

# Dossier d'Architecture Technique

## 1 - Contexte : besoins fonctionnels et non fonctionnels

L'application doit être téléchargée via Github et fonctionner en local. Le code à télécharger est le build rendu par le build d'angular et le script de la BDD sans données.

La base de données et l'accès à celle-ci doit être protégé, car il peut y avoir des données sensibles qui transitent. Il faudra protéger les formulaires pour éviter la dégradation des données en cas de fausse manipulation.

Le nombre d'utilisateurs sera d'une seule personne.

Il n'aura pas besoin d'une grosse puissance de calcul, car les traitements se résument à des requêtes SQL sur une base de données avec peu de données et de simples additions.

## 2 - La représentation opérationnelle

Les données seront utilisées par l'application de manière personnelle. Elles seront sauvegardées dans une base de données relationnelle. C'est l'utilisateur de l'application qui a accès à toutes les données.

## 3 - La représentation fonctionnelle

Le but premier de l'application est de permettre à l'utilisateur d'organiser ses tâches et d'avoir une vue globale du temps passé à les réaliser.

L'application doit pouvoir stocker et retourner des données, ce sont les données d'une base de données qui sont manipulées directement.

## 4 - La représentation applicative

Une application permettant d'afficher les différentes tâches que l'utilisateur a créé, permet une vue globale du temps passé à réaliser les tâches selon tâche/jour. Pour cela on aura une application en angular qui communiquera avec une Base de données. L'application dépendra de la BDD.

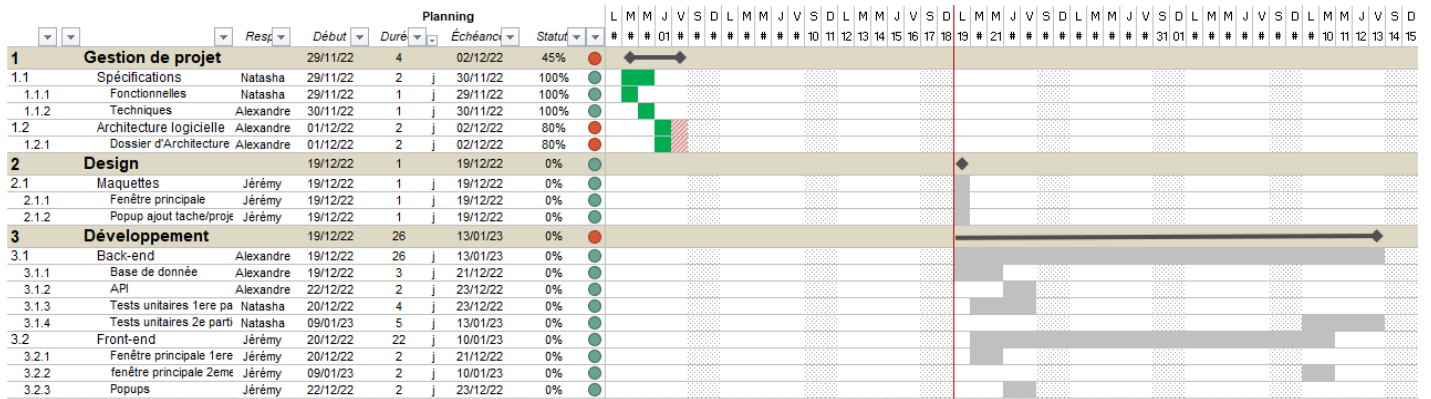
## 5 - La représentation technique

Pour ce projet nous aurons besoin d'une BDD classique et serveur de capacité minimum car l'application est désignée pour un seul utilisateur. Il n'y aura pas beaucoup de transferts de données.

## 6 - Le choix de l'architecture

Pour ce projet nous opterons pour une architecture monolithique du fait qu'il n'y a pas de scaling à préparer (une application pour un seul client) et que les services ne seront pas découplés car tout se situera sur la même page et sera relié.

## 7 - Le planning de réalisation



## 8 - Les risques

Si un service plante suite à une erreur entraînant l'impossibilité d'utiliser l'application dû à son architecture monolithique.

( Solution => Redondance du service monolithique )

Si l'application subit une injection qui toucherait la bdd

( Solution => Sécuriser les entrées de données )

Si l'utilisateur n'a pas installé toutes les technos dont dépend l'application.

( Solution => Fournir un plan de déploiement pour l'application )

Si il y a un problème avec la bdd, l'utilisateur va perdre ses informations.

( Solution => Faire des back-up réguliers de la bdd )