

# Journal de bord

---

## Mardi 03.03.2020

Le cahier des charges est terminé et validé.

La conception commence, une liste des tâches à effectuer est écrite, Il manque cependant encore des informations afin de compléter cette liste. Un planning sous la forme d'un gantt est fait et la phase de conception est organisée. Des recherches pour l'état de l'art sont faites, pour des packages ou des bibliothèques pouvant aider à concevoir le planning.

Objectifs à faire :

- Avancer dans le planning, l'état de l'art
- Commencer les maquettes

## Mardi 10.03.2020

L'état de l'art à avancer, les maquettes également.

## Mardi 17.03.2020

Pour le planning, le problème de la forme de l'enregistrement se pose toujours. Le remplissage du planning reste aussi questionnable. Le mieux sera sans doute d'avoir un template qu'on met ensuite sur le planning réel. Il faut refaire des maquettes de cette idée

## Mardi 24.03.2020

Rendez-vous par skype avec M. Soussi et M. Ouerhani. Un serveur sera mis à ma disposition et la base de données sera faite par moi. Il faut donc faire le modèle de la base de données

## Mardi 31.03.2020

Le modèle de la base de données est fait, il faut également compléter les maquettes ainsi que les issues et les spécifications détaillées.

## Mardi 07.04.2020

Les maquettes ont été complétées et les issues ont commencé à être écrites après avoir mis au clair la liste des tâches.

## Mardi 21.04.2020

La mise en place du projet django a été faite ainsi que la mise en place d'un dualboot avec ubuntu.

## Mardi 26.04.2020

Après avoir mis en place le projet, des recherches et des discussions poussent le projet à prendre une tournure mettant plutôt en place une API avec un framework frontend.

## Lundi 18.05.2020

La mise en place de l'application ainsi que des User dans le modele a été faite.

## Mardi 19.05.2020

Après discussions avec Mme Liscia, certains légers changements dans la base de données seront faits. La mise en place des modèles pour l'API django avance et le choix du framework (React) a été fait. Afin de mettre en place les users, j'ai étendu le model AbstractUser. Cependant, il faut encore que je fasse des recherches afin d'assurer de faire le meilleur choix. Il faut également que je crée un field au user permettant de noter les vacances

## Mercredi 20.05.2020

Je désire mettre en place le framework React. Il y a plusieurs options pour le faire. La première est que React soit son propre frontend et que l'application django charge une seule page Html et laisse React manager le frontend. La deuxième option est d'utiliser django REST comme une API autonome et React comme une SPA autonome également. La troisième option est de créer un min app React dans les templates Django.

L'API a été changée afin de personnaliser l'authentification.

## Vendredi 22.05.2020

Les tables n'étaient pas bien rentrées car il fallait faire python manage.py makemigrations intranetapp. Si on oublie de mettre le nom de l'application, les migrations de template et calendar ne se font pas. La mise en place de react est faite, il faut maintenant mettre en place un register temporaire et un login avec une page de base.

Le but étant de faire une SPA (Single Page Application), il faut également avoir une page main.

## Lundi 25.05.2020

La mise en place de la SPA est faite avec une page main, une page de login, une page de register, une page de resetpassword et un logout.

Ces pages sont inspirées de l'application CSRuby.

J'ai eu un soucis de commit qui contenais des données (node\_modules) qu'il ne fallait pas commit car le .gitignore n'était pas à jour. Il a fallu faire un rebase afin d'aller changer le commit en question pour ne pas commit cela afin de pouvoir push.

## Mardi 26.05.2020

Après avoir merge avec dev et réglé quelques soucis de merge, une autre branche est créée afin de faire les différents onglets.

Afin de limiter l'accès aux onglets uniquement aux utilisateurs, on peut utiliser AuthContext.Consumer afin de tester si l'utilisateur est connecté directement dans le html afin d'afficher les bons onglets. Cependant, en faisant cela, il sera également nécessaire de créer sur chaque page une condition pour tester l'authentification de l'utilisateur.

## Mercredi 27.05.2020

Le usernames sont prit par l'api et affichés sur la page userHandler. Il a fallu changer quelques points dans l'api afin de get tous les users. Il faut aussi réfléchir si on met un state loaded. Au niveau de la sécurité, il faut encore se renseigner pour voir si c'est la bonne façon de faire. Pour le get de l'api premièrement et également pour le test sur chaque page ou on test avec `this.context.getIsAuthenticated()`. On catch aussi une erreur si le get ne s'est pas fait mais il faut vérifier si c'est nécessaire.

## Jeudi 28.05.2020

---

On commence à essayer d'afficher le planning sur la page planning. Il faut cependant pouvoir créer des plannings auparavant. Afin de parser les tables json, l'utilisation de react-json-table sera une aide précieuse. On se concentrera d'abord sur les templates.

## Vendredi 29.05.2020

Des changements sont faits dans l'API afin d'avoir des get, post, update etc. L'utilisation de django-rest-framework est intéressante pour cela

## Mardi 02.06.2020

Malade

## Mercredi 03.06.2020

On continue de modifier l'API afin d'avoir le résultat voulu. On se base sur Django REST framework. L'utilisation de ModelViewSet est fait et permet de faire un Crud. On peut installer postman afin de tester l'API. On va utiliser react-table pour afficher nos plannings.

## Jeudi 04.06.2020

L'api fonctionne pour les templates et les plannings (calendar).

## Vendredi 05.06.2020

Après avoir voulu utiliser le module react-json-table, il s'avère qu'il y a des problèmes de compatibilité et il sera donc mieux d'utiliser react-table. Il faut sauvegarder les colonnes du template mais également les datas dans le tableau.

## Lundi 08.06.2020

on utilise react-table, pour cela, il faut faire des json compatible. C'est ce qui est fait. On doit encore mettre une liste pour sélectionner le template, une liste pour selectionner le template à delete, voir si AppProvider et Page sont nécessaire pour la suite.

## Mardi 09.06.2020

Il faut avoir le nom de tous les templates afin de pouvoir choisir lequel afficher. De plus, il faut vérifier que le nom ne soit pas déjà prit quand on veut créer un nouveau template. Certains problèmes d'installations ont eu

lieu avec npm, il faut mettre à jour. On installe ensuite react-select. On peut désormais choisir le template à supprimer et à afficher. On peut également créer des templates.

## Mercredi 10.06.2020

Changement de la base du template pour utiliser React-table pour que les cellules soient éditables. Il faut changer la logique du json et mettre tout dans un json.

## Jeudi 11.06.2020

Après avoir chercher, on arrive à parse le json dans la table.

## Vendredi 12.06.2020

Début de logique de sauvegarde et modification du json.

## Lundi 15.06.2020

On peut maintenant modifier le json ce qui modifie l'état. Il y a un problème avec le patch encore actuellement afin de sauvegarder les données.

## Mardi 16.06.2020

Malgré encore un soucis dans le patch avec l'api, on s'attaque au planning. On met en place le système de date.

## Mercredi 17.06.2020

On met en place le planning, le parse avec le template doit être fait. le soucis du patch est réglé.

## Jeudi 18.06.2020

Le parsing du template dans le planning fonctionne, il y a cependant un problème dans les nested columns pour mettre les datas.

## Vendredi 19.06.2020

La sauvegarde des datas n'est pas encore réglé, on fait d'abord cela dans la page template.

## Lundi 22.06.2020

La sauvegarde dans la page template fonctionne. Il y a un problème avec le get qui ne met pas les datas à jour.

## Mardi 23.06.2020

Maintenant que le planning fonctionne, il faut gérer la création, l'édition et la suppression d'utilisateur. Pour la création c'est fait en s'inspirant du sign up déjà présent. Reste à faire l'édition, la création (ajouter ville)

## Mercredi 24.06.2020

La gestion des utilisateurs fonctionne avec le create et le patch et le delete.

## Jeudi 25.06.2020

On fait maintenant la gestion si l'utilisateur est un manager et les différents accès. Il y a également les noms des plannings qui sont bons.

## Vendredi 26.06.2020

On se renseigne sur l'écriture du rapport. On test également certains style css afin de rendre le planning lisible.

## Lundi 29.06.2020

On a changé quelques détails en css. Il reste à régler la taille des plannings et également l'édition ou non selon les droits des utilisateurs. Il rest à régler le reset de mot de passe (automatique pour chaque nouvel utilisateur).

## Mardi 30.06.2020

L'architecture du rapport a été faite.

## Mercredi 01.07.2020

On vérifie que le mois que l'on veut créer ne le soit pas déjà. On fait également du css afin d'afficher sur une page le planning

## Jeudi 02.07.2020

On cherche à faire fonctionner le get des horaires

## Vendredi 03.07.2020

Après avoir essayer de faire le get dans le framework, il s'avère qu'il est possible de le faire dans l'api avec cron. (crontab)

## Lundi 06.07.2020

Configuration de la requete à l'api pour les horaires. Au final il faut prendre depuis l'api django et sauvegarder dans la base de donnée. On va utiliser des cronjob On a fait le reset de mot de passe.

On configure également l'oubli et le reset de mail.

## Mardi 07.07.2020

Maintenant qu'on peut get et sauvegarder dans la base de données les horaires, il faut les afficher dans react.

## Mercredi 08.07.2020

Mise en place de la sécurité dans l'api

## Jeudi 09.07.2020

Mise en place des tests

## Vendredi 10.07.2020

Résolution de problème dans l'horaire et de l'affichage de l'horaire

## Lundi 13.07.2020

Mise en place des tests katalon et decision de déploiement vendredi 17.07.2020

## Mardi 14.07.2020

Création de la page des notes d'informations et ajout des notes d'infos.

## Mercredi 15.07.2020

Ajout de l'exportation en pdf des plannings

## Jeudi 16.07.2020

Préparation au déploiement avec les requierements.

## Vendredi 17.07.2020

Déploiement. Correction de petites erreurs.

## Lundi 20.07.2020

Rapport

## Mardi 21.07.2020

Rapport et tests du site. Constatation de certaines erreurs et corrections

## Mercredi 22.07.2020

Correction des crontabs qui ne fonctionnent pas.

## Jeudi 23.07.2020

Finalisation du rapport

## Vendredi 24.07.2020

Rendu