Application pour conception de maquettes de formation

Encadrants : LESAINT David, BEHUET Corentin, SIMON Aurélien

Contexte :
Ce sujet s'inscrit dans le cadre d'un projet mené au LERIA sur l'élaboration d'emplois du temps universitaires (EDT). Quelques briques logicielles dont un solveur de contraintes ont été développées et intégrées pour répondre à cette problématique. Ce système d'EDT repose sur un SGBD et est exposé via des API REST.

Objectifs :
Les maquettes de formation se décomposent en unités d'enseignement, obligatoires ou non, selon les parcours et options proposés pour chaque formation. Chaque unité se décline en une ou plusieurs modalités pédagogiques (cours magistral, TP, TD, projet tutoré, conférence, etc.) et d'évaluation (CC, CT, soutenance, etc.), chacune ayant sa période de temps et son volume horaire.
L'objectif est ici de créer une application web permettant de créer, amender et visualiser des maquettes de formation sous différents formats (tabulaire, arborescent ou autre). Cette application doit permettre :

- d'ajouter, modifier et supprimer des éléments de maquette
- d'identifier les éléments communs à plusieurs parcours ou formations
- d'aider chaque responsable d'unité à découper le volume horaire attribué à une modalité en un ou plusieurs blocs de séances (p. ex. 10h de CM découpé en 1 bloc de 4 séances de 2h et 1 bloc de 2 séances de 1h).
- d'inscrire des étudiants aux formations en les guidant interactivement dans leur choix de parcours et d'options selon les règles codifiées.

Technologies :
Des API du système d'EDT seront mises à disposition pour que votre application puisse extraire et communiquer les données de maquettes et d'inscription. Pour faciliter déploiement et intégration, votre application devra être conteneurisée avec Docker. Le choix des technologies pour le front-end est libre (React, Vue.js, etc). La communication avec les API devra se faire via un back-end développé en Python (Django, Flask, FastAPI, etc.).

Création graphique de règles métiers pour emplois du temps

Encadrants : LESAINT David, BEHUET Corentin, SIMON Aurélien

Contexte :
Ce sujet s'inscrit dans le cadre d'un projet mené au LERIA sur l'élaboration d'emplois du temps universitaires (EDT). Quelques briques logicielles dont un solveur de contraintes ont été développées et intégrées pour répondre à cette problématique. Ce système d'EDT repose sur un SGBD et est exposé via des API REST.

Objectifs :
La création et l'édition des règles à satisfaire par un EDT constituent un point crucial à l'utilisation du système. En l'état, ces règles se programment dans un langage formel. Il est donc essentiel de développer des interfaces permettant à quiconque de les exprimer facilement sans recourir à un éditeur de code.
L'objectif est ici de concevoir une interface qui offre, dans un premier temps, une visualisation de l'agencement des séances d'un ou plusieurs cours sous forme de graphe de type flow. Dans un second temps, l'interface devra permettre d'exprimer graphiquement des règles d'orchestration (séquencement de séances, périodes autorisées/interdites, volume de séances par jour ou par semaine). Par exemple, le séquencement des séances pourra être défini par un graphe dont les nœuds représentent les séances et les arêtes l'ordre imposé.

Technologies :
Des API du système d'EDT seront mises à disposition pour que votre application puisse extraire et communiquer règles et séances de cours. Pour faciliter déploiement et intégration, votre application devra être conteneurisée avec Docker. Le choix des technologies pour le front-end est libre (React, Vue.js, etc). La communication avec les API devra se faire via un back-end développé en Python (Django, Flask, FastAPI, etc.).

Bibliothèque d'édition collaborative d'emploi du temps

Encadrants : LESAINT David, BEHUET Corentin, SIMON Aurélien

Contexte :
Ce sujet s'inscrit dans le cadre d'un projet mené au LERIA sur l'élaboration d'emplois du temps universitaires (EDT). Quelques briques logicielles dont un solveur de contraintes ont été développées et intégrées pour répondre à cette problématique. Ce système d'EDT repose sur un SGBD et est exposé via des API REST.

Objectifs :
La visualisation des emplois du temps constitue un enjeu clé du projet. De nombreuses bibliothèques JavaScript existent, mais elles présentent trois inconvénients majeurs : navigation lourde entre différents niveaux de représentation (jour, semaine, mois, année), interactivité limitée pour modifier des EDT, absence de support pour l'édition collaborative et simultanée d'EDT (co-construction en temps-réel).

L'objectif est ici de créer une bibliothèque de visualisation d'EDT qui offre des fonctionnalités de construction et modification interactive (p. ex. insertion de séance ou reprogrammation de séance par glisser-déposer). Ce composant graphique devra faciliter la navigation entre les différents niveaux de représentation (jour, semaine, etc.) lorsque l'utilisateur est en mode construction/modification. Il devra aussi, dans le cadre d'une application web, nativement gérer les mises à jour lorsque plusieurs utilisateurs collaborent en temps-réel à la conception d'un même EDT.

Technologies :
La bibliothèque doit être développée en JS et CSS sans utilisation de librairies/frameworks tierces autres que des outils de pré-processing (utilitaires CSS, compilateurs Typescript, etc.). Elle sera intégrée à une application web minimaliste - à développer - permettant la collaboration temps-réel via web sockets. Des API du système d'EDT seront mises à disposition pour que votre application puisse extraire et communiquer les données d'EDT. Pour faciliter déploiement et intégration, l'application devra être conteneurisée avec Docker.

Conception et configuration de modèles de caractéristiques

Encadrants : LESAINT David, BEHUET Corentin, SIMON Aurélien

Contexte :
Un modèle de caractéristiques (FM - feature model en anglais) est une représentation de type arborescent permettant de décrire toutes les variantes d'un même système (logiciel, produit, service, etc.). Les nœuds incarnent les différentes caractéristiques du modèle et peuvent être soumis à différentes contraintes de parenté et de dépendances. Un utilisateur peut alors configurer son propre système à partir du FM par choix de caractéristiques en respectant les contraintes imposées par le FM.

Objectifs :
L'objectif du projet est de développer une application web permettant la création, la visualisation et la configuration d'un modèle de fonctionnalités dédié au domaine des emplois du temps. FM et configurations seront sérialisés au format XML selon deux schémas. En phase de configuration, l'application devra valider les choix de l'utilisateur et le guider interactivement en fonction des contraintes du FM.

Technologies :
Deux schémas XML - un pour le modèle de caractéristiques, l'autre pour les configurations - vous seront fournis qu'il conviendra d'améliorer. Le choix des technologies front/back-end est libre. L'application devra être conteneurisée via Docker.