

DUMORA-DANEZAN Jacques
TRAVAUX Louis
MARTINEZ Hugo
BRIOLLET Florian
SERRES Valentin

IA Pau
Projet ING1 GI

PROJET ING-1 GI

IA PAU



Sommaire

Introduction :	2
I - Conception	3
I.I - Organisation de la base de données.....	3
I.III - Technologie et choix de l'organisation.....	4
II - Fonctionnalités	6
II.I - Statut d'administrateur.....	6
II.II - Statut de gestionnaire.....	6
II.III - Statut d'étudiant.....	7
II.IV - Messagerie.....	7
II.V - Analyse de code.....	7
III - Répartition des tâches au sein du groupe	8
Conclusion :	8

Introduction :

Durant ces trois dernières semaines, nous avons travaillé sur un projet donné par l'association IA Pau. Nous devons réaliser un site permettant de créer, gérer et participer à des data Challenges et des data Battle, accessibles par trois types d'utilisateurs : *administrateur*, *gestionnaire* et *étudiant*.

Ce rapport va permettre de vous résumer notre travail. Pour cela nous allons dans un premier temps développer la façon dont nous avons conçu notre site. Ensuite nous présenterons ses différentes fonctionnalités. Enfin, nous conclurons sur les possibilités d'amélioration, implémentables par la suite si notre projet était retenu par l'association.

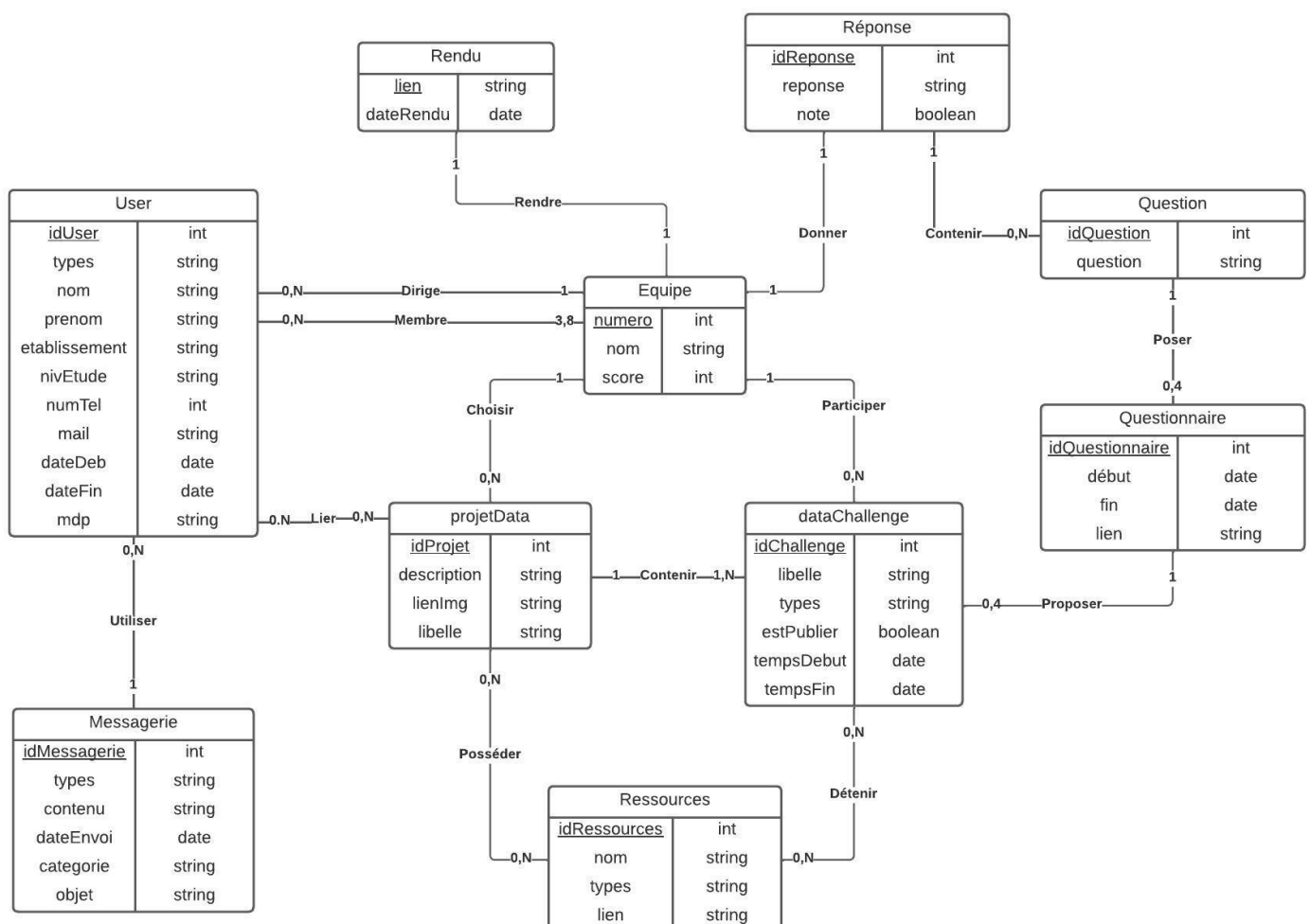
I - Conception

Afin de concevoir notre site, nous avons dû réfléchir à différents points :

- L'organisation de la base de données, structure sur laquelle repose notre site,
- Le choix d'une charte graphique et l'organisation du visuel global du site,
- Les technologies que nous allons utiliser,
- L'organisation de notre code.

I.1 - Organisation de la base de données

Nous avons commencé par concevoir un MCD permettant de représenter l'organisation de la base de données. Nous avons donc obtenu le MCD suivant :



Ce dernier a été réalisé après avoir analysé et interprété le sujet. Il permet de comprendre l'architecture de notre base de données. Par la suite nous avons pu réaliser le MLD correspondant et ainsi créer les différentes Tables de notre base de données.

Dans le MCD ci-dessus, nous avons décidé de laisser une table pour l'utilisateur qui changera de "types" en fonction du type d'utilisateur, c'est-à-dire s'il s'agit d'un *étudiant*, *gestionnaire* ou *admin*. Ensuite nous avons confondu les Data Challenges et les Data Battle parce que ces derniers possèdent les mêmes attributs, les seules différences sont leur durée, et s'ils possèdent ou non des questionnaires. Pour finir sur la description de notre MCD, nous avons décidé que pour s'inscrire à un data challenge, il faudrait d'abord faire partie d'une équipe.

I.II - Chartes Graphique

La charte graphique que nous avons adoptée est en grande partie inspirée de celle du site web de l'association IA Pau. Nous avons sélectionné ses points forts, mais aussi ajusté ce qui nous est apparu ajustable. Par exemple, nous avons particulièrement apprécié la couleur dominante du logo, et avons estimé qu'elle n'était que trop peu utilisée. Ainsi, nous en avons fait notre dominante.

Aussi, nous avons choisi de réutiliser la page d'accueil du site iapau.org pour éviter de désorienter nos éventuels visiteurs.

Par ailleurs, les éléments fréquemment utilisés sont normalisés, afin d'obtenir une cohérence globale, et ainsi un confort pour l'utilisateur : cela concerne les boutons par exemple, ou encore les champs d'entrée utilisateur.

I.III - Technologies et choix de l'organisation

Notre site repose sur une architecture *Modèle - Vue - Contrôleur* codée en PHP. Cette architecture permet une séparation claire des responsabilités entre les différentes parties du code. Ce dernier est donc plus simplement modulable, maintenable et réutilisable.

Le modèle est responsable de l'interaction avec la base de données (back-end), la vue de l'affichage de ces données (front-end), et le contrôleur de faire le lien entre les deux. Afin de faciliter l'implémentation de cette architecture nous avons développé une petite bibliothèque nous permettant de nous simplifier la tâche.

Nous avons choisi PHP dans le but de grouper le développement du back-end et du front-end et ainsi arriver rapidement à une solution opérationnelle, agréable d'utilisation et facilement maintenable.

II - Fonctionnalités

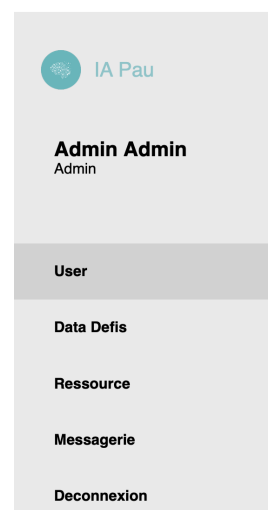
II.I - Statut d'administrateur

L'administrateur dispose de quatre fonctionnalités principales. La première, *User*, permet de créer, modifier ou supprimer des utilisateurs. Il peut créer des *Étudiants*, des *Gestionnaires*, mais aussi des *Administrateurs*.

Via l'onglet *Data Défis*, il peut créer et configurer des *Data Challenge* et des *Data Battles*. C'est au travers de cet onglet qu'il crée les *Projets* rattachés aux *Data Challenges*. Il peut directement y attacher une ressource, ou le faire par la suite.

L'onglet *Ressource* permet, comme son nom l'indique, de créer et d'associer des ressources aux *Projets* préalablement créés.

Enfin, tout comme le gestionnaire, l'administrateur peut publier des messages dans la *Messagerie*, dans la catégorie *GÉNÉRAL*, ou dans la section correspondant à un *Data Challenge*.



II.II - Statut de gestionnaire

Le gestionnaire est en mesure, tout comme les étudiants, de mettre à jour ses informations personnelles mais aussi d'afficher les *Data Challenges* actuellement disponibles, et surtout, de configurer les *Data Battles* : créer ou modifier les questionnaires, et corriger les réponses des *Étudiants*.

L'onglet *Rendu* lui permet consulter le travail effectué par les *Étudiants* : un lien vers github/gitlab/... où il peut récupérer aisément le code.

Pour finir, le gestionnaire a accès à la fonctionnalité d'analyse de code, lui permettant d'obtenir des statistiques sur le code fourni par les étudiants.

Comme les autres utilisateurs, il dispose d'un accès à la messagerie.

II.III - Statut d'étudiant

De même que pour les *Gestionnaires*, les *Étudiants* ont la possibilité de consulter leurs informations personnelles ainsi que de les modifier. Ils peuvent modifier, par exemple, leur adresse, leur numéro de téléphone ou encore leur mot de passe. Ils ont également la possibilité de contacter l'administrateur par mail.

Les étudiants ont la possibilité de consulter les projets auxquels ils participent, les data challenges ainsi que les data battles publiés, les équipes dont ils font partie ainsi que les messages envoyés par les gestionnaires et les administrateurs sur les différents canaux de communications qui leurs correspondent.

Ils ont également à leur disposition un analyseur de code, qui permet, comme son nom l'indique, d'analyser un code python et ainsi de leur donner des informations sur ce code.

Enfin les étudiants ont la possibilité de créer une équipe et de rajouter des membres afin de s'inscrire aux différents projets proposés sur le site.

II.IV - Messagerie

La messagerie permet aux administrateurs et gestionnaires de publier des messages par catégorie. Ces catégories correspondent aux noms des *Data Challenges* disponibles, plus une catégorie *générale*, pour les annonces globales.

Les *étudiants*, quant à eux, ne disposent pas de la possibilité de publier dans cette messagerie. Ils peuvent simplement consulter les messages des *Data Challenge* auxquels ils participent.

II.V - Analyse de code

Notre analyseur de code permet d'obtenir des informations de base sur un code python fourni sur le site : le nombre de lignes et de fonctions, ainsi que la longueur minimale, maximale et moyenne des fonctions.

Il s'appuie sur une API simplifiée suivant les principes de l'architecture REST, permettant un traitement de requêtes HTTP GET ou POST pour effectuer les actions d'analyse.

III - Répartition des tâches au sein du groupe

Chacun d'entre nous a, lorsqu'il a développé une fonctionnalité du front-end (une *Vue*), développé aussi la partie *Contrôleur*, assurant ainsi le lien entre l'affichage client et le serveur.

Jacques Dumora-Danezan : Les pages d'affichage et de management des *Data Battles* et *Challenges*, la création d'équipes, Gestion des ressources , des utilisateurs et des projets.

Louis Travaux : La page d'accueil, la page d'inscription et celle de connexion, la structure du dashboard, la messagerie.

Hugo Martinez : L'analyseur de code et API REST en Java.

Valentin Serres : La conception de la base de données, son script d'installation, les requêtes d'interaction avec la base de données, la section *Mon compte*, permettant la modification des données d'un utilisateur, mais aussi le développement de la page permettant la création et la notation des questionnaires des *Data Battles*.

Conclusion :

Pour conclure nous pouvons dire que notre projet respecte la plupart des objectifs demandés dans le sujet. Cependant il nous reste encore quelques points à modifier, comme le fait de gérer les liens des questionnaires en les générant de façon automatique à leur création, et également de gérer le classement des différentes équipes. L'analyseur de code peut aussi être amélioré avec de nouvelles fonctionnalités plus poussées, comme un calcul approximatif de complexité des fonctions. Nous voulions également récupérer les fichiers de code python automatiquement depuis un répertoire github pour les analyser, cependant, par manque de temps nous nous sommes contentés d'une analyse manuelle par fichier.

Finalement ce projet a permis de renforcer nos connaissances et de mettre en pratique les différents langages et concepts vus cette année et donc d'améliorer nos compétences.