UNIVERSITÉ Grenoble Alpes

Rapport de Projet

IRLEI

UNIVERSITÉ Grenoble Alpes

Rapport de projet

IRLEI

Les informations d'identification du document

Référence du document :	D8
Version du document :	1.01
Date du document :	17 juin 2024
Auteur(s) :	Alexandre Pernier Jules Sage

Les éléments de vérification du document

Validé par :	
Validé le :	17/06/2024
Soumis le :	17/06/2024
Type de diffusion :	Document électronique (.odt)

Sommaire

1. Introduction

Ce document regroupe les points qui n'ont pas été abordés dans les autres livrables, points qui sont nommés dans la table des matières. Ce document sert globalement à dresser le bilan du projet, pour établir ces forces et faiblesses et ce sur quoi il faut progresser dans le futur ou continuer à perfectionner.

2. Rappel du contexte

Le projet IRLEI est une commande d'une équipe du laboratoire d'Informatique de Grenoble devant répondre à un besoin qui se manifeste sous forme d'une application proposant un affichage de données sous forme de diagrammes statistiques en fonction de source de données introduite de manière manuelle dans un système.

Il a été convenu d'un cahier des charges (trouvable en référence) où sont spécifiées différentes fonctionnalités des plus essentielles aux plus facultatives.

3. Bilan

3.1. Architecture de l'application

Le projet est structuré autour d'une architecture client-serveur, combinant des technologies modernes pour offrir une expérience utilisateur fluide et une gestion efficace des données. Le frontend est développé avec Vue.js, un framework JavaScript progressif permettant de créer des interfaces utilisateur dynamiques. Pour gérer l'état de l'application de manière centralisée et prévisible, Vuex est utilisé, facilitant le partage et la manipulation des données entre les différents composants. Axios est intégré pour effectuer des requêtes HTTP, assurant une communication transparente avec le backend.

Le backend est construit avec Django, un framework web Python, garantissant une structure solide et sécurisée pour le développement d'applications web. Django REST Framework est employé pour créer des APIs RESTful, permettant une interaction efficace entre le frontend et le backend. L'authentification et la gestion des utilisateurs sont simplifiées grâce à Djoser, qui offre des endpoints prêts à l'emploi pour les opérations courantes comme l'inscription et la connexion. Enfin, SQLite est choisi comme base de données, offrant une solution légère et intégrée pour le stockage des données, idéale pour le développement et les tests.

3.2. État des lieux

En reprenant les fonctions décrites dans le cahier des charges, nous retrouvons dans le projet à ce jour, bien les fonctionnalités minimum requises :

- Une analyse descriptive des systèmes sur toutes les mesures disponibles
- Le choix des systèmes à analyser
- Un prétraitement facilitant l'analyse : tri et comparaison à la moyenne

Pour ce qui est des fonctionnalités dites supplémentaires, il est possible de :

- Regrouper plusieurs systèmes sous un même label
- Charger les données brutes de manières interactive
- Consulter les statistiques d'un seul système et de voir où il se situe par rapport à la moyenne

Les fonctionnalités non implémentées sont toutes facultatives comme :

- La prise en charge des utilisateurs non-administrateurs
- La classification claire des requêtes, des époques et/ou des systèmes

3.3. Analyse des écarts

Au niveau des fonctionnalités, le projet est fonctionnel en tant que tel : toutes les fonctionnalités minimum requises sont satisfaites. Cependant, si nous comparons au prototype de base, il y a quelques fonctionnalités qui n'ont pas été implémentées comme la standardisation des données.

Tableau comparatifs des autres fonctionnalités			
Nom de la fonctionnalité	Non implémentée dans le nouveau système	Fonctionnalité identique/similaire	Amélioration par rapport au prototype de base
Affichage des données (Round/Queries)		X	
Affichage par rapport à la moyenne			X
Affichage par rapport à un système	X		
Affichage des mesures standardisées	X		
Option de tri			X
Ajout de systèmes			X
Création de compte			X
Analyse des systèmes un à un			X

À l'heure de la rédaction du document, le nouveau système n'a pas encore été livré sur un serveur dédié. Le produit fini est téléchargeable depuis Github et utilisable sur un ordinateur personnel.

3.4. Gestion des contraintes

La contrainte qui a posé le plus problème était l'apprentissage des nouveaux outils qui était nombreux et complexe et devait être utilisé dès le début de la phase de développement du projet : il était compliqué d'apprendre plusieurs outils en même temps et de s'habituer aux outils alors que notre temps consacré au projet était au départ limité. Le problème s'est volatilisé petit à petit de lui-même lors du passage à plein temps sur le projet puisque l'apprentissage était bien plus optimisé.

3.5. Méthode

Un trello a été mis en place, mais peu utilisé, car les objectifs hebdomadaires étaient déjà inscrits dans les conversations, rapport et feedbacks des réunions avec les commanditaires ou dans l'équipe. L'utilisation d'un outil de gestion de projet supplémentaire n'était donc pas nécessaire, car les informations qu'on pouvait y trouver n'étaient ni très variées, ni très complexes.

3.6. Ce qui est à retenir en positif

Les motifs de rendez-vous ont été fixés à chaque réunion, les objectifs hebdomadaires aussi. Un test du système à mi-parcours a été mis en place en cas de doute, ce qui a permis de converger vers la vision des commanditaires. Aucun changement/problème majeur en raison d'une incompréhension de la demande n'a été constatés, preuve d'une phase de conception réussie.

3.7. Axe d'amélioration

Nous pourrons noter quelques points de progression au niveau de la structure du code dans le frontend qui manque parfois de propreté et de clarté (manque de commentaire, nom de fonction pas toujours très clair), notamment à cause du fait que nous n'avons pas eu le temps de faire une revue de code à la fin.

4. Conclusion

Ce projet a été une bonne opportunité pour appliquer les méthodologies vu en cours théorique et nous donne une indication de notre niveau d'organisation en plus de notre niveau technique.

5. Glossaire

Trello : Logiciel de gestion de projet

Round : Correspond à un temps T, où on évalue les systèmes de recherche pour les comparer dans le temps.

Queries : En français = requêtes. Variables dépendantes des tests des systèmes. Correspond à un ensemble de mots permettant de faire correspondre des documents à un système.

6. Références

Cahier des charges :
https://github.com/Kitsggwp/ProjetTER_IRLEI/blob/main/Documentation/Cahier_des_charges.pdf