

ABID Julien  
WALTER Damien  
WAWRZYNIAK Maxime

# Rapport – Projet A52

IUT Robert Schuman

---

2023- 2024

## Table des matières

Table des matières .....	2
Analyse et Spécification des besoins.....	3
1. Contexte du Projet : .....	3
2. Fonctionnalités Requises :.....	3
3. Contraintes techniques : .....	3
4. Délai : .....	3
5. Livrables : .....	3
7. Remarques : .....	3
Gestion de projet .....	4
Workflow :.....	4
Revue de code : .....	4

# Analyse et Spécification des besoins

En prenant comme référence le document *specifications.md* disponible sur le dépôt *Gitlab* du projet :

## 1. Contexte du Projet :

Le projet consiste à développer une application web de tableau KanBan répondant à des spécifications détaillées. L'objectif est de créer une interface fonctionnelle et visuellement soignée permettant la gestion des tâches dans trois colonnes distinctes : TODO, WIP, et DONE.

## 2. Fonctionnalités Requises :

- Interface utilisateur avec trois colonnes : TODO, WIP, et DONE.
- Possibilité d'ajouter des tâches avec les informations suivantes :
- Date de début et de fin
  - Description
  - Titre
  - Affectation
  - Tags
  - Code couleur
- Utilisation exclusive de HTML, CSS et Vanilla Javascript.
- Hébergement de l'application web chez l'équipe de développement.
- Assurer une disponibilité constante de la dernière version stable.

## 3. Contraintes techniques :

- Utilisation exclusive de HTML, CSS et Vanilla Javascript.
- Respect de l'approche minimaliste pour une application sobre et fonctionnelle.

## 4. Délai :

Le projet doit être livré dans un délai approximatif de trois jours à partir du 14/12/2023, ainsi qu'une date finale fixée au 17/12/2023 12 h.

## 5. Livrables :

- Application web fonctionnelle conforme aux spécifications.
- Code source du projet.
- Documentation succincte pour l'installation et la maintenance.

## 7. Remarques :

- La simplicité et la fluidité de l'expérience utilisateur sont des priorités.
- La version initiale peut ne pas inclure toutes les fonctionnalités demandées, mais doit être stable.
- Le design du site doit respecter la charte graphique fournie par le client

## Gestion de projet

### Workflow :

Pour ce projet, l'organisation du dépôt s'articule autour de *GitFlow*, qui utilise des branches spécifiques pour organiser le développement collaboratif. Les branches principales sont *Main* (versions stables) et *Develop* (en cours de développement). Les fonctionnalités sont développées dans des branches *Feature*, les versions dans des branches *Release*, et les corrections de bugs dans des branches *Hotfix*.

La convention des commits utilisé est la suivante :

<type>[(scope)] : <description>

Pour le nommage des commits, nous utilisons *commitlint* qui est un outil utilisé pour imposer et vérifier le format des messages de commit. Il garantit que les messages de commit suivent un format cohérent et normalisé dans l'ensemble de notre projet. Les règles qu'il vérifie sont :

- La présence d'un type (feat, fix, build, etc.)
- La présence d'un scope
- Si le sujet est vide
- Le sujet doit être en minuscule
- La longueur max du commit. Dans notre cas, on a mis 120 caractères maximum.

Pour avoir plus de détail sur l'outil : <https://github.com/conventional-changelog/commitlint>

Pour l'analyse du code javascript lors d'un commit, nous utilisons l'outil *Eslint* qui permet de faire de l'analyse statique de code. Il permet de définir et d'appliquer des règles spécifiques au style de code, à la qualité et la détection d'erreurs. Les règles que nous avons mis en place sont :

- Détection des variables non utilisées
- Pas de multiple espace
- Mettre les noms des variables en camelCase

Pour plus d'information sur les autres règles : <https://eslint.org/docs/latest/rules/>

### Revue de code :

La révision du code s'est effectuée par-dessus l'épaule ou en binôme, ainsi que durant les pulls request.

### 3. Organisation du travail et affectation des tâches :

Tâches	Qui s'en occupe ?	Branche
CSS Page	Maxime	feature/css-form-page
CSS Formulaire	Julien	feature/css-form
Organisation HTML	Damien	feature/base-html
CSS Post-it	Julien	feature/css-form-page
CSS & Form informations de la tâche	Damien	feature/form-info
<i>Javascript</i>		
Ajout de tâches	Maxime	feature/add-task
Visualisation des tâches	Maxime	feature/Visualization-of-task
Déplacement des tâches	Damien	feature/drag-and-drop
Suppression des tâches	Damien	feature/delete-task
Stockage des données	Maxime	feature/store-data
<i>Revue de code</i>		
Vérification des commits (commitlint)	Maxime	feature/commitlint
Vérification de la mise en forme du code javascript (eslint)	Maxime	feature/eslint
<i>Autre</i>		
Documentation	Maxime	/
Ecriture des tests	Maxime	ci/unit-test
Rapport	Julien	/
Readme	Maxime/Julien	/
Emails au client	Julien	/
Cahier des charges	Julien	doc/cahier-des-charges
License	/	/
CI/CD	Julien	main