Projet refonte du Crew Web Access

Introduction

Le monde se modernise rapidement, et le besoin de suivre le pas afin de rester dans la course se fait ressentir ; Pour cela, comme vous le savez, une refonte totale de la solution CWA (Crew Web Access) a été initiée.

Après avoir analysé le POC (Proof Of Concept) existant de l'application mobile, voici mes conclusions :

- Le choix des technologies utilisées est bon et en adéquation avec le besoin .
- Quelques erreurs liées à la sécurité
- Aucune architecture mise en place (hormis celle imposée par la technologie)

Le dernier point est la cause principale de ma volonté de reprendre de zéro. En effet, si on ne réfléchit pas bien à l'architecture (l'organisation) du code, on finit toujours par tomber, au fil du temps, sur les mêmes problématiques qu'avec l'ancien KEOPS : manque de lisibilité du code, donc maintenabilité difficile et chronophage.

Afin d'être certain de ne pas faire fausse route dès le début, j'ai décidé de faire l'analyse du besoin et le contour fonctionnel de l'application. De plus, j'ai élaboré l'architecture des classes constituant l'application. Le projet étant de petite envergure, il est encore temps de pouvoir tout reprendre de zéro.

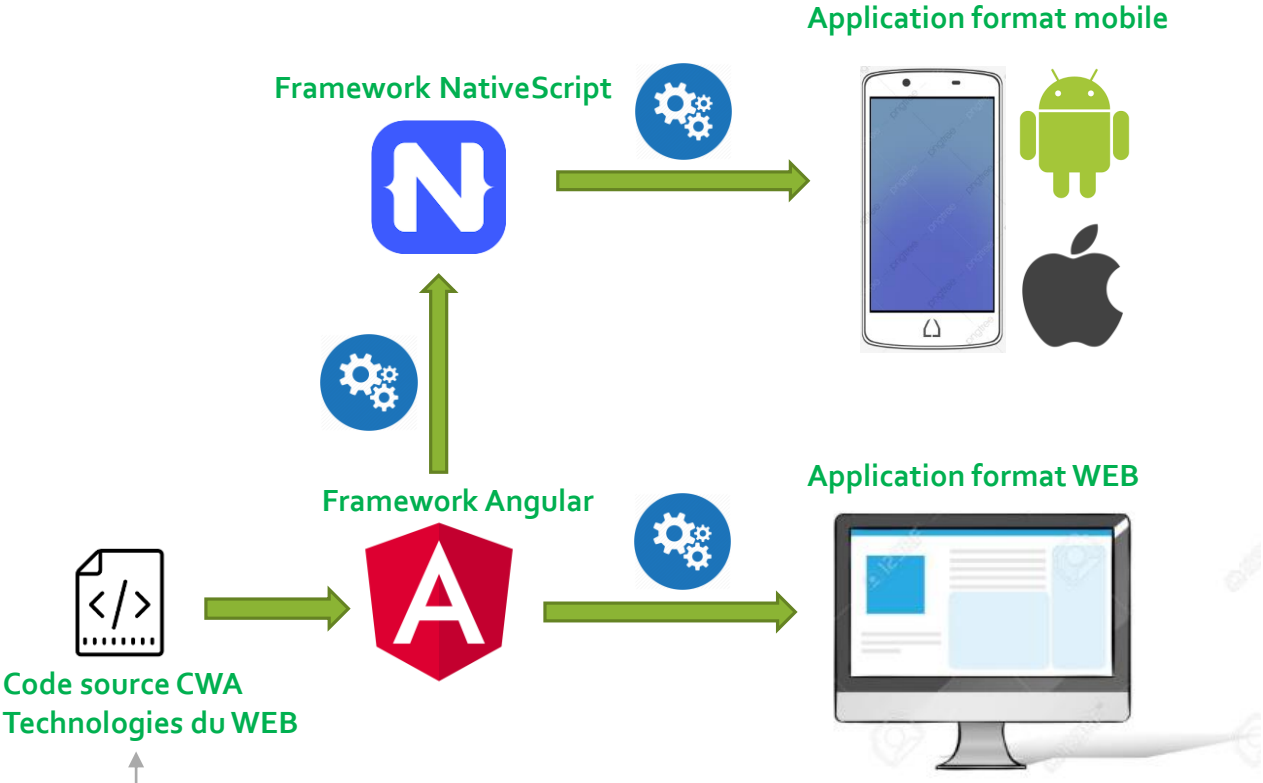
Ce dont parle ce document

Le choix des technologies

Angular permet de développer des application WEB (site web) de manière rapide et efficace.

NativeScript permet de créer des applications native pour mobile (Android et IOS) en utilisant des technologies WEB.

Principe de fonctionnement :



Peu importe la nature du livrable, le code source ne change pas. Seules les interfaces sont différentes.

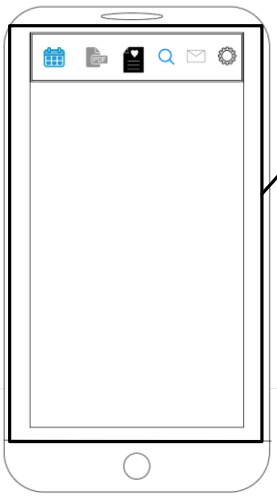
Pourquoi choisir ces technologies ?
J'ai une bonne connaissance du FrameWork Angular
Permettent de générer des livrables pour mobile ET navigateur WEB en utilisant le même code source (moins de travail)
Sont bien pensées et permettent de moduler le code à convenance pour une meilleure maintenabilité de l'application

Maquettage des interfaces

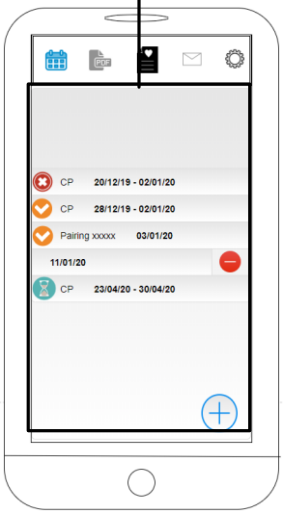
Ci-dessous, les croquis des interfaces utilisateurs. La technologie employée (Angular) fonctionne avec le principe de composant. Chaque composant représente une page ou une partie de la page. Pour faciliter l'élaboration du modèle conceptuel, j'ai identifié les différents composants constituant l'application (interfaces non contractuelles) :

cwa-login

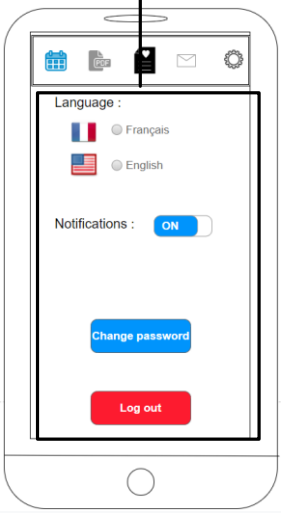
cwa-root



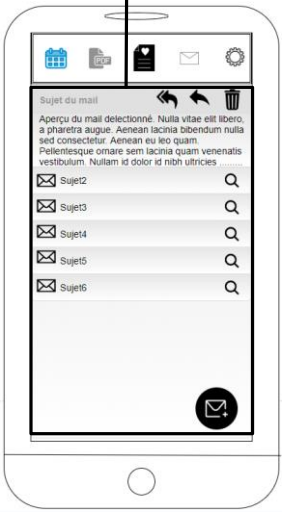
cwa-wishes-list



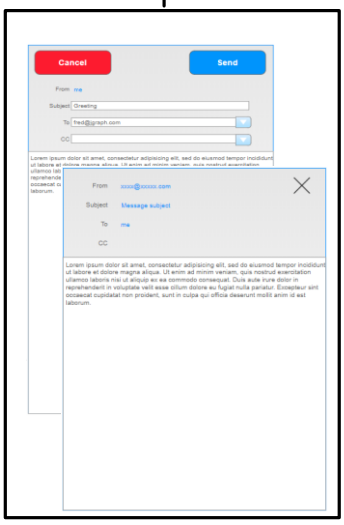
cwa-settings



cwa-messages-list

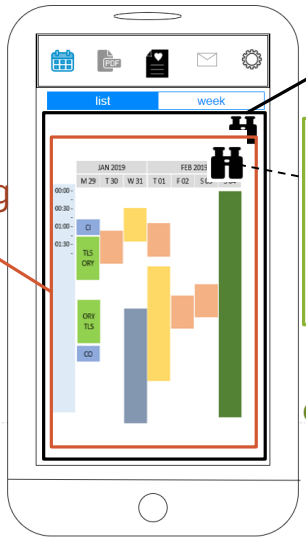


cwa-message



cwa-planning

cwa-week-planning

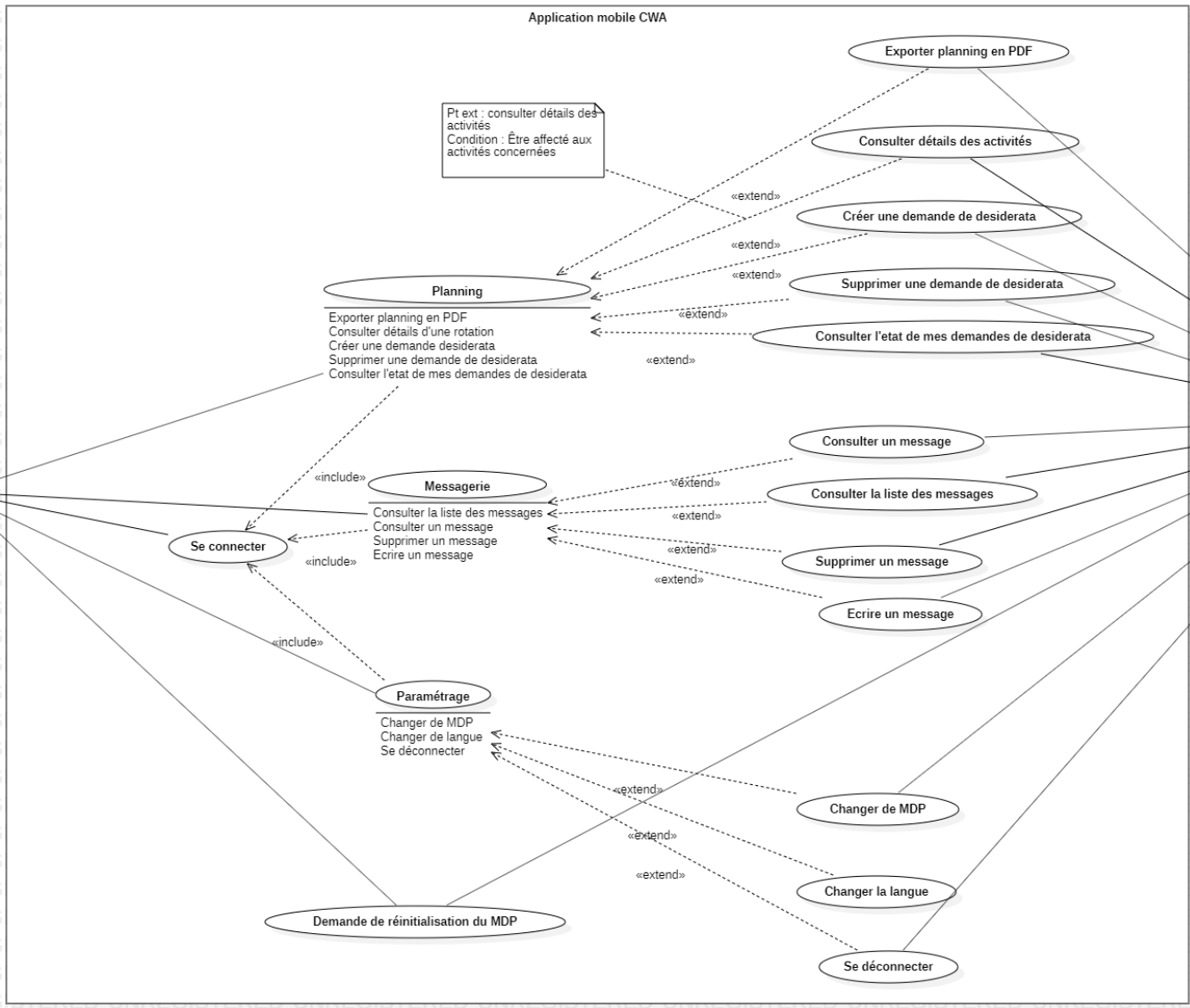


cwa-month-planning

cwa-list-planning

Le contour fonctionnel

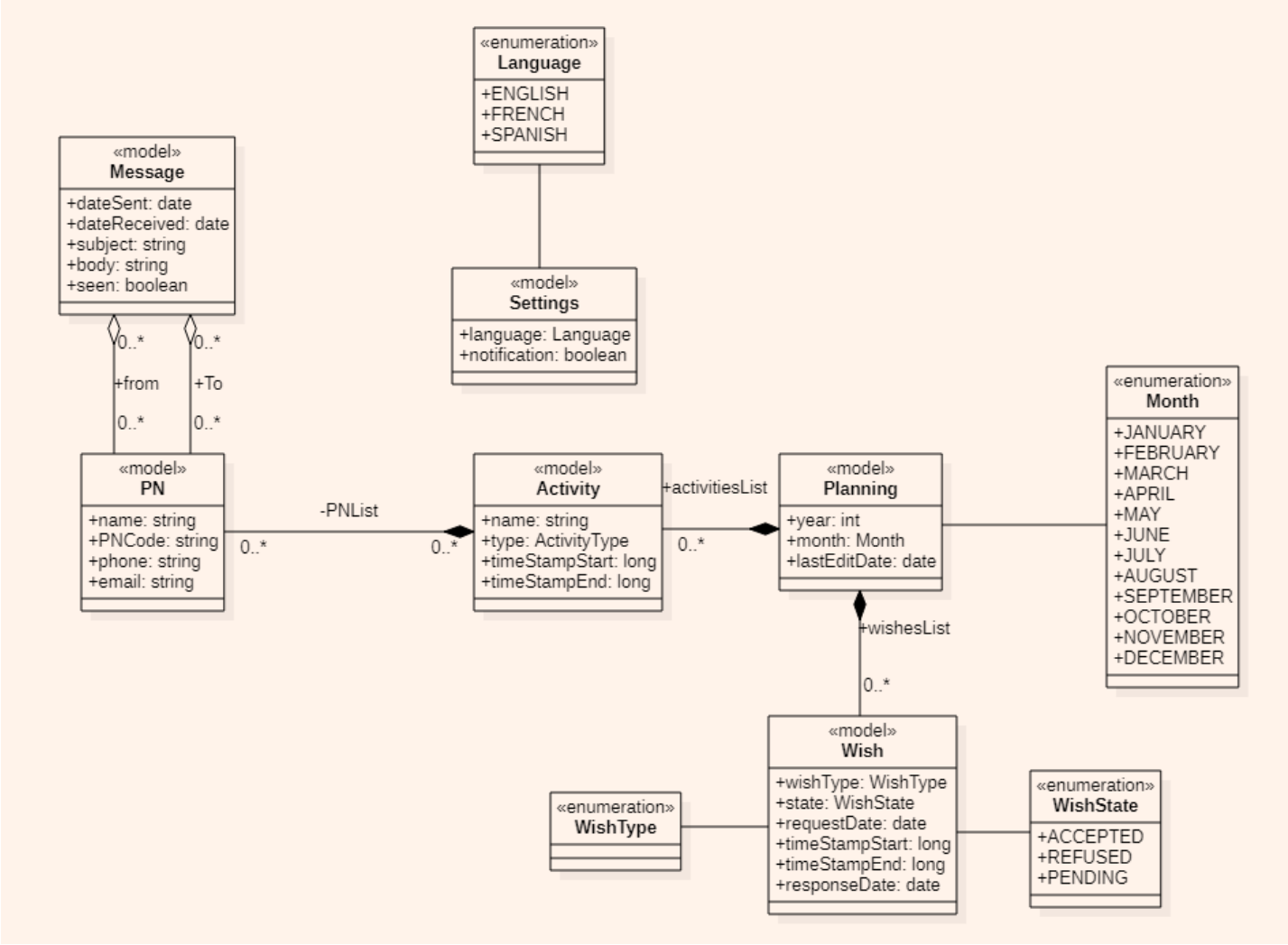
Ci-dessous le diagramme des cas d'utilisation en langage UML me permettant de bien connaitre les fonctionnalités attendues de l'application :



Pourquoi prendre du temps à faire ce diagramme ?
Être sûr de ne pas oublier de fonctionnalité
Permet à un nouvel arrivant de connaitre rapidement les fonctionnalités attendues de l'application.

La structure des données

Ci-dessous se trouve le modèle de données. Ce diagramme représente la structure des données et les différents états que peut prendre le système d'information :



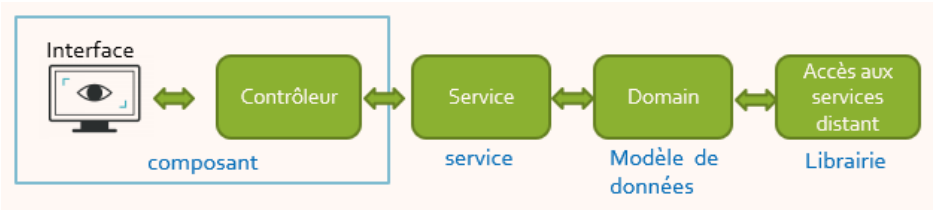
Pourquoi prendre du temps à faire ce diagramme ?

Permet de bien réfléchir à une structure de données robuste afin d'éviter que le système entre dans un état incohérent (données corrompues par exemple).

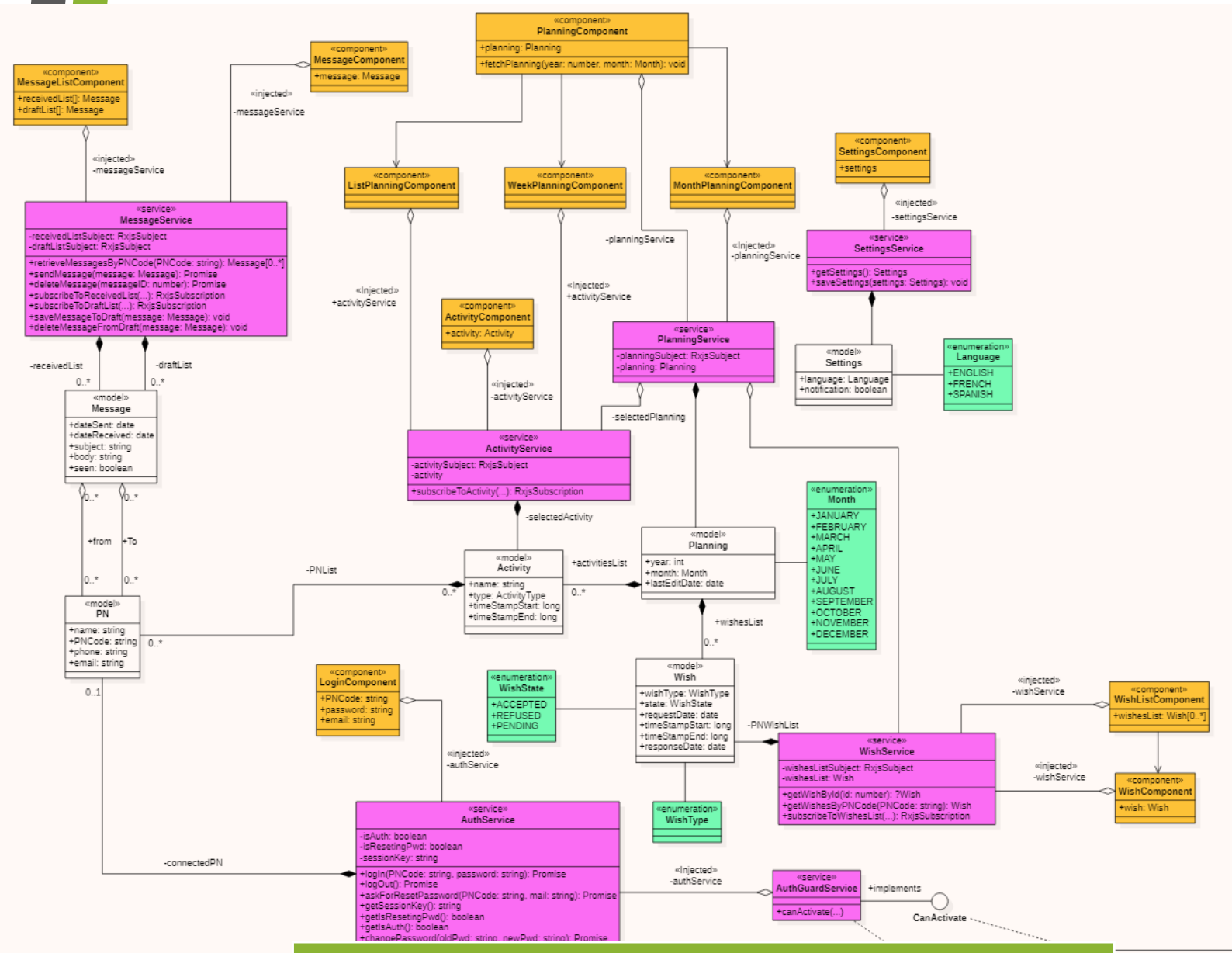
Permet à un nouvel arrivant de connaitre rapidement la manière dont sont organisées les données.

L'architecture

Le découpage de l'application en couche :



Ci-dessous, l'organisation des entités constituant l'application (structure de données + les constituants de la couche service) :



Pourquoi prendre du temps à faire ce diagramme ?

Permet d'avoir une visibilité sur le développement de l'application et d'accélérer le développement, car il suffit de retranscrire ce diagramme dans le langage de programmation choisi.

Permet à un nouvel arrivant de connaitre rapidement la manière dont est organisée l'application et permet d'unifier la manière de coder pour le projet.

Conclusion

Grâce à toute cette phase d'analyse et de conception, le développement de l'application se fera de manière plus rapide et plus clair.

De plus, le contenu de cette présentation confère une bonne avancée pour la rédaction de la documentation technique, car environ 75% sera réutilisé.