

RAPPORT TECHNIQUE  
PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
DANS LE CADRE DU COURS LOG795 PROJET DE FIN D'ÉTUDES EN GÉNIE LOGICIEL

**Back-End du Calculateur LEED pour bâtiment durable**  
**UBUBI**

Version 1.3 – 15/08/2017

**Auteurs**

Gergel, Sébastien - GERS32109105

Messena Jean, Yomé - JEAY03098804

Picard, Etienne - PICE19099002

### Suivi des changements

\*A – Ajouté    M – Modifié    S – Supprimé

Numéro de Version	Date	Numéro de figure, table ou section	*A M S	Brève description du changement	Numéro de demande changement
1.0	2017/07/01	Tout	*A	Première version du document	-
1.1	2017/08/15	Tout	M	Revue du document par Prof. A.April	
1.2	2017/08/15	Tout	M	Revue du document par Mathieu Dupuis	
1.3	2017/08/15	Tout	M	Intégration des changements	

## **REMERCIEMENTS**

Nous tenons à remercier monsieur Mathieu Dupuis pour son temps et ses conseils tout au long de ce projet. Sans sa contribution, il aurait été infiniment plus complexe de comprendre le domaine d'affaires ainsi que de communiquer avec le client.

Nous remercions aussi le professeur Alain April pour ses conseils et pour l'aide qu'il nous a apporté au cours de ce projet.

**BACK-END DU CALCULATEUR LEED POUR BÂTIMENT DURABLE UBUBI****Sébastien Gergel, YoméMessena Jean, Etienne Picard****RÉSUMÉ**

Ce rapport de projet présente la synthèse des travaux du projet de fin d'études réalisée dans le cadre du cours LOG795 à l'ÉTS durant la session d'été 2017. Ce projet logiciel vise la conception d'un calculateur de cycle de vie pour bâtiments utilisant les données numériques provenant de logiciel de modélisation 3D. Durant la session, nous avons conçues, documentées et implémentées plusieurs fonctionnalités, tel qu'un module de gestion des utilisateurs, un module de création de «workspaces», l'importation des données provenant d'un modèle de bâtiment 3D, et la gestion de plusieurs règles d'affaires. Ce projet est développé en collaboration avec l'école de Polytechnique de Montréal sous la direction de l'étudiant en doctorat Mathieu Dupuis et de son superviseur, le professeur Alain April.

## Table des matières

LISTE DES TABLEAUX.....	7
LISTE DES FIGURES.....	8
LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES.....	9
INTRODUCTION .....	10
CHAPITRE 1 Problématique et contexte .....	11
Chapitre 2 Objectifs du projet.....	12
Chapitre 3 Méthodologie.....	13
Chapitre 4 Planification du projet.....	14
Chapitre 5 Livrables .....	15
Chapitre 6 Sommaire des travaux réalisés.....	16
6.1 Conception du système d'utilisateurs et de Workspaces.....	16
6.2 Création de «workspaces».....	17
6.3 Changement de workspace .....	18
6.4 Mise à jour du modèle .....	20
6.5 Création d'un processus d'énergie .....	21
6.6 Changements du code unformat des objets et types d'objets .....	21
6.7 Gestion des unités .....	21
6.8 Ajout d'échange avec les bons types.....	22
6.9 Empêcher la suppression de certains type d'échanges.....	22
Chapitre 7 Risques.....	23
7.1 Risques prévus .....	23
7.2 Risques Encourus .....	24
Chapitre 8 Post mortem du projet.....	25
8.1 Objectifs non-atteint.....	25
8.2 Changements qui auraient pu être apportés en cours de projet .....	25
8.3 Leçons tirées .....	26
Chapitre 9 Contribution des membres .....	27
9.1 Gergel, Sébastien .....	27
9.2 Jean, YoméMessena .....	27

9.3 Picard, Etienne .....	27
9.4 Contributions à parts égales .....	28
CONCLUSION .....	28
LISTE DE RÉFÉRENCES .....	29
Annexe A Plan de Travail .....	31
Annexe B Capture d'écran de l'application.....	36

## **LISTE DES TABLEAUX**

Tableau 1: Abréviations, sigles et acronymes.....	9
Tableau 2: Description des artefacts .....	15
Tableau 3: Risques prévus .....	23
Tableau 4: Plan de travail .....	31

## LISTE DES FIGURES

Figure 1: Schéma de la section usager de la base de données.....	17
Figure 2: Page de connexion.....	36
Figure 3: Page d'enregistrement .....	36
Figure 4:Exemple de création de processus d'énergie .....	37
Figure 5: Exemple de sous processus d'énergy .....	37
Figure 6: Insertion des codes unifomats.....	37
Figure 7: Téléchargement d'un nouveau modèle vers le serveur .....	38
Figure 8: Liste des modèles après upload.....	38



## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

Tableau 1: Abréviations, sigles et acronymes

Terme	Définition
UBUBI	Uncertain But Useful Building Information, le nom du projet.
Play! 2 Framework	Le cadriciel utilisé pour l'implémentation d'UBUBI.
JPA	Java Persistence API, une interface de programmation pour les données relationnelles.
SCRUM	Méthode agile pour le développement de projets.
Autodesk Revit	Logiciel de modélisation de bâtiment 3D.
Hibernate	ORM Java pour la gestion de la persistance d'objets.
OpenLCA	Logiciel d'analyse de cycles de vie, UBUBI utilise son propre modèle de base de données.
ORM	Object Relational Mapping, permet de produire une base de données orientée objet à partir d'une base de données relationnelles en définissant des correspondances entre cette base de données et les objets du langage utilisé.
Plug-ins	Composant logiciel qui ajoute une fonctionnalité spécifique à un programme informatique existant.
Workspace	Environnement de travail créé par un utilisateur contenant des projets.
Datasource	Nom donné à la connexion configurée dans une base de données à partir d'un serveur.
Spring	Spring est un cadriciel de développement java.
Multitenancy	Mode de fonctionnement d'un logiciel où plusieurs instances indépendantes fonctionnent dans un environnement partagé.

## INTRODUCTION

Depuis quelques années, la conception de bâtiments se fait à l'aide de logiciels de modélisation 3D. Ces technologies combinées à l'intérêt grandissant pour la protection de l'environnement et le développement durable donnent naissance à un besoin pour de nouveaux outils qui tirent avantage de ces nouvelles technologies. Le but du projet UBUBI est d'offrir un outil simple pour faire l'analyse du cycle de vie des bâtiments compatible avec les logiciels utilisés par les architectes et les ingénieurs du domaine du bâtiment.

## CHAPITRE 1

### Problématique et contexte

UBUBI est un logiciel libre qui sera utilisé par plusieurs universités et compagnies dans le domaine de la construction. Cette solution a pour but d'offrir un service infonuagique qui permettra de calculer les impacts environnementaux d'un bâtiment pour tout son cycle de vie. Les calculs sont faits à partir de données exportées de logiciels de conception de bâtiment et de base de données d'impacts environnementaux. Le projet UBUBI est un projet de grande envergure. Cette équipe de projet de fin d'étude s'est donc concentrée sur un sous-ensemble des fonctionnalités requises par l'équipe de chercheurs de la Polytechnique de Montréal.

Notre équipe avait pour mission d'analyser et d'implémenter les fonctionnalités reliées à la gestion des données extraites d'un bâtiment. Nous étions responsables de la création d'un module de gestion des utilisateurs et de la mise en œuvre d'un certain nombre de règles d'affaires dictées par le client.

Le module de gestion des utilisateurs permettra de gérer le travail de plusieurs utilisateurs séparément et permettra aussi de partager un projet parmi plusieurs utilisateurs dans une optique de coopération. Le second module est le module d'importation et de modification des données de bâtiment. Ce module vise à importer les fichiers de type IED provenant de l'application Revit<sup>1</sup>. Finalement l'équipe de projet de fin d'étude a eu pour tâche d'implémenter un certain nombre de règles d'affaires concernant cette application.

Le module de calcul et de résultat sera développé par une autre équipe de projet de fin d'étude.

---

<sup>1</sup><http://www.openlca.org/>

## Chapitre 2

### Objectifs du projet

Le premier objectif de ce projet de fin d'étude est de rendre le prototype UBUBI actuel plus sécuritaire en ajoutant un mécanisme d'inscription et d'authentification. Le but de cette fonctionnalité est de limiter l'accès aux divers projets. Les différents projets créés par les utilisateurs seront isolés dans différents environnements de travail, qui seront appelés «workspaces». Les propriétaires de ces «workspaces» pourront donner l'accès à d'autres utilisateurs.

Le deuxième objectif du projet est de permettre à l'utilisateur de mettre à jour l'analyse environnemental à partir de fichiers IED provenant du logiciel de conception de bâtiment Revit. Ce module d'importation devra détecter tout changement effectué dans le modèle 3D du bâtiment, par exemple : un changement de quantité, un changement de type de mur, et bien d'autres. Il devra aussi permettre de modifier l'analyse environnementale en conséquence.

Le troisième objectif du projet est de mettre en œuvre des ajouts et changements à diverses règles d'affaires qui s'appliquent lors de la modification d'une analyse environnementale. Ceci a pour but de faciliter l'utilisation du logiciel par l'utilisateur et d'empêcher l'entrée de données erronées.

## Chapitre 3

### Méthodologie

La méthodologie choisie pour le processus de développement logiciel de ce prototype d'application est basée sur la méthodologie AGILE<sup>2</sup>. Nous avons eu des rencontres, de type SCRUM<sup>3</sup>, sur une base hebdomadaire avec le promoteur et le professeur superviseur dans le but de valider notre avancement. De plus, le promoteur a été disponible pour répondre à nos questions par courriel ou vidéoconférence pour des questions urgentes. Chaque rencontre a porté sur le progrès de la semaine et les bloquants, et a permis d'établir les tâches à réaliser pour la prochaine itération. La progression du travail et la gestion du projet ont été effectuées avec l'outil Trello.

Les différentes décisions de conception et d'analyse ont été compilées et documentées tout au long du projet dans un journal d'évènement situé sur la plateforme Google drive. Il a été statué avec le promoteur et le professeur superviseur que les tests automatisés ne feraient pas partie de ce projet à cause de contraintes de temps. Il a toutefois été déterminé qu'il sera important de tester l'application et de proposer une stratégie de test au moment où cette partie du cycle de développement sera atteinte.

Pour la gestion des versions, il a été décidé d'utiliser le système Gitlab fourni par l'école. Chaque développeur peut, par contre, utiliser l'outil Git avec lequel il est le plus familier. Chaque développeur doit effectuer ses changements dans une branche avant d'être ajouté à la branche maitresse. Avant la fusion à la branche maitresse, une vérification sera effectuée par le promoteur pour s'assurer de la qualité du code livré.

L'IDE de développement a été IntelliJ, car c'est l'un des seuls outils accessibles et faciles d'utilisation qui permet l'implémentation de code Java et Scala simultanément, en plus de permettre l'utilisation du Framework « Play ». Pour faciliter le développement, une machine virtuelle contenant la base de données était fournie par le promoteur. Cette base de données accessible était modifiable avec l'outil MySQL Workbench.

---

<sup>2</sup>[http://fr.wikipedia.org/w/index.php?title=M%C3%A9thode\\_agile&oldid=139615140](http://fr.wikipedia.org/w/index.php?title=M%C3%A9thode_agile&oldid=139615140)

<sup>3</sup>[http://fr.wikipedia.org/w/index.php?title=Scrum\\_\(Boite\\_%C3%A0\\_outils\)&oldid=138835287](http://fr.wikipedia.org/w/index.php?title=Scrum_(Boite_%C3%A0_outils)&oldid=138835287)

## Chapitre 4

### Planification du projet

En raison de la nature itérative du projet, il a été difficile d'estimer de manière précise l'effort et la taille des différentes tâches. Ainsi, le responsable d'une tâche a déterminé sa planification au fur et à mesure que le projet progressait et en discutait lors de la revue SCRUM hebdomadaire. L'ensemble de la planification et les détails des tâches sont présentés à l'Annexe I suite à leur analyse.

Au cours de la révision de la planification effectuée au milieu du projet dans le rapport d'étape, nous avons comptabilisé le nombre d'heures travaillées et l'avons comparé à nos estimations initiales. Nous avons alors remarqué un écart assez important entre l'effort réel nécessaire à la réalisation des tâches et ce qui avait été prévu. Nous avons alors déterminé que cela était dû principalement au fait qu'aucun membre de l'équipe n'est un expert en application web et qu'aucun des membres de l'équipe n'a eu l'occasion de travailler avec le cadriciel « Play ». Nous avons donc décidé à partir de la mi-session et du rapport d'étape d'être plus généreux dans nos prédictions initiales pour chaque élément pour ainsi éviter les mauvaises surprises.

De plus, tout au long du développement, l'analyse plus en profondeur des fonctionnalités a en effet fait apparaître des éléments qui n'avaient pas été discutés avec le promoteur et qui ont donc eu besoin d'être éclaircis. Par exemple, la création de «workspaces», de projets et d'utilisateurs avait été discutée et élaborée en détails, mais la modification des permissions en dehors de l'ajout n'avait jamais été abordée. Les fonctionnalités de recherche d'utilisateur et de projet n'avaient pas non plus été abordées. Il a donc été nécessaire de discuter des éléments qui avaient déjà été considérés comme presque terminés et de déterminer la direction que nous voulions prendre. Finalement, il a aussi été nécessaire d'établir une période de consolidation et de tests de fin de projet pour s'assurer de la qualité de la livraison finale.

## Chapitre 5

### Livrables

Tableau 2: Description des artefacts

Nom de l'artefact	Description
Update(s) SCRUM	Chacune des rencontres SCRUM comporte ses livrables hebdomadaires. La nature de ces livrables varie en fonction des objectifs qui ont été déterminés lors de la dernière rencontre.
Proposition de projet	Document de planification initiale du projet.
Rapport d'étape	Document servant à informer diverses parties prenantes de la progression du projet et de l'atteinte des objectifs. Il permet de documenter le progrès du projet à la mi-session.
Présentation individuelle	Présentation des décisions prises pendant le projet, les travaux effectués et les moyens pris pour adresser les éléments bloquants tout au long du projet.
Rapport final	Le rapport final du projet de fin d'études contenant tous les éléments de l'analyse, de la conception et de l'implémentation de la solution.
Code d'intégration de l'authentification	Code du module d'authentification.
Code d'intégration de l'importation des données du bâtiment	Code du module de gestion des données et prototype fonctionnel.

## Chapitre 6

### Sommaire des travaux réalisés

Plusieurs tâches ont été réalisées lors du projet. Ces tâches sont les suivantes: la conception du système de gestion d'utilisateurs et de «workspaces», la création des pages d'enregistrement et de connexion, l'envoi de fichiers IED par un formulaire web, la création d'un processus d'énergie, l'ajout d'échange avec les bons types de flux, la mise à jour du modèle, la gestion des unités lors d'ajout d'échanges et le changement du code Uniformat d'un objet. Certaines de ces tâches ont nécessité un plus grand effort d'analyse que d'autres, tel que présenté dans les sous-sections suivantes.

#### 6.1 Conception du système d'utilisateurs et de Workspaces

Pour le système d'authentification, nous avons envisagé deux solutions possibles. La première était de trouver un plug-in pour le cadriciel Play qui permettrait d'ajouter un système d'authentification local au serveur et qui ferait une grande partie du travail et laisserait l'équipe se concentrer sur d'autres tâches. Par contre, lors de nos recherches, la plupart des plug-ins d'authentification, tels que «Play! Authenticate»<sup>4</sup> et «Play-pac4j»<sup>5</sup>, que nous avons pu trouver nous limitait à authentifier l'utilisateur à l'aide d'autres services, tels que Facebook et Google, ce qui ne répondait pas aux besoins du projet. Le seul plug-in qui semblait permettre l'ajout de comptes locaux était non fonctionnel. Le projet de démonstration inclus avec le plug-in ne permettait pas la création de comptes. Il a donc été décidé d'implémenter une solution nous-même.

Nous avons finalement convenu d'utiliser trois tables. La première table serait les utilisateurs (Users). Celle-ci contiendrait les informations sur l'utilisateur, leurs noms, prénoms, adresse courriel et le mot de passe. Ensuite, nous avons créé une table «workspace», qui contient son nom ainsi qu'une référence vers son créateur. Finalement, une table contenant la liste des utilisateurs a été créée. Chaque ligne de celle-ci contient une référence vers un utilisateur et une référence vers un «workspace». Les mots de passe sont hachés par la fonction SHA-256 incluse dans les bibliothèques de base de Java. L'idée étant de permettre à des utilisateurs autres que les créateurs originaux d'utiliser les «workspaces». Nous avons réalisé que ce design, bien que simple, serait amplement suffisant.

---

<sup>4</sup><https://joscha.github.io/play-authenticate/>

<sup>5</sup><https://github.com/pac4j/play-pac4j>



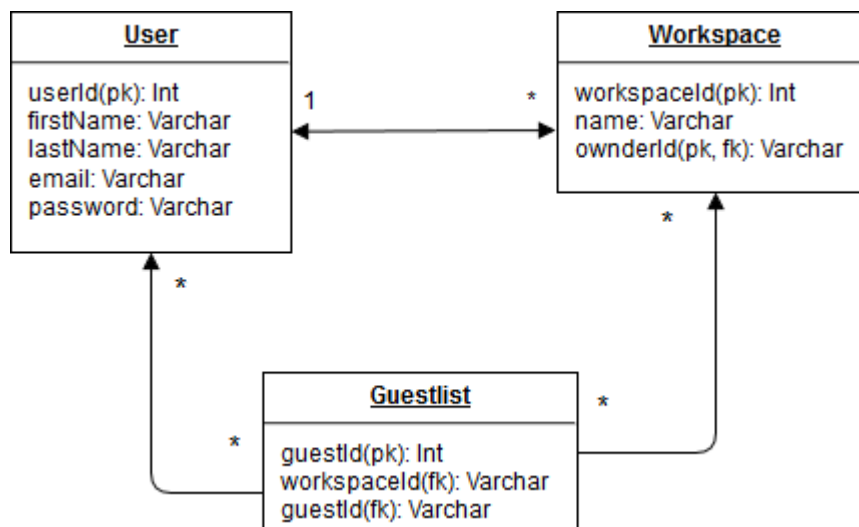


Figure 1: Schéma de la section usager de la base de données

## 6.2 Création de «workspaces»

Afin de séparer les environnements des utilisateurs, nous avons identifiées trois solutions possibles. Une première solution serait de garder les données de tous les utilisateurs dans les mêmes tables de la même base de données et effectuer la séparation grâce à la clé primaire des utilisateurs sur la base de données. Il aurait donc fallu remanier un peu l'architecture de la base de données pour intégrer les colonnes «userId» dans les tables affectées. Cela nous aurait forcé à modifier les tables d'OpenLCA, ce qui n'était pas permis selon nos contraintes, car le client voulait garder la compatibilité de son logiciel avec OpenLCA. Une autre approche aurait été d'effectuer la séparation par base de données, créant ainsi une base de données par environnement de travail. Finalement, pour la dernière option, on pouvait séparer par schéma, créant ainsi un nouveau schéma par environnement de travail dans la base de données.

Puisqu'il n'était nécessaire que de répliquer la structure des tables, nous avons décidé de prendre l'approche de la réplcation du schéma patron et de créer un schéma par «workspaces». De plus, il a été décidé de dupliquer toute la base de données. Toute analyse utilisant le modèle d'OpenLCA demande au minimum environ 80% des données contenues dans celle-ci, alors tenter de ne copier que les données utiles pour un «workspace» spécifique ne sauverait pas beaucoup d'espace en plus de demander énormément de temps de développement. Selon les commentaires émis par le représentant du client, l'espace ne sera pas un problème en raison des coûts peu élevés de stockage sur AWS.

Une fois la décision prise pour la méthode de création des «workspaces», la première étape consistait à pouvoir créer les schémas à partir d'un schéma patron. Pour se faire, on utilise la fonctionnalité de «mysqldump», utile lors de l'installation de MySQL pour télécharger un fichier SQL contenant l'information de la copie du schéma patron. Ensuite, encore avec une fonction de MySQL, on utilise le script SQL généré pour créer un nouveau schéma en tout point similaire avec le schéma patron. Utiliser un script de copie de base de données aurait été plus complexe, plus long et aurait pu causer plus de problèmes. De plus, le tout serait plus difficile à supporter en cas de modification. Ensuite, pour faire la création du nouveau schéma, on utilise l'exécutable mysql de concert avec le fichier créé par «mysqldump». Ce schéma est une copie sans donnée du schéma du patron. Le nouveau schéma créé a le nom choisi pour le «workspace» par l'utilisateur et ce nom sera celui qui sera utilisé pour choisir le «workspace» de l'utilisateur.

### 6.3 Changement de workspace

Une fois le «workspace» créé, il faut pouvoir y accéder à partir de l'application. Un défi technique s'est révélé ici, car avec le logiciel libre «Hibernate», il n'est pas facile de changer le contexte de la connexion. Donc, pour surmonter cela, on a fait des recherches sur la bonne manière d'accéder à un autre schéma lors de l'exécution. Plusieurs solutions ont été envisagées, telles que l'utilisation de HibernateShards<sup>6</sup>, implémenter un «Entity Manager» personnalisé, et le «multi tenancy».

L'approche avec HibernateShards semblait prometteuse puisque le cadriciel semblait permettre d'accéder à plusieurs bases de données à l'exécution pour manipuler des données. Mais, ce n'était pas exactement ce dont on avait besoin comme solution, car ce cadriciel s'applique dans le cas de partitionnement horizontal. Le partitionnement horizontal divise une table en plusieurs tables. Chaque table contient alors le même nombre de colonnes, mais moins de lignes. Par exemple, une table qui contient 1 milliard de lignes peut être segmentée horizontalement en 12 tables, chacune de ces tables représentant un mois de données lors d'une année. Toute requête recherchant les données d'un mois spécifique ne consulte que la table concernée. De plus le développement et support semblent être arrêtés sur ce projet libre.

La deuxième approche envisagée était de créer plusieurs data sources (c.-à-d. «SessionFactory») qui permettent de manipuler les données dépendamment de l'utilisateur connecté. Il y avait deux façons d'effectuer cette approche. Avec le cadriciel Play, selon la documentation, on pouvait modifier le fichier de configuration de l'application et ajouter autant de configuration de base de données qu'on le souhaite et ensuite changer de contexte dépendamment de la situation. Spring<sup>7</sup>

---

<sup>6</sup><http://hibernate.org/others/>

<sup>7</sup><https://spring.io/>

permet de contourner le `dataSource` avec `AbstractDataSource` et se faire sa propre `dataSource` qui sera utilisée pour les transactions. Alternativement, on aurait pu aussi définir une `SessionFactory` pour chaque tenant. Dans la version 2 du cadriciel Play, la solution proposée pour accéder à de multiples bases de données est de créer une connexion pool qui serait écrite dans le fichier de configuration à l'avance<sup>8</sup>. Ensuite, il serait possible de changer le connecteur puisque l'information se trouve dans le dossier de configuration. Évidemment, cette solution ne semblait pas attrayante dû au manque de dynamisme de la liste de schéma qu'il faudrait créer dans le fichier de configuration. Mais cette approche n'était pas « scalable » et maintenable pour plusieurs utilisateurs créés dynamiquement et utilisant le système en même temps. On veut pouvoir créer dynamiquement les schémas et tout de suite pouvoir les changer sans avoir à recharger l'application. De plus, on veut pouvoir changer le contexte du connecteur Hibernate pour ainsi éviter un « refactoring » majeur du code, étant donné que ce que Play permet de faire est de simplement rajouter dans le fichier de configuration les diverses bases de données.

La troisième approche était de créer son propre « Entity Manager » avec JPA, mais les actions qui n'utilisent pas explicitement JPA ne sont pas adressées. Il pourrait servir à contourner l'utilisation usuelle de JPA. Cette approche a comme désavantage qu'il faudrait effectuer un « refactoring » important du code et que les appels des entités Hibernate se feraient encore dans le contexte par défaut.

La quatrième approche est de tenter le « multi tenancy ». Le terme « multi tenancy » s'applique lorsqu'on a une architecture ou une seule instance d'une application qui sert plusieurs clients (de l'anglais « tenants ») différents<sup>9</sup>. C'est une solution logicielle comme service très commune quand on souhaite isoler les données par clients. Il faudrait donc changer les classes pour qu'elles soient « tenant Aware », et changer la perspective de la connexion. Il y a beaucoup de solutions d'implémentation pour une architecture qui serait avec Spring, JPA et Hibernate. En effet, Spring offre plusieurs classes qui supportent le « multi tenancy » et qui permettent le changement dynamique de la source des données. Par contre, dans le cas de la version 2 du cadriciel Play, il y a peu de classes internes qui supportent le « multi tenancy ». Effectivement, les quelques exemples disponibles montrent une approche que ressemble à la troisième approche de solution proposée ou on crée un « Entity Manager » personnalisé<sup>10 11</sup>.

---

<sup>8</sup> <https://www.playframework.com/documentation/2.5.x/ScalaDatabase>

<sup>9</sup>

[https://docs.jboss.org/hibernate/orm/current/userguide/html\\_single/Hibernate\\_User\\_Guide.html#multitenancy](https://docs.jboss.org/hibernate/orm/current/userguide/html_single/Hibernate_User_Guide.html#multitenancy)

<sup>10</sup> <https://alots.wordpress.com/2014/04/10/multi-tenant-approach-with-java-8-in-a-play-project/>

<sup>11</sup> <https://github.com/rajendrag/play-jpa-multitenant>

Pour terminer, bien que ce ne fût pas possible pour nous de trouver une solution dans le temps alloué, nous croyons qu'il est possible d'atteindre l'objectif en contournant les problématiques technologiques rencontrées. Il faudrait appliquer une deuxième approche couplée avec une édition dynamique du fichier de configuration. Cette méthode de contournement serait en théorie possible et nous avons discuté de son implémentation, mais à cause du besoin d'avancer d'autres fonctionnalités, le travail sur le «workspace» a été interrompu.

## 6.4 Mise à jour du modèle

Le but de la mise à jour du modèle est de permettre à l'utilisateur de téléverser une série de fichiers IED vers le serveur représentant le même bâtiment après que de nouvelles modifications aient été apportées au modèle dans Revit. Ces changements proviennent du fait qu'un bâtiment changera plusieurs fois au cours du design, à la fois pour mieux répondre aux besoins pour lequel celui-ci est conçu et suite à une analyse de cycle de vie qui ne serait pas satisfaisante. L'utilisateur entre certaines informations dans le logiciel avant de faire l'analyse, le but de la mise à jour du modèle est de s'assurer de ne pas perdre les informations entrées manuellement par celui-ci. Quelques options ont été prises en compte pour l'implémentation de cette fonctionnalité.

La première a été l'utilisation de l'introspection pour créer un algorithme de comparaison des objets qui requièrent très peu de modifications en cas de modification du modèle. Deux raisons ont mené à l'abandon de cette option. La première est que l'introspection requiert le changement des paramètres de sécurité de la machine virtuelle Java. Demander le changement de ces paramètres aurait premièrement réduit la confiance des utilisateurs futurs envers l'application. De plus, faire ces manipulations aurait demandé un plus grand niveau de compétence technique, ce qui aurait sans doute repoussé des utilisateurs potentiels.

La solution choisie a été de simplement créer un nouveau modèle à partir du nouveau fichier, sans le sauvegarder dans la base de données, et comparer celui-ci au modèle original instance par instance. Cette solution, plus traditionnelle, a demandé beaucoup de code assez répétitif (une grande quantité de fonctions «equals», qui ne pouvaient se retrouver dans le code des objets OpenLCA), ce qui a donc pris énormément de temps. De plus, le mécanisme de création de nouveaux modèles à partir de fichiers IED sauvegardait ceux-ci dans la base de données immédiatement lorsque ceux-ci étaient créés ce qui nous a empêché de réutiliser ce code.

## 6.5 Création d'un processus d'énergie

Concernant la création du processus d'énergie, il y a eu peu de choix de conception. La majorité de la dynamique du problème venait du fait que le module d'importation et de création de processus était complexe à comprendre. La création d'un processus d'énergie a demandé l'implémentation d'une entité BIMEnergyType qui contient les différents types d'énergie dans un énumérateur ainsi qu'une méthode permettant de retourner leurs unités respectives. Il a aussi fallu modifier la création d'un flow pour permettre d'utiliser les nouvelles unités ainsi créées. La création des processus d'énergie comprend la création d'un processus racine ayant comme clé prédéterminée la lettre «G» qui est reliée au modèle. Par la suite sont créés les sous-processus d'énergie qui sont finalement reliés au processus racine d'énergie.

## 6.6 Changements du code unformat des objets et types d'objets

Pour ce qui est du changement de code Unformat, le choix de conception principal a été de décider la manière dont les codes seraient enregistrés. Nous avons le choix entre un fichier texte, un fichier de type XML ou encore de les ajouter à la base de données. En raison de notre méthode de réplication de base de données expliquée dans la section des «workspaces», nous avons décidé de créer une table dans la base de données et de la peupler grâce à un script d'importation. Cette méthode nous fournit une plus grande flexibilité pour ce qui est de la modification des codes s'ils devaient changer, et cela nous permettrait potentiellement de faire l'utilisation de «trigger SQL» ainsi que d'autres fonctionnalités semblables. L'implémentation n'est pas complète à cause du manque de temps, seule l'importation de données a été complétée. Il reste à implémenter la méthode qui limite les choix de codes ainsi que la structure permettant de refaire le lien vers l'arbre une fois que celui-ci a été brisé.

## 6.7 Gestion des unités

Le flux de référence désigne la quantité du produit analysé ainsi que la quantité de consommables utilisés par ce produit nécessaire pour couvrir les besoins de l'unité fonctionnelle. Dans la fonction d'ajout d'échange, il affiche seulement les bonnes unités pour le type de flux choisis. Par exemple, il est plus pertinent de représenter un mur en termes de m<sup>3</sup> plutôt que d'utiliser des kJ. Donc, dans ce cas, la liste devrait juste contenir des unités de surfaces et non des unités d'énergie.

Il a fallu créer une méthode qui retourne seulement les unités qui répondent à certains critères, notamment c'est l'identifiant du flux qui a été choisi. De ce fait, la liste des unités se rafraichit maintenant quand le flux choisi change.

## 6.8 Ajout d'échange avec les bons types.

Pour la gestion de l'ajout des échanges, deux validations ont été mises en place pour filtrer le type d'échange que l'on peut ajouter. Le premier filtre mis en place est la possibilité d'ajouter des échanges (liens) à un type d'objet vers un matériel ou un processus background. Le second filtre mis en place est la possibilité d'ajouter des échanges (liens) à un matériel vers un processus d'arrière-plan. Dans ces deux cas, ils étaient requis de permettre l'ajout d'échanges. Pour ce faire, il faut regarder dans la base de données pour s'assurer que le processus ou le matériel vers lequel on fait le lien est bien un processus ou matériel background. Pour savoir si c'est un échange ou processus background, on vérifie si son identifiant est présent dans la table «tbl\_BIMExchange» ou «tbl\_BIMProcess». Cette validation se fait dans la méthode «allowInsert» dans la classe Exchange dans le modèle.

## 6.9 Empêcher la suppression de certains type d'échanges.

Encore dans le paradigme des échanges, un autre aspect des règles d'affaires était la gestion des suppressions de certains types d'échanges. Il fallait empêcher la suppression d'échanges de produits sortants. Pour savoir si c'est un échange de produit sortant on vérifie dans la colonne «quantitativeExchange» dans la table «tbl\_process». Une autre validation était d'empêcher la suppression d'échanges ou de processus qui a été généré par BIM. On reconnaît un processus qui a été généré par BIM, car il est créé à l'importation non ajoutée par l'utilisateur et son identifiant se trouve donc dans la table «tbl\_BIMExchange». Pour se faire, il fallut rajouter des validations avec la BD dans la méthode «allowDelete» dans la classe Exchange dans le modèle.

## Chapitre 7

### Risques

#### 7.1 Risques prévus

Tableau 3: Risques prévus

Risque	Impact	Probabilité	Mitigation / Atténuation
Manque de temps en raison d'une mauvaise estimation de la charge de travail et de l'effort requis en début de projet.	Élevé	Forte	Le développement de type itératif et l'utilisation de la méthode AGILE serviront à atténuer ces risques.
Mauvaise compréhension des besoins initiaux et du domaine.	Élevé	Modéré	Une communication constante avec le promoteur et des livrables hebdomadaires de petite taille devraient permettre de corriger la situation si elle survient.
Problème de communication dans l'équipe ou avec le promoteur.	Moyen	Forte	L'utilisation d'outil de collaboration comme Trello et des rencontres fréquentes devrait permettre d'éviter le problème.
Problème avec la technologie utilisée en raison du manque d'expérience.	Élevé	Forte	La compilation et le partage de guides et d'informations dans l'équipe devraient réduire le risque de blocage.
Demande de changement en cours de projet.	Faible	Forte	L'utilisation d'une méthode itérative de développement devrait permettre de réduire l'impact.

Ralentissement des développements dus à des facteurs externes.	Moyen	Modéré	Une révision de la charge de travail hebdomadaire en fonction des éléments connus (période d'examen) devrait réduire le risque.
--	-------	--------	---

## 7.2 Risques Encourus

Lors du projet, plusieurs risques se sont matérialisés. Par exemple, à quelques reprises, les besoins du domaine ont mal été compris, ce qui a typiquement requis une rencontre en personne pour régler le problème. Quelques exemples de besoins ayant mal été compris ont été la mise à jour du modèle, la création d'un processus d'énergie et le mécanisme de «workspaces».

Ensuite, nous avons eu plusieurs problèmes avec les technologies utilisées, autant à cause du manque d'expérience qu'avec les limitations de celles-ci. Par exemple, le Cadriciel Play implémente une interprétation du patron de conception Modèle-Vue-Contrôleur avec laquelle nous n'étions pas familiers, ce qui a demandé beaucoup d'essais-erreur, car la documentation du cadriciel n'était pas optimale. Contrairement à d'autres cadriciels plus populaires, il est difficile d'obtenir des informations pour des problèmes précis et il est souvent nécessaire d'utiliser la documentation générique et incomplète du cadriciel. C'est un problème récurrent quand on utilise une nouvelle version d'une technologie existante.

L'ORM Hibernate nous a aussi posé plusieurs problèmes lors du développement du mécanisme de «workspaces». En effet, on a passé beaucoup de temps à s'acharner pour trouver une solution respectant les contraintes imposées.

De plus, un problème de communication est survenu entre le client et le promoteur. Le client a envoyé le mauvais fichier pour la liaison des données et n'a pas été disponible depuis. Cela a forcé l'abandon de cette fonctionnalité.

Finalement, en raison de conflit d'horaire et de facteurs externe comme les travaux de fin de session et les problèmes reliés à l'utilisation du GitLab, un nombre d'heures plus petit que prévu a été disponible pour avancer le travail de manière productive dans les dernières semaines de développement. Tous ces contretemps ont contribué au ralentissement du développement du projet et quand cette perte de temps a été combinée avec notre analyse initiale optimiste cela nous a donc forcés d'abandonner certaines fonctionnalités du projet.



## Chapitre 8

### Post mortem du projet

#### 8.1 Objectifs non-atteint

Certains objectifs n'ont pas été atteints par manque de temps et d'expertise. Les détails de ces objectifs ont déjà été discutés dans les sections précédentes, mais nous avons décidé de prendre le temps de faire une liste exhaustive.

La création dynamique de «workspaces» a été implémentée, mais aucune solution n'a été implémentée pour ce qui est du changement dynamique entre les «workspaces». Ce code sera donc mis dans une branche à part pour pouvoir être utilisé par les prochains développeurs. La création et la gestion des différents détails des comptes des usagers ont été implémentées, mais comme le but principal de ce module était de permettre de séparer les «workspaces», le code n'a pas encore d'utilité. Ce code sera donc mis dans une branche à part pour pouvoir être utilisé par les prochains développeurs.

Les tâches reliées à la liaison automatique des données ont aussi été abandonnées, car le client ne nous a pas envoyé les bons fichiers et ne pouvait pas les fournir avant la fin des développements. Le changement de code Unifomat n'a pas été complété, il reste à implanter les règles d'affaires pour limiter les modifications. Comme l'importation des codes Unifomat ne touche pas d'autres fonctionnalités, le script d'importation ainsi que la structure de table ont été inclus dans la version finale du code. La mise à jour du modèle n'a pas pu être complétée. Le mécanisme actuel d'importation de fichiers est tout simplement trop complexe et a demandé trop de temps de développement pour simplement comprendre. Le travail effectué jusqu'à maintenant se retrouvera dans une branche sur le Gitlab. Finalement certaines règles d'affaires n'ont pas été implémentées et la gestion des paramètres locaux et globaux n'a pas été analysée ou implémentée par manque de temps.

#### 8.2 Changements qui auraient pu être apportés en cours de projet

Certains changements auraient dû être apportés ou du moins abordés et discutés en détail lors du projet. Nous considérons que si ces changements avaient été implémentés ils auraient pu avoir un impact positif sur le développement.

Par exemple, l'horaire des rencontres choisi n'était pas optimal pour tous les membres de l'équipe. En fonction de notre horaire respectif, la meilleure journée pour le développement était le mercredi soit la veille de la rencontre. Cela avait pour effet de nous aligner pour travailler avec un modèle de juste-à-temps en plus de faire en sorte que des détails de la rencontre précédente pouvaient facilement être oubliés. Cela causait aussi des problèmes quand on atteignait un élément bloquant et nous a empêché d'utiliser notre méthode de développement agile au meilleur de sa capacité. Nous aurions dû à la mi-session ou même avant, proposer un horaire différent qui aurait mieux convenu à notre équipe sans causer de problèmes au client.

Nous pensons aussi que d'avoir une rencontre une fois par semaine à distance avec seulement les membres de l'équipe pour se garder à jour sur le reste du projet et pour explorer le code aurait été un bon moyen d'éviter de tomber dans les mêmes pièges. L'utilisation du «peerprogramming» ou une variation de celui-ci aurait potentiellement pu nous permettre d'augmenter la qualité du code et aurait pu nous permettre de mieux remplir nos objectifs.

### 8.3 Leçons tirées

En plus des changements précédemment explorés, nous avons décidé de revenir sur notre expérience et d'essayer d'en tirer le plus de leçons possibles. Une leçon tirée de ce projet est de bien faire les recherches sur les technologies qui seront utilisées pour s'assurer qu'elles soient adéquates et bien documentées. Il faut prendre plus de temps pour s'assurer qu'une solution est vraiment possible avant de commencer l'implémentation pour éviter de perdre du temps précieux.

Une autre leçon est qu'il faut faire attention de ne pas tomber dans le piège de penser que la solution est juste sur le point d'être découverte et de s'entêter pendant plusieurs semaines gaspillant ainsi des ressources et du temps de développement. Ce projet nous a montré qu'il était important de faire preuve de débrouillardise et de considérer les solutions de contournements comme étant des solutions viables et non simplement du «patching» de mauvaise qualité. Quitte à revenir travailler sur la fonction en fin de projet ou dans une version future quand la technologie sera adaptée pour le rendre possible.

Finalement, nous croyons qu'il est important d'adopter une attitude plus proactive et de ne pas se laisser abattre par les limitations et les contraintes, mais plutôt de trouver un moyen de faire ressortir les possibilités en changeant les règles et notre vision des choses.

## Chapitre 9

### Contribution des membres

Cette section décrit dans les grandes lignes les contributions personnelles de chaque membre de l'équipe. Pour chaque livrable et chaque partie principale du projet, l'auteur principal sera listé. Il est important de noter que la plupart des décisions de conception ainsi que les détails de l'implémentation ont été longuement discutés en groupe, tous les membres de l'équipe ont donc contribué à chacun des livrables.

#### 9.1 Gergel, Sébastien

Durant la session, Sébastien a travaillé sur plusieurs parties du projet. Ces livrables principaux ont été la production de la proposition de projet initiale, ainsi que celle du rapport d'étape à la-mi session. De plus, il a implémenté la création d'un processus d'énergie et a travaillé sur le changement du code Uniformat. Il a aussi été responsable de la communication au nom de l'équipe et s'occupait de faire les résumés de rencontre.

#### 9.2 Jean, YoméMessena

Yomé a travaillé sur la structure de la base de données pour les utilisateurs ainsi que les «workspaces». Il a travaillé sur la mécanique de fonctionnement des «workspaces» dans l'application. Ces derniers ont demandé énormément de son temps. Finalement, il a travaillé sur les différentes règles d'affaire du projet.

#### 9.3 Picard, Etienne

Durant la session, Étienne a contribué à plusieurs parties du travail. Il a implémenté les formulaires d'authentification et d'inscription, plus le mécanisme de connexion final. Il a effectué l'importation des fichiers de données à travers le client. Cette tâche n'a pas été mentionnée plus haut, car elle ne contenait pas de vrais problèmes d'ingénierie. Il a suffi de trouver comment faire un téléversement à partir du code JavaScript, puis utilisé une fonction existante pour les transformer et sauvegarder dans la base de données. Finalement, il a travaillé sur la mise à jour du modèle, qui n'a pu être complété dû à un manque de temps.

## 9.4 Contributions à parts égales

Cette section décrit les tâches sur lesquelles les trois membres de l'équipe ont contribué à parts égales. Ces sections sont la présentation du PFE. Chaque membre a fait les sections qu'il allait présenter, en plus de relire celle des autres et de donner des conseils pour les améliorer. Chacun a écrit plusieurs sections du rapport final, qui a été relu par les autres membres de l'équipe. Et finalement, les trois se sont mis ensemble pour faire le «merge» final du code sur Gitlab.

## CONCLUSION

En conclusion, nous avons pu constater que le projet proposé était de trop grande envergure pour le temps alloué. Plusieurs tâches ont été accomplies, telles que la gestion des utilisateurs et la création d'un processus d'énergie. Par contre, plusieurs autres tâches n'ont pu être accomplies de manière complète, telle que la création de «workspaces» et la mise à jour du modèle. Les tâches accomplies en partie ont tout de même été incluses dans une branche pour permettre aux futurs développeurs de profiter du travail accompli. Il aurait donc fallu allouer plus de temps à ce projet ou revoir de manière plus drastique nos méthodes de développement à la-mi session. Une équipe avec des connaissances plus approfondies en développement web ou de la technologie utilisée aurait peut-être été capable de remplir les objectifs.

## LISTE DE RÉFÉRENCES

Page d'accueil du logiciel OpenLCA. Page consultée le 14 août 2017 à partir de <http://www.openlca.org/>

Page d'accueil du logiciel Revit. Page consultée le 14 août 2017 à partir de <https://www.autodesk.ca/en/products/revit-family/overview>

Méthode agile. (2017, août 9). *Wikipédia, l'encyclopédie libre*. Page consultée le 14 août 2017 à partir de [http://fr.wikipedia.org/w/index.php?title=M%C3%A9thode\\_agile&oldid=139615140](http://fr.wikipedia.org/w/index.php?title=M%C3%A9thode_agile&oldid=139615140).

Scrum (Boite à outils). (2017, juillet 10). *Wikipédia, l'encyclopédie libre*. Page consultée le 14 août 2017 à partir de [http://fr.wikipedia.org/w/index.php?title=Scrum\\_\(Boite\\_%C3%A0\\_outils\)&oldid=138835287](http://fr.wikipedia.org/w/index.php?title=Scrum_(Boite_%C3%A0_outils)&oldid=138835287).

Page d'accueil du logiciel Revit. Page consultée le 14 août 2017 à partir de <https://trello.com/>

Page d'accueil du module play-authenticate. Page consultée le 14 août 2017 à partir de <https://joscha.github.io/play-authenticate/>

Page d'accueil du module play-pac4j. Page consultée le 14 août 2017 à partir de <https://github.com/pac4j/play-pac4j>

Page d'accueil du projet HibernateShards. Page consultée le 14 août 2017 à partir de <http://hibernate.org/others/>

Page d'accueil du CadricielSpring. Page consultée le 14 août 2017 à partir de <https://spring.io/>

Multi-Tenancy using JPA, Spring and Hibernate parts 1 and 2. Pages consultées le 14 août 2017 à partir de

<https://www.ricston.com/blog/multitenancy-jpa-spring-hibernate-part-1/>

<https://www.ricston.com/blog/hibernate-jpa-spring-tenants-part-2/>

Section sur le Multitenancy de la documentation Hibernate. Page consultée le 14 août 2017 à partir de

[https://docs.jboss.org/hibernate/orm/current/userguide/html\\_single/Hibernate\\_User\\_Guide.html#multitenancy-intro](https://docs.jboss.org/hibernate/orm/current/userguide/html_single/Hibernate_User_Guide.html#multitenancy-intro)

Exemple d'implémentation de multi tenancy avec Play 2. Page consultée le 14 août 2017 à partir de

<https://github.com/rajendrag/play-jpa-multitenant>

Documentation Play 2 pour connection sur de multiples base de données. Page consultée le 14 août 2017 à partir de <https://www.playframework.com/documentation/2.5.x/ScalaDatabase>

## Annexe A

### Plan de Travail

S: Gergel, Sébastien

E: Picard, Étienne

J: Jean, YoméMessena

**Tableau 4: Plan de travail**

#	Debut (dd/mm)	Fin (dd/mm)	Effort estimé s	Effort réel	Tâches / Jalon	Livrable(s) / Artefact(s)	Responsables(s)
1	26/04		S:3, E:3, J:3	S:3, E:3, J:3	Première rencontre de l'équipe		S, E, J
1.1	26/04	26/04	S:1, E:1, J:1	S:1, E:1, J:1	Rédaction du contrat d'équipe		S, E, J
2	10/05		S:3, E:3, J:3	S:3, E:3, J:3	1 <sup>re</sup> rencontre avec le promoteur et présentation du projet		S, E, J
2.1	10/05	20/05	S:5, E:5, J:5	S:6, E:4, J:3	Création d'un prototype de login et familiarisation avec les outils de développement		S, E, J
3	20/05		S:2, E:2, J:2	S:1, J:1	2 <sup>e</sup> rencontre avec le promoteur	Un prototype du module de login par membres de l'équipe	S, E, J
3.1	20/05	25/05	S:3, E:3, J:3	S:3, E:3, J:3	Analyse préliminaire du module de login et de la gestion des «workspaces»		S, E, J
4	25/05		S:2, E:2, J:2	S:1, E:1, J:1	3 <sup>e</sup> rencontre avec le promoteur	Analyse préliminaire	S, E, J
4.1	25/05	01/06	S:2,	S:3, E:3,	Créer les tables		S, E, J

			J:5, E:3	J:3	globales dans Hibernate		
4.2	25/05	TBD	S:1, J:1, E:1	S:5, E:2, J:3	Création du mécanisme de «workspaces»		S, E, J
4.3	25/05	01/06	S:5, J:1	S:7, E:2, J:1	Proposition de projet		S, P
5	01/06		S:2, E:2, J:2	S:1, E:1, J:1	4 <sup>e</sup> rencontre avec le promoteur	Proposition de projet Création des tables dans Hibernate	S, E, J
5.1	01/06	08/06	E:5	E:4	Création des formulaire d'authentificatio n		E
5.2	01/06	TBD	S:5, J: 5	S:1, J:7	Création du mécanisme de « workspaces »		S, J
6	08/06		S:2, E:2, J:2	E:1	5 <sup>e</sup> rencontre avec le promoteur	Prototype du module de Login et d'inscription	E
6.1	08/06	TBD	J:5	J:15	Module de gestion de «workspaces»		J
6.2	08/06	13/06	E:4	S:2, E:4	Module d'importation des données		S, E
6.3	08/06	13/06	S:1	S:3	Début de la rédaction du rapport d'étape		S
6.4	08/06	13/06	S:1	S:3	Début de la consolidation des choix de conception et des artefacts		S
6.5	08/06	13/06	E:6	E:8	Module d'authentificatio n		E
7	13/06		S:2, E:2, J:2	S:1, E:1, J:1	6 <sup>e</sup> rencontre avec le promoteur	Module d'importatio n des données	S, E, J
7.1	13/06	29/06	S:5,	S:6, E:4,	Rédaction du		S, E, J



			E:2, J:1	J:1	rapport d'étape		
7.2	13/06	29/06	S:10	S:10	Création d'un processus d'énergie		S
7.3	13/06	29/06	E:10	E:40, S:5	Mise à jour du modèle	Mise à jour du modèle	E
7.4	13/16	28/06	J:10	J:15, S:1	Module de gestion de «workspaces»	Module de gestion de «workspaces»	J
7.5	28/06	28/06	E:2, J:2	E:3, J:3	Consolidation des différents travaux réalisés		E, J
8	22/06		S:1, E:1, J:1	S:0, E:0, J:0	Rencontre annulée		S, E, J
9	29/06		S:1, E:1, J:1	S:1, E:1, J:1	8 <sup>e</sup> rencontre avec le promoteur	Rapport d'étape	S, E, J
9.1	29/06	-	S:1, E:1, J:1	S:0, E:0, J:0 (annulé)	Création de la structure de données		-
9.2	29/06	06/07	S:5	S:5	Création d'un processus d'énergie		S
9.3	29/06	06/07	J:2	J:5	Module de gestion de «workspaces»		J
10	06/07		S:1, E:1, J:1	S:1, E:1, J:1	9 <sup>e</sup> rencontre avec le promoteur		S, E, J
10.1	06/07	13/07	S:5	S:7	Création d'un processus d'énergie		S
10.1	06/07	-	S:1, E:1, J:1	S:0, E:0, J:0 (annulé)	Liaison automatique des données		-
11	13/07		S:1, E:1, J:1	S:1, E:1, J:1	10 <sup>e</sup> rencontre avec le promoteur		S, E, J
11.1			S:1, E:1, J:1	J:2	Ajout d'échanges avec le bon type de flux		J
11.	13/07	27/07	S:15,	S:0, E:0,	Préparation pour		S, E, J

2			E:15, J:15	J:0 (annulé )	la présentation du projet au représentant utilisateur		
11. 3	13/07	27/07	S:1	S:1	Changement du processus d'énergie selon feedback	Création d'un processus d'énergie.	S
11. 4	13/07	27/07	S:10	S:5	Modification du code Unifomat		S
12. 3				J:4	Implémentation des règles d'affaires		J
12	20/07		S:3, E:3, J:3	S:1, E:1, J:1	11 <sup>e</sup> rencontre avec le promoteur	Présentation au représentant usager Pascal Lesage (annulé)	S, E, J
12. 1	20/07	27/07	S:1, E:1, J:1	S:0, E:0, Y:0 (annulé )	Gestion des paramètres globaux et Locaux		-
12. 2	20/07	15/08	S:15, E:5, J:5	S:15, E: 10, J:10	Rédaction de la documentation du projet		S, E, J
	20/07	27/07	S:5	S:5	Modification du code Unifomat	Importation des données Unifomat	S
12. 3				J:5	Implémentation des règles d'affaires		J
13	27/07		S:1, E:1, J:1	S:1, E:1, J:1	12 <sup>e</sup> rencontre avec le promoteur		S, E, J
13. 1	27/07	TBD	S:5, E:5, J:5	S:11, E:15, J:15	Préparation de la présentation du PFE		S, E, J
13. 2	27/07	15/08	S:10, E:10, J:10	S:9, E:10, J:15	Consolidation du code et des travaux réalisés et ajout de commentaires	Branche de code non utilisée	S, E, J
13. 3	27/07	15/08	S:10, E:10,	S:6, E:5, J:5	Tests exhaustifs		S, E, J

			J:10				
14	03/08	08/08	-	-	Semaine de présentation du PFE	Présentation du PFE	S, E, J
15	09/08	15/08	-	-		Remise du rapport final + code	S, E, J

## Annexe B

### Capture d'écran de l'application

UBUBI 0.0.1-Snapshot

username

password

Login

Register

Figure 2: Page de connexion

UBUBI 0.0.1-Snapshot

Back to Login

Email

Username

Password

Confirm Password

First Name

Last Name

Register

0

Figure 3: Page d'enregistrement

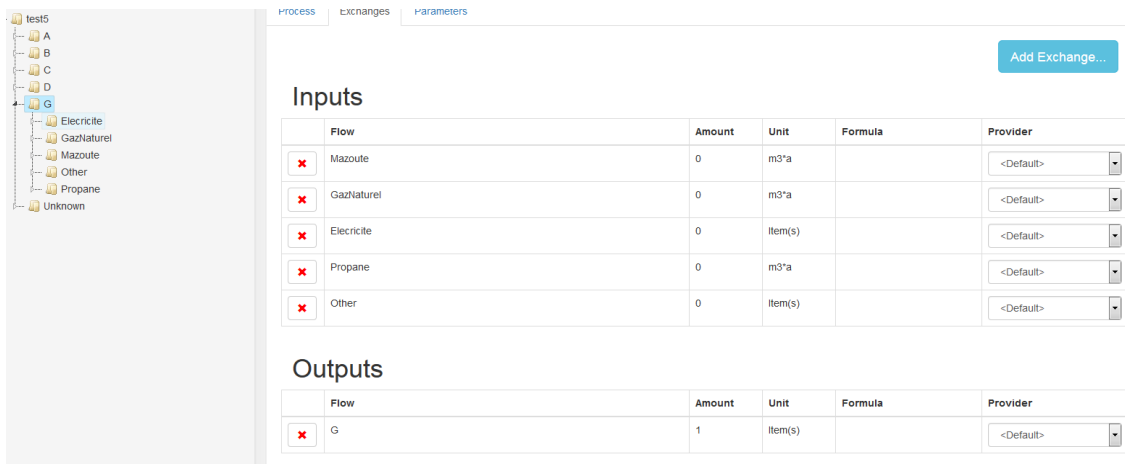


Figure 4: Exemple de création de processus d'énergie

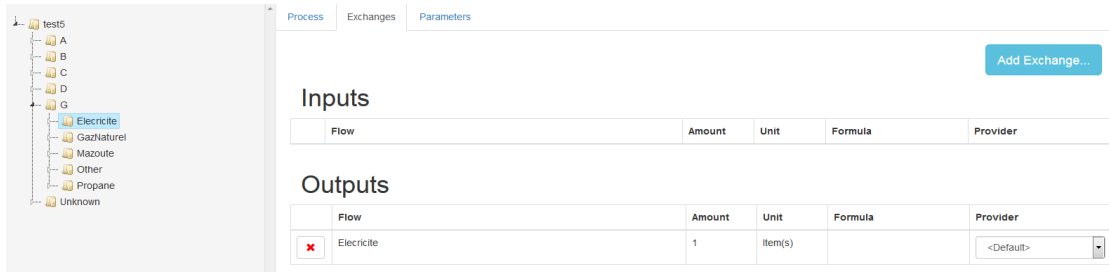
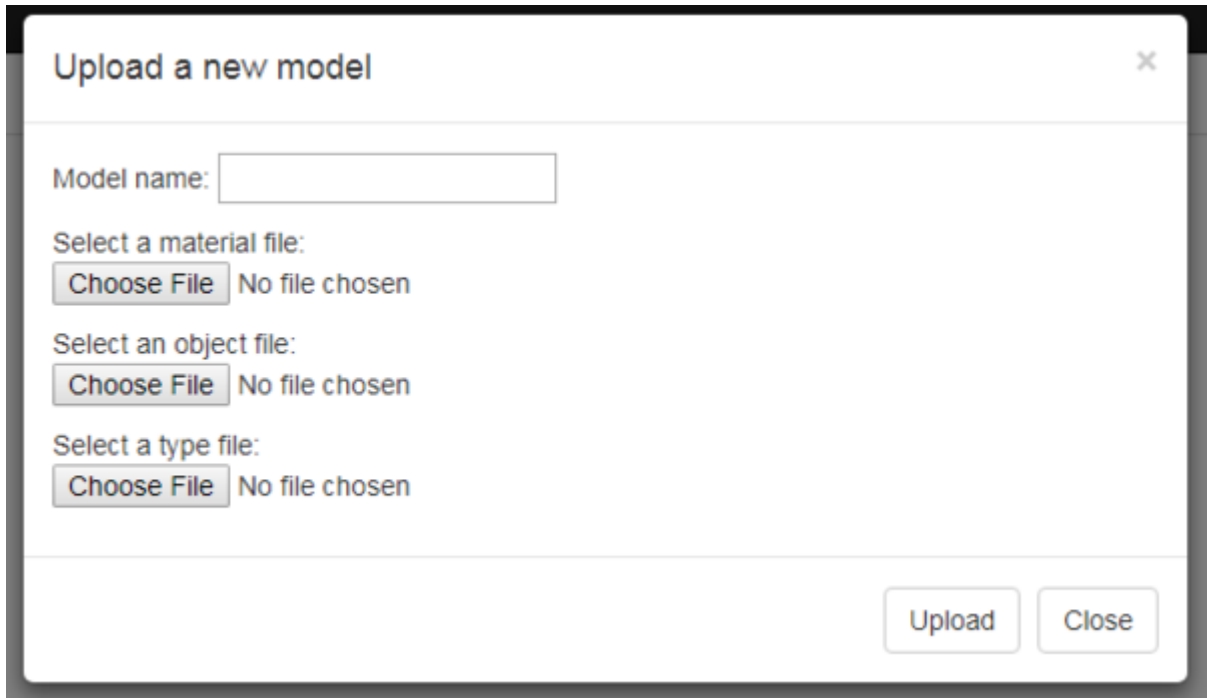


Figure 5: Exemple de sous processus d'énergie

de	niveau	structure
A	1	SUBSTRUCTURE
A10	2	Foundations
A1010	3	Standard Foundations
A1020	3	Special Foundations
A1030	3	Slab on Grade
A20	2	Basement Construction
A2010	3	Basement Excavation
A2020	3	Basement Walls
B	1	SHELL
B10	2	Superstructure
B1010	3	Floor Construction
B1020	3	Roof Construction
B20	2	Exterior Enclosure
B2010	3	Exterior Walls
B2020	3	Exterior Windows
B2030	3	Exterior Doors
B30	2	Roofing
B3010	3	Roof Coverings
B3020	3	Roof Openings
C	1	INTERIORS
C10	2	Interior Construction
C1010	3	Partitions
C1020	3	Interior Doors
C1030	3	Fittings

Figure 6: Insertion des codes unifomats



The image shows a dialog box titled "Upload a new model" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Model name:** A text input field.
- Select a material file:** A "Choose File" button followed by the text "No file chosen".
- Select an object file:** A "Choose File" button followed by the text "No file chosen".
- Select a type file:** A "Choose File" button followed by the text "No file chosen".

At the bottom right of the dialog, there are two buttons: "Upload" and "Close".

Figure 7: Téléchargement d'un nouveau modèle vers le serveur

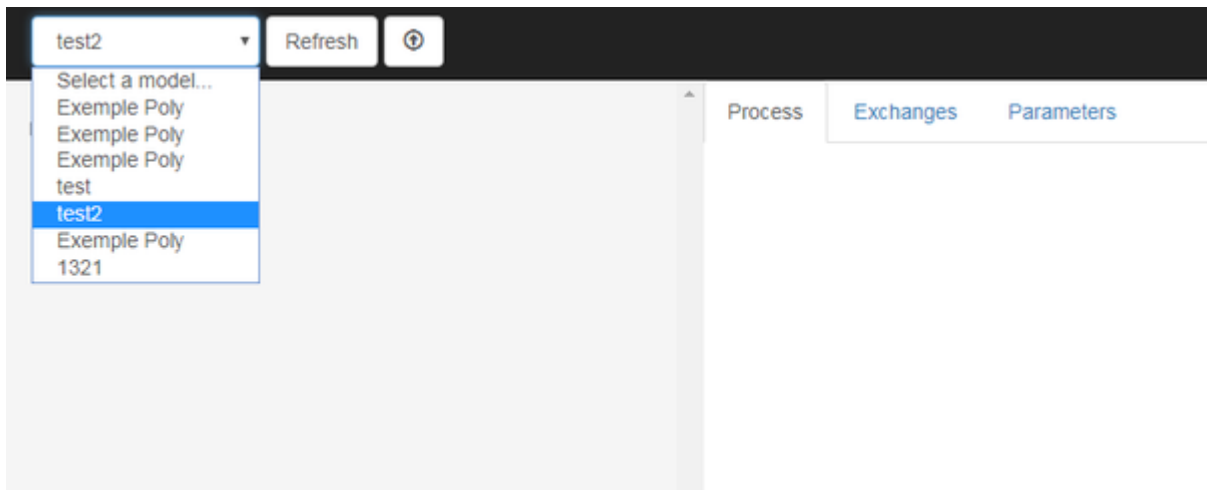


Figure 8: Liste des modèles après upload