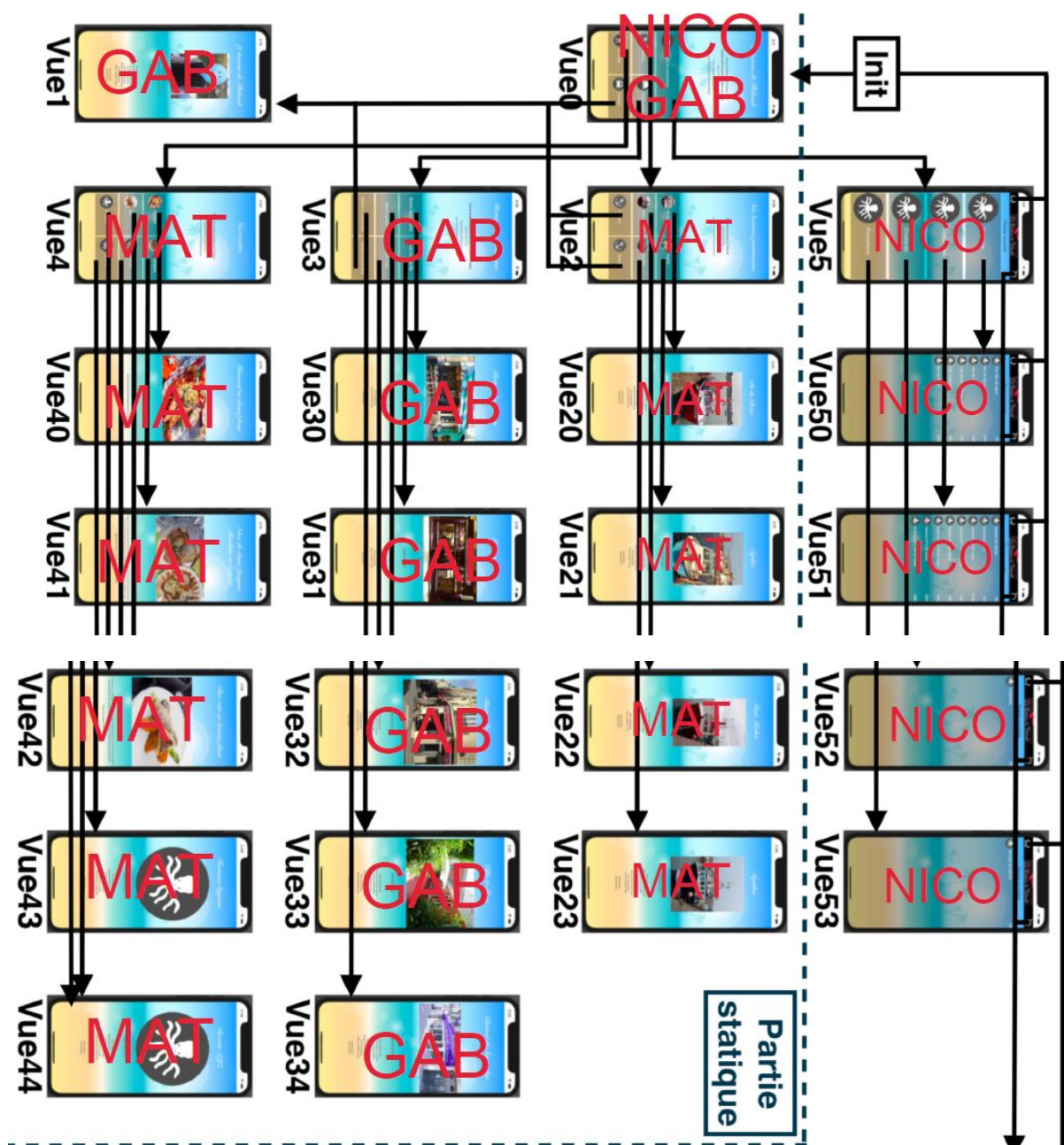
9

Le premier sprint consistait à découvrir Angular en essayant de créer la page Home. À la fin de chaque sprint, nous avons effectué une sprint rétrospective permettant de faire un point sur les aspects qui ont bien marché et ceux qui ont posé problème. Ceci nous a permis de régler un certain nombre de problèmes tout en faisant connaissance avec le langage, et les membres de l'équipe, leurs préférences, et leur fonctionnement.

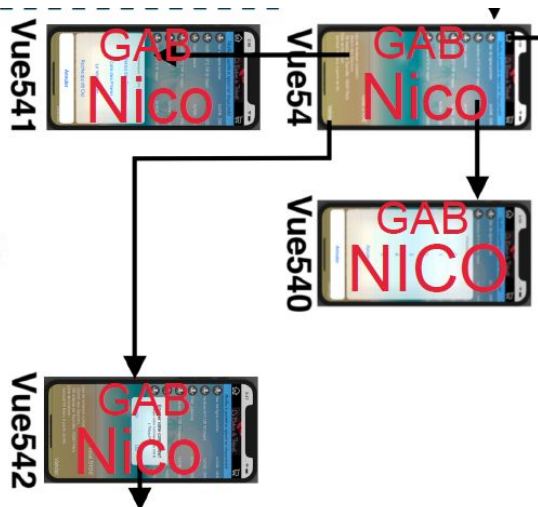
Nous avons aussi dû faire face à un problème matériel du côté de Matthieu, qui n'avait pas d'ordinateur durant deux jours. De plus, nous avons eu un

souci de conflit lors des premiers push sur le git entre le travail push et le travail en local sur chacune des trois machines utilisées. Le problème en local a été résolu avec ½ JH (nous étions limité par la vitesse maximale de copie des disques dur en notre possession). Cette méthode nous a permis de bien gérer l’attribution des ressources, ce qui a diminuer le coût des imprévu sur le résultat final. Nous avons donc pu apprendre les base du développement Web, découvrir Angular en 10,5JH, malgré les imprévus.

La répartition des tâches s’est donc effectué ainsi :



GESTION GIT : GAB
PRESENTATION : GAB



Architecture logicielle

Nous avons décidé de diviser l'application en 4 parties : **pages**, **components**, **services** et **assets**..

Pages

→ Différentes vues de l'application

Components

→ Différents outils reproductibles

Assets/Data

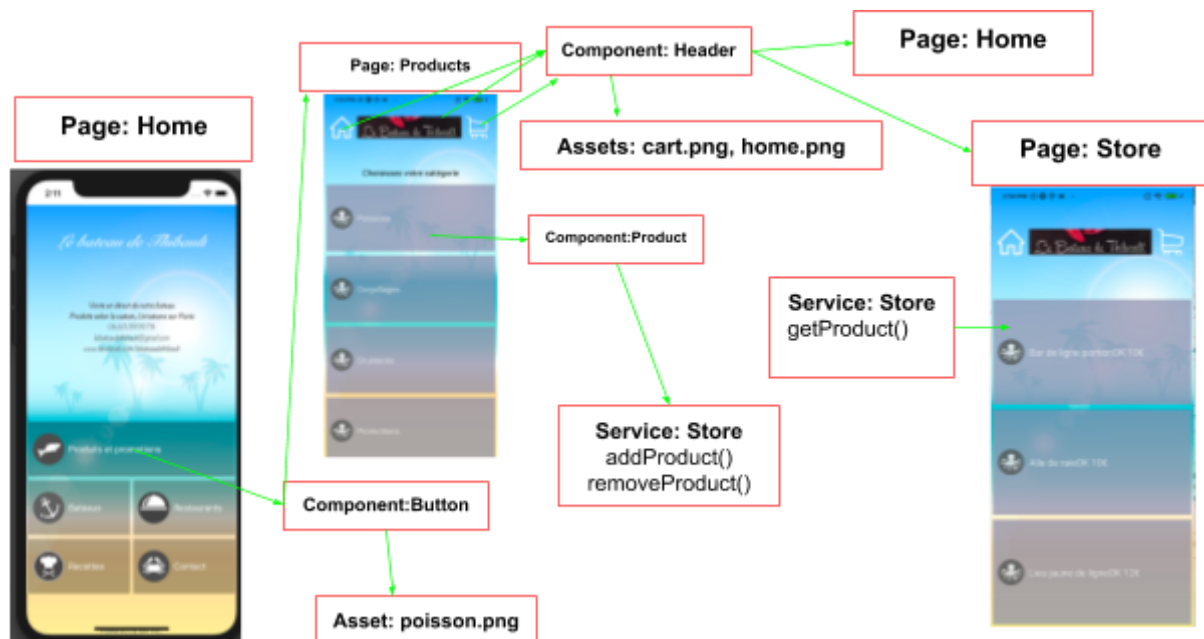
→ Stockage des images et données au format JSON

Services

→ Récupération des données via HTTP
→ Gestion du panier de commande

- Les pages représentent les différentes vues de l'application. Ainsi, lorsqu'une navigation se fait au sein de l'application, on procède à un changement de page.
- Les composants sont les différents outils reproductibles utilisés au sein de l'application. Dans notre application, ils sont au nombre de trois à savoir :
 - Header : barre d'outil avec accès rapide à la page d'accueil et au panier de commande.
 - Button : bouton utilisé pour la navigation avec passage de données entre les pages si nécessaire.
 - Product : bouton représentant un produit achetable. Une pression dessus ajoute le produit correspondant au panier et une autre le retire.
- Deux services sont utilisés dans l'application :
 - WebService : ce service permet la récupération des données sur un serveur ou en local via une requête HTTP.
 - StoreService : ce service permet de gérer le panier de commande via le module IonicStorageModule.
- Les assets / data sont stockés dans le dossier assets et sont donc les différentes images utilisées pour l'application ainsi que les fichiers .json contenant des données de l'application.

Vues



Voici un exemple des différents appels effectués lors de l'utilisation de l'application.

Ici, nous partons de la page *Home* qui est la page principale de l'application. Nous pouvons voir que cette page est composée de plusieurs boutons qui vont être utilisés pour la navigation. Nous avons donc une utilisation du component *Button* qui utilise lui-même un asset, ici *poisson.png*.

En appuyant sur produits nous arrivons sur la page *Products* qui est composé des différents produits en fonction de la catégorie choisie. Nous pouvons voir que cette page contient une barre de navigation, donc une utilisation du component *Header*. Deux images sont également utilisées pour le header, à savoir, *cart.png* et *home.png*. Cette page est également composée de boutons. Ces boutons sont des components *Product*. Ainsi, lorsque l'on appui sur l'un d'eux, le produit sera ajouté au panier, et si l'on appui une nouvelle fois, le produit sera retiré du panier. Ce component utilise donc le service *Store* et ses fonctions *addProduct()* et *removeProduct*.

Nous arrivons donc sur le panier de commande. Cette page contient tous les produits que nous avons ajouté au panier depuis la page précédente. Pour cela, elle utilise le service *Store* et sa fonction *getProduct()*.

Nous avons donc vu les différentes utilisations des composants et des services par les pages lors d'une utilisation de l'application.

Conclusion

Cette expérience nous a permis de découvrir la programmation Web, et de mettre à profit les outils de travail et de communication tel que ceux utilisés au fil de ce projet.

Au delà de l'aspect technique nous avons pu voir que même dans un cas de figure où les compétences techniques ne sont pas présentes, grâce à une bonne méthode de gestion de projet, on parvient à bien gérer les ressources, à faire face aux imprévus, et utiliser le temps plus efficacement.

De plus, la découverte de Angular, nous a permis de gagner en autonomie, en méthodologie d'auto-formation et de résolution de problème, ce qui nous sera capital pour la suite de notre carrière.