

Activité d'intégration
UE 307
Application entreprise multi-tiers

Bachelier en Informatique de gestion Mons

PAE

Groupe n°B3

**Charlier Guillaume - Dhaeyer Sasha – Trempont Mathieu -
Van Bossuyt Nicolas**

2020 - 2021

Remerciements

Nous tenons d'abord remercier la directrice de notre section économique Mme DUROISIN pour avoir joué le rôle de product owner durant cette semaine. Elle a réussi à nous conseiller sur la voie que nous devons prendre pour mener à bien ce projet. Elle nous a aussi donné son avis sur les interfaces pour que nous puissions au mieux faire une application agréable et intuitive pour chaque utilisateur.

Ensuite, nous remercions chaque professeur de notre section pour l'enseignement qu'ils nous ont apporté. Chaque méthode de travail différente soit elle, nous a permis d'acquérir de l'expérience, de la maturité et une conduite de travail rigoureuse qui nous a formés afin de pouvoir réaliser des projets de plus grande envergure.

Tables des matières

Remerciements	1
Tables des matières	2
Introduction	3
Contexte	3
Objectifs pédagogiques	3
Description générale	3
Analyse	5
Product backlog	5
Diagrammes de classe	6
Modèle	6
EJB	8
Contrôleur	8
Site map diagram	9
VTE	10
Description de la technologie	10
Domaine d'application, avantages, inconvénients	10
Technologies alternatives ou concurrentes	11
Code source contextualisé et commenté représentatif de l'intégration dans un projet	11
Extraits de code Commenté	14
Implémentation	17
Travail de groupe	23
Conclusion	24
Bibliographie	25
Fiche technique JEE	26

Introduction

Contexte

Dans le cadre du cours d'Application entreprise multi-tiers, nous avons dû créer des groupes de 4 étudiants maximum et réaliser pendant une semaine une application web dynamique implémentant les fonctionnalités principales définies par le product owner qui est dans notre cas la directrice de la section économique de la HelHa Mons.

Cependant, pour gérer les prototypes des PAE des étudiants, nous devons utiliser JAKARTA pour notre projet. Il nous était aussi imposé que nos noms de classes et méthodes aient suffisamment de commentaires pour simplifier la compréhension du programme.

Deux scénarios nous ont été proposés pour cette application. Un scénario minimal sans pré-encodage de la validation des UE et AA. Cela signifie qu'aucun résultat n'est importé depuis le fichier excel, l'utilisateur encode les réussites/échecs "en direct". L'autre scénario comporte, cette fois-ci, une option de pré-encodage des résultats des étudiants. Ces résultats sont chargés depuis un fichier excel, l'utilisateur doit seulement apporter les modifications éventuelles pendant la délibération.

Ce fichier EXCEL contient les informations de l'étudiant ainsi que la liste des UE avec les AA de sa section.

Objectifs pédagogiques

Ce hackathon a pour but d'améliorer nos compétences en programmation ainsi que d'approfondir nos connaissances en Java Entreprise, ainsi que dans d'autres langages. Ainsi, nous avons pu apprendre React durant cette semaine, que nous n'avions pas vu avant et qui nous a permis de créer le frontend de notre application.

Description générale

L'application se compose essentiellement en 4 parties.

La première partie comprend la connexion à la base de données. Ainsi, l'utilisateur pourra se connecter avec un compte existant afin de pouvoir commencer à utiliser l'application.

La deuxième partie est celle de la gestion des utilisateurs.

En effet, une fois connecté, il sera possible, en allant dans les options d'administrateur, de

1. pouvoir consulter la liste des utilisateurs inscrits en base de données
2. créer un nouvel utilisateur
3. modifier un utilisateur en le sélectionnant dans la liste affichée
4. le supprimer de la base de données une fois un utilisateur sélectionné

La troisième partie comporte l'importation.

Une fois arrivé sur la page d'accueil, l'utilisateur pourra importer son fichier comprenant la liste des étudiants, leur section, leur bloc, leurs points, ... Pour ce faire, il suffit de glisser le

fichier dans la partie prévue à cet effet, de cliquer dans le champ afin d'ouvrir l'explorateur de fichier. Quand un fichier est sélectionné, l'utilisateur pourra voir si le fichier est au bon format (dans le cas contraire, une erreur s'affiche), si il s'est correctement importé et si il s'est bien encodé dans la base de données.

Une fois le fichier importé, il sera redirigé vers la page d'accueil ou il verra les différents blocs et sections.

La quatrième partie est celle concernant les étudiants.

En sélectionnant un bloc provenant d'une section, la liste des étudiants de ce bloc s'affiche, avec sa progression en crédits par rapport à chaque bloc. Sélectionner un étudiant de cette liste amène l'utilisateur à la page concernant cet étudiant avec les blocs validés qui sont déjà cochés, et ceux qui pourront être ajoutés à son pae. En cochant un cours non validé, celui-ci sera ajouté à son PAE. Une fois l'utilisateur satisfait du PAE de l'étudiant, il pourra cliquer sur le bouton "Imprimer", ce qui créera un fichier pdf du PAE de cet étudiant.

Aussi, l'utilisateur pourra décider d'imprimer la liste des PAE de chaque étudiant d'un bloc s'il le désire.

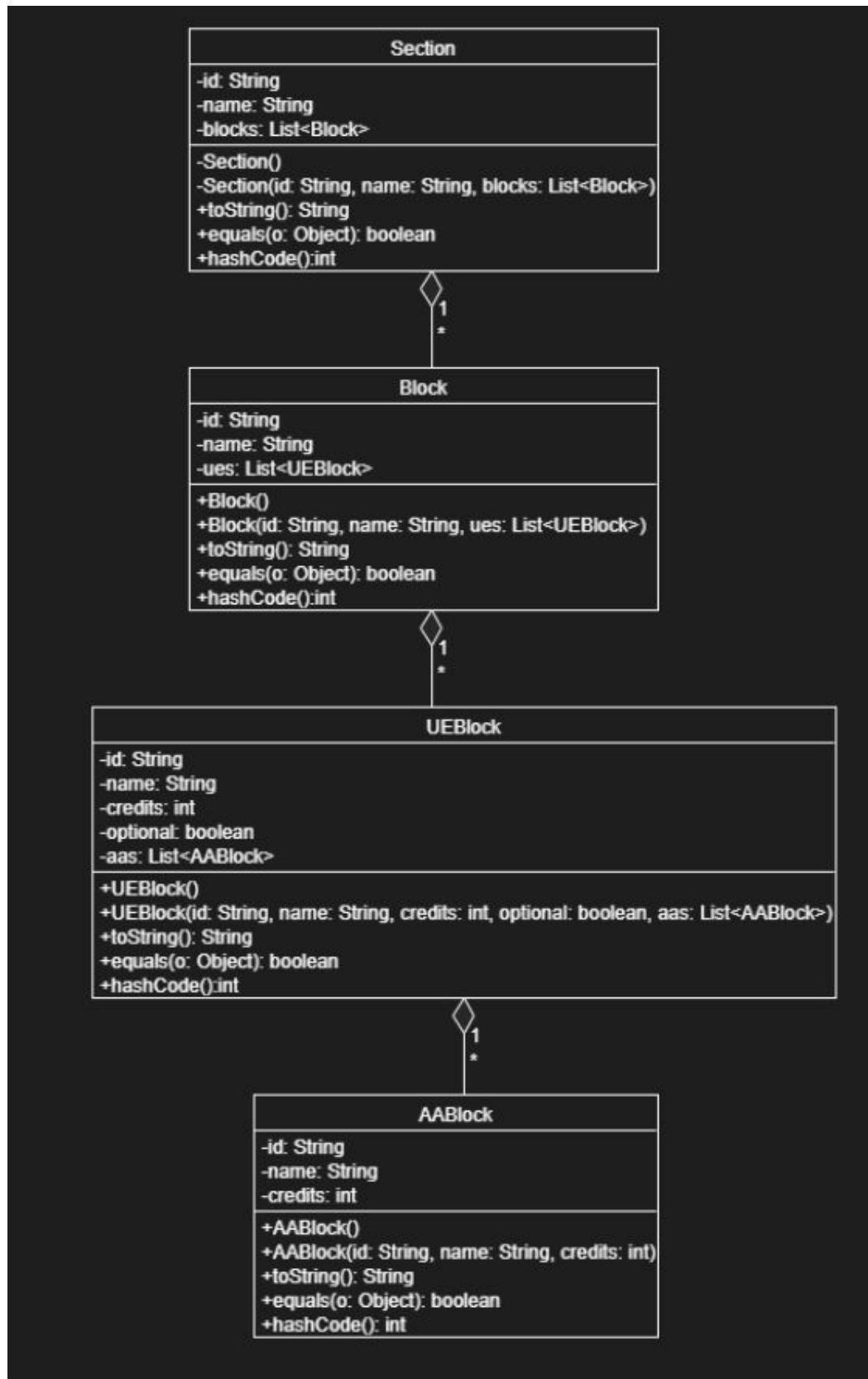
Analyse

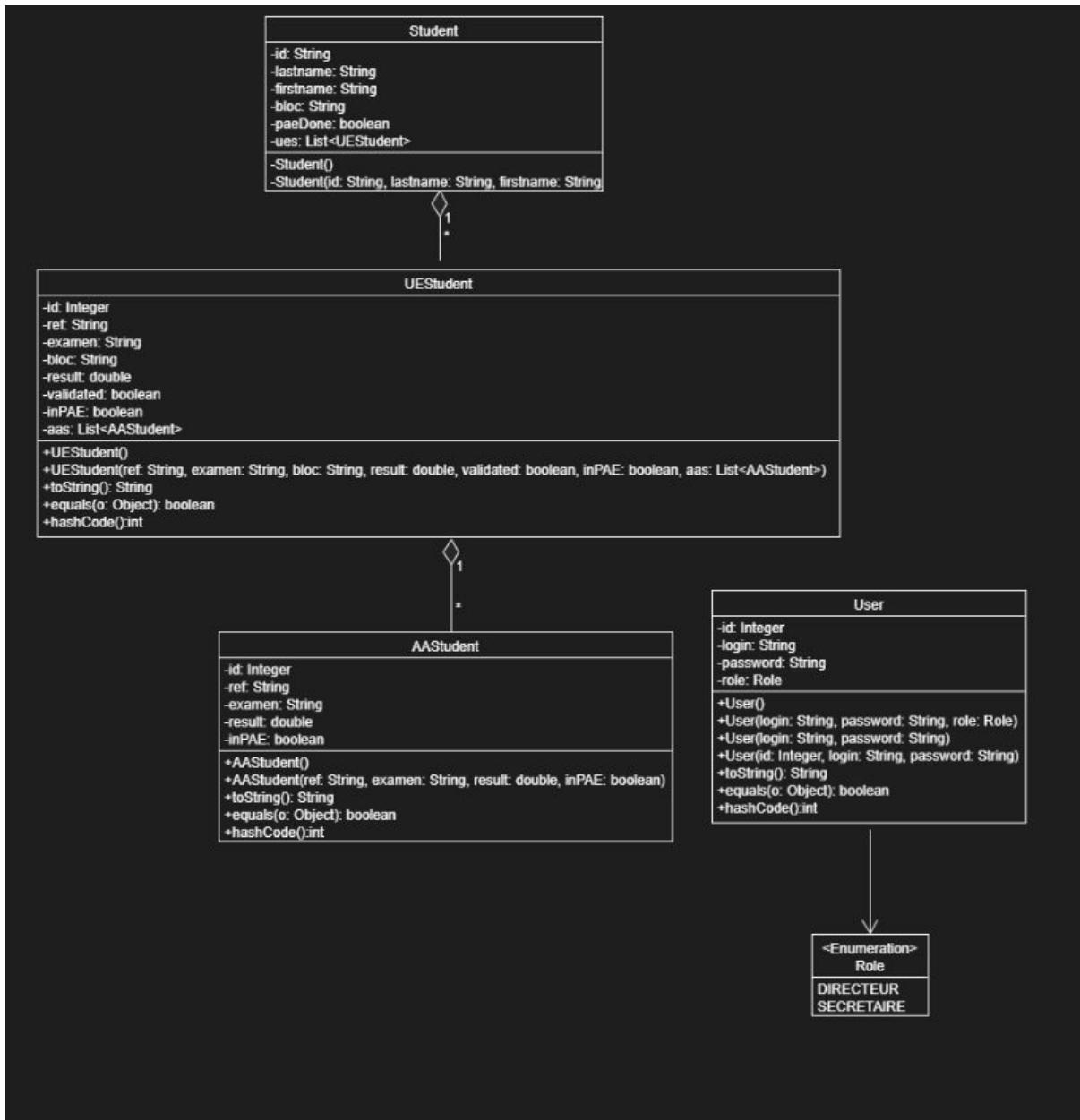
Product backlog

US-01	En tant qu'utilisateur je peux charger la liste des étudiants d'un bloc d'une section à partir d'un fichier excel pour les intégrer au logiciel.
US-02	En tant qu'utilisateur je peux charger la liste des ue et aa d'une section à partir d'un fichier excel pour les intégrer au logiciel.
US-03	En tant qu'utilisateur, je peux afficher la liste des étudiants d'une section par année et cliquer un le nom d'un étudiant pour éditer sa fiche.
US-04	En tant qu'utilisateur je peux afficher la fiche d'un étudiant qui contient les informations d'identification des étudiants ainsi que la liste des cours.
US-05	En tant qu'utilisateur je peux cocher sur la fiche de l'étudiant les AA réussies ainsi que les AA qui feront partie de son PAE.
US-06	En tant qu'utilisateur, je peux générer la liste des PAE des étudiants au format pdf à partir des fiches des étudiants.

Diagrammes de classe

Modèle





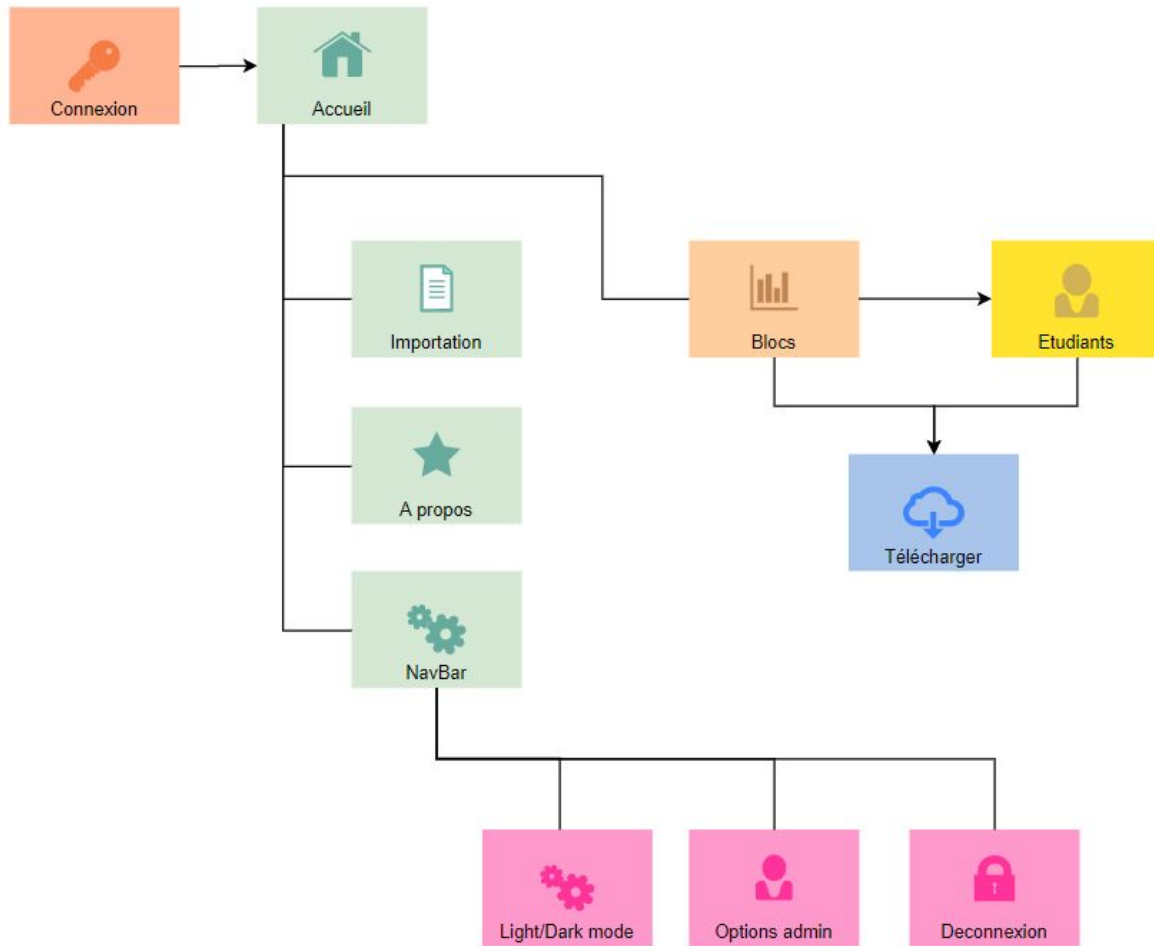
EJB

GenericEJB
-dao:GenericDAO +findAll(klass: Class<TEntity>): <TEntity> TEntity +findById(klass: Class<TEntity>, id: Object): <TEntity> TEntity +findAllByField(klass: Class<TEntity>, field: String, value: Object): <TEntity>List<TEntity> +findFirstById(klass: Class<TEntity>, field: String, value: Object): <TEntity> TEntity +add(klass: Class<TEntity>, entity: TEntity): <TEntity extends IEntity> TEntity +addFastAndUnsafe(klass: Class<TEntity>, entity: TEntity): <TEntity extends IEntity> TEntity +update(klass: Class<TEntity>, oldEntity: TEntity, newEntity: TEntity): <TEntity extends IEntity> boolean +update(klass: Class<TEntity>, entity: TEntity): <TEntity extends IEntity> boolean +remove(klass: Class<TEntity>, entity: TEntity): <TEntity extends IEntity> boolean

Contrôleur

<table><tr><th>AuthController</th></tr><tr><td>-context: UriInfo</td></tr><tr><td>-ejb: GenericEJB</td></tr><tr><td>+AuthController()</td></tr><tr><td>+authentication(login: String, password:String): String</td></tr></table>	AuthController	-context: UriInfo	-ejb: GenericEJB	+AuthController()	+authentication(login: String, password:String): String	<table><tr><th>SectionController</th></tr><tr><td>-ejb: GenericEJB</td></tr><tr><td>-authService: AuthService</td></tr><tr><td>+importSections(token:String, newSections: String): String</td></tr><tr><td>+sendSections(token: String): String</td></tr></table>	SectionController	-ejb: GenericEJB	-authService: AuthService	+importSections(token:String, newSections: String): String	+sendSections(token: String): String			
AuthController														
-context: UriInfo														
-ejb: GenericEJB														
+AuthController()														
+authentication(login: String, password:String): String														
SectionController														
-ejb: GenericEJB														
-authService: AuthService														
+importSections(token:String, newSections: String): String														
+sendSections(token: String): String														
<table><tr><th>StudentController</th></tr><tr><td>-ejb: GenericEJB</td></tr><tr><td>-authService: AuthService</td></tr><tr><td>+importStudents(token: String, newStudents: String): String</td></tr><tr><td>+sendStudents(token: String): String</td></tr></table>	StudentController	-ejb: GenericEJB	-authService: AuthService	+importStudents(token: String, newStudents: String): String	+sendStudents(token: String): String	<table><tr><th>UserController</th></tr><tr><td>-context: UriContext</td></tr><tr><td>-ejb: GenericEJB</td></tr><tr><td>-authService: AuthService</td></tr><tr><td>+showAll(token: String):String</td></tr><tr><td>+create(token: String, login: String, password: String): String</td></tr><tr><td>+update(token: String, id: int, login: String, password: String): String</td></tr><tr><td>+delete(token: String, id: int): String</td></tr></table>	UserController	-context: UriContext	-ejb: GenericEJB	-authService: AuthService	+showAll(token: String):String	+create(token: String, login: String, password: String): String	+update(token: String, id: int, login: String, password: String): String	+delete(token: String, id: int): String
StudentController														
-ejb: GenericEJB														
-authService: AuthService														
+importStudents(token: String, newStudents: String): String														
+sendStudents(token: String): String														
UserController														
-context: UriContext														
-ejb: GenericEJB														
-authService: AuthService														
+showAll(token: String):String														
+create(token: String, login: String, password: String): String														
+update(token: String, id: int, login: String, password: String): String														
+delete(token: String, id: int): String														

Site map diagram



VTE

Description de la technologie

React est une bibliothèque JavaScript développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page HTML à chaque changement d'état.

Domaine d'application, avantages, inconvénients

React.js est parfait pour développer des applications web et mobiles. Il est rapide, sécurisé et scalable. Il offre à la fois une excellente expérience utilisateur et une fabuleuse expérience développeur. Sa popularité s'accroît grâce au soutien de Facebook et d'une communauté dynamique.

Avantages :

- Composants réutilisables : l'un des avantages majeurs de React, c'est de pouvoir facilement créer des composant réutilisables
- Le JavaScript est plus simple à écrire : ReactJS utilise une syntaxe spéciale appelée JSX, qui permet de mixer l'HTML et le Javascript, qui est très pratique car elle permet de créer des composants très facilement.
- L'intelligibilité : ReactJS produit du code « propre » (simple à lire), sa lecture permet de déterminer immédiatement quelles sont les fonctionnalités de votre application
- La rapidité : La rapidité du rendu fait partie des fonctionnalités-phares de React et lui donne un avantage certain sur Angular. Cette technologie embarque des méthodes de réduction du nombre d'opérations sur le DOM, ainsi que d'optimisation et d'accélération du processus de mise à jour. Le DOM (Document Object Model) est d'une grande utilité en cas de gestion de grosses bases de données.

Inconvénients :

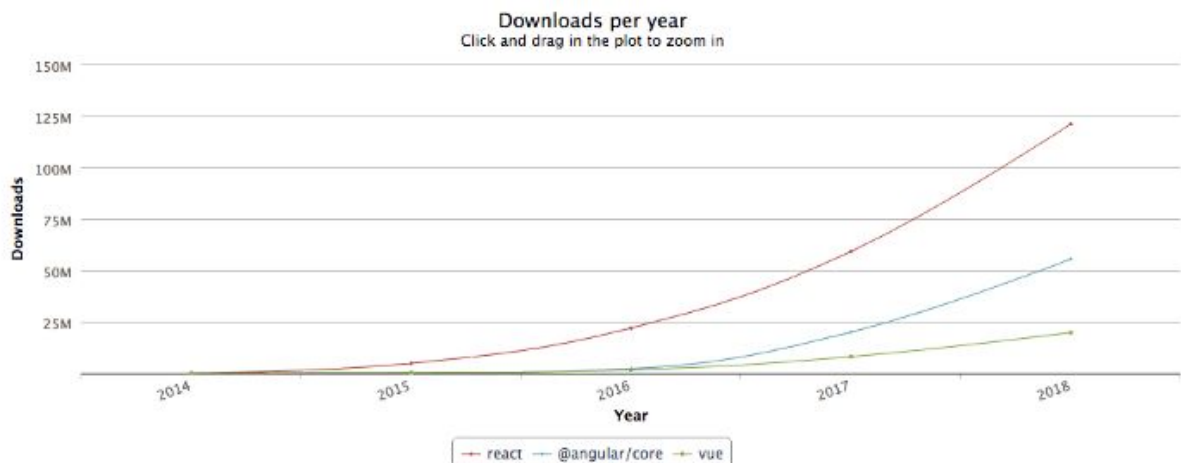
- Connaissances approfondies nécessaires : développer une application mobile sur ReactJS demande de très fortes compétences en développement. La documentation officielle reste plutôt légère. De plus, le langage JavaScript est un langage de programmation qui ne garantit pas les erreurs possibles de programmation. Il faut donc accorder une importance pointue lors de la phase de développement.
- Manque de documentation officielle : le développement ultrarapide de ReactJS ne laisse aucune place à la documentation appropriée

Technologies alternatives ou concurrentes

Il existe différentes technologies alternatives à React : Preact, Inferno JS, Backbone JS, Ember JS mais les plus connues sont AngularJS et VueJS.

AngularJS est très utilisé pour la construction d'applications web hautement interactives et VueJS convient parfaitement à la création d'interfaces utilisateur hautement adaptables et d'applications Single-page complexes.

Chacune de ces technologies a ses avantages et inconvénients, cependant, au cours des dernières années, le nombre des téléchargements de packages React.js avec NPM par rapport aux deux autres frameworks populaires que sont Angular/Core et Vue s'est considérablement accru.



Code source contextualisé et commenté représentatif de l'intégration dans un projet

```

1  import Header from "../components/Header";
2  import { mdiEmail, mdiLaptop, mdiPrinter } from "@mdi/js";
3  import { Link, useParams } from "react-router-dom";
4  import Button from "../components/Button";
5  import { FindStudentsByBloc } from "../services/StudentsService";
6  import {
7    FindBlockById,
8    FindSectionFromBlocId,
9  } from "../services/SectionService";
10
11 import Loading from "../components/Loading";
12
13 import { useEffect, useState } from "react";
14 import { StudentHasValidatedUE } from "../model/Student";
15
16 // Charge the students' data into a table
17 // Configuration of datas
18 export function Progress(props) {
19   return (
20     <div className="h-3 relative rounded-full overflow-hidden">
21       <div className="w-full h-full bg-gray-200 absolute"></div>
22       <div
23         style={{ width: props.value + "%" }}
24         className="h-full bg-helha_blue relative"
25       ></div>
26     </div>
27   );
28 }
29

```

```

30 // Sum of the credits to display it properly later
31 function SumBloc(student, ues, blocs, numBloc) {
32     let somme = 0;
33     let ueBloc = [];
34
35     // Configure the ues to correspond with the ues' bloc
36     ues.forEach((ue) => {
37         if (ue.bloc === numBloc) {
38             ueBloc.push(ue);
39         }
40     });
41
42     // Verify if the bloc is validated
43     for (let i = 0; i < ueBloc.length; i++) {
44         if (StudentHasValidatedUE(student, ueBloc[i].ref)) {
45             somme += blocs.ues[i].credits;
46         }
47     }
48     return (somme / 6) * 10;
49 }
50
51 // Display a student in the page
52 export function Student(props) {
53     let student = props.student;
54     let blocs = props.blocs;
55
56     console.log(student);
57
58     return (
59         <tr>
60             { /* Student name */}
61             <td>
62                 <Link to={"/student/" + student.id}>
63                     {student.lastname} {student.firstname}
64                 </Link>
65             </td>
66
67             { /* Progress bar per bloc */}
68
69             { /* Bloc 1 */}
70             <td>
71                 <Progress
72                     value={SumBloc(student, student.ues, blocs[0], blocs[0].id)}
73                 />
74             </td>
75
76             { /* Bloc 2 */}
77             <td>
78                 <Progress
79                     value={SumBloc(student, student.ues, blocs[1], blocs[1].id)}
80                 />
81             </td>
82
83             { /* Bloc 3 */}
84             <td>
85                 <Progress
86                     value={SumBloc(student, student.ues, blocs[2], blocs[2].id)}
87                 />
88             </td>
89
90             { /* Check box, checked if the pae is done */}
91             <td>
92                 <center>
93                     <input type="checkbox" />
94                 </center>
95             </td>

```



```

96     </tr>
97   );
98 }
99
100 // Display all student in the page
101 export default function Bloc() {
102   // Find the bloc id
103   let { blocId } = useParams();
104
105   let [state, setState] = useState(undefined);
106
107   // Find the datas' in the database
108   useEffect(() => {
109     if (state !== undefined) {
110       return;
111     }
112
113     Promise.all([
114       FindBlockById(blocId),
115       FindSectionFromBlocId(blocId),
116       FindStudentsByBloc(blocId),
117     ]).then((data) => {
118       setState({ bloc: data[0], section: data[1], students: data[2] });
119     });
120   });
121
122   if (state === undefined) {
123     return <Loading />;
124   }
125
126   return (
127     <div className="bg-gray-100 dark:bg-helha_dark_grey flex-1">
128       { /* Display the header */ }
129       <Header
130         icon={mdiLaptop}
131         title={state.bloc.name}
132         description={state.section.name}
133       >
134         <Button text="Imprimer" icon={mdiPrinter} />
135         <Button text="Envoyer" icon={mdiEmail} />
136       </Header>
137
138       { /* Display all students */ }
139
140       <div className="shadow rounded p-4 my-8 bg-white dark:bg-helha_grey max-w-4xl mx-auto flex flex-col gap-2">
141         <table>
142           <thead>
143             <tr className="border-b-2">
144               <th>Nom et prénom</th>
145               <th className="flex-1">Bloc 1</th>
146               <th className="flex-1">Bloc 2</th>
147               <th className="flex-1">Bloc 3</th>
148               <th>Fait?</th>
149             </tr>
150           </thead>
151           <tbody>
152             { /* Search all students */ }
153             {state.students.map((student, index) => (
154               <Student
155                 key={student.id}
156                 student={student}
157                 blocs={state.section.blocs}
158               />
159             ))}
160           </tbody>
161         </table>
162       </div>
163     </div>
164   );
165 }
166
167 }
168

```

Extraits de code Commenté

DAO Générique

```

1 package be.helha.groupeb3.storage;
2
3 import java.util.List;
4
5
6
7
8
9
10 @Stateless
11 public class GenericDAO {
12
13     @PersistenceContext(unitName = "groupeB3JTA")
14     private EntityManager manager;
15
16     /**
17      * Find all entities
18      * @param <TEntity> : type of the data
19      * @param klass : name of the variable, which represent the class used
20      * @return : a list of entities (of type TEntity)
21      */
22     public <TEntity> List<TEntity> findAll(Class<TEntity> klass) {
23         var queryString = "select entity from " + klass.getName() + " entity";
24
25         return manager.createQuery(queryString).getResultList();
26     }
27
28     /**
29      * Return an entity by the id
30      * @param <TEntity> : type of the data
31      * @param klass : name of the variable, which represent the class used
32      * @param id : id of the object
33      * @return : an entity (of type TEntity)
34      */
35     public <TEntity> TEntity findById(Class<TEntity> klass, Object id) {
36         return manager.find(klass, id);
37     }
38

```

```
/**
 * Return entities compared to the field passed in parameter
 * @param <TEntity> : type of the data
 * @param klass : name of the variable, which represent the class used
 * @param field : name of the field in the database
 * @param value : value of the field researched
 * @return : a list of entities (of type TEntity)
 */
public <TEntity> List<TEntity> findAllByField(Class<TEntity> klass, String field, Object value) {
    String request = "select entity from " + klass.getName() + " entity where entity." + field + "=?1";

    TypedQuery<TEntity> query = manager.createQuery(request, klass);
    query.setParameter(1, value);

    return query.getResultList();
}

/**
 * Add an entity
 * @param <TEntity> : type of the data
 * @param klass : name of the variable, which represent the class used
 * @param entity : the entity that'll be added
 * @return : the entity added (of type TEntity)
 */
public <TEntity extends IEntity> TEntity add(Class<TEntity> klass, TEntity entity) {
    if (entity == null) {
        return null;
    }

    if (entity.getId() != null && findById(klass, entity.getId()) != null) {
        return null;
    }

    manager.persist(entity);
    return entity;
}
```

```
/**
 * Add an entity without checking the id
 * @param <TEntity> : type of the data
 * @param klass : name of the variable, which represent the class used
 * @param entity : the entity that'll be added
 * @return : the entity added (of type TEntity)
 */
public <TEntity extends IEntity> TEntity addFastAndUnsafe(Class<TEntity> klass, TEntity entity) {
    if (entity == null) {
        return null;
    }

    manager.persist(entity);

    return entity;
}
```



```
/**
 * Update an entity with the new entity passed in parameter
 * @param <TEntity> : type of the data
 * @param klass : name of the variable, which represent the class used
 * @param oldEntity : the entity that will be replaced by the new one
 * @param newEntity : the new entity that'll replace the old one
 * @return : a boolean (true or false)
 */
public <TEntity extends IEntity> Boolean update(Class<TEntity> klass, TEntity oldEntity, TEntity newEntity) {
    if (oldEntity == null || newEntity == null) {
        return false;
    }

    TEntity entity = findById(klass, oldEntity.getId());
    manager.clear();

    if (entity == null) {
        return false;
    }

    entity = newEntity;
    entity.setId(oldEntity.getId());
    manager.merge(entity);

    return true;
}
```

```
/**
 * Update an entity
 * @param <TEntity> : type of the data
 * @param klass : name of the variable, which represent the class used
 * @param entity : the entity that'll be updated
 * @return : a boolean (true or false)
 */
public <TEntity extends IEntity> Boolean update(Class<TEntity> klass, TEntity entity) {

    TEntity ent= findById(klass, entity.getId());
    manager.clear();

    return update(klass, ent, entity);
}

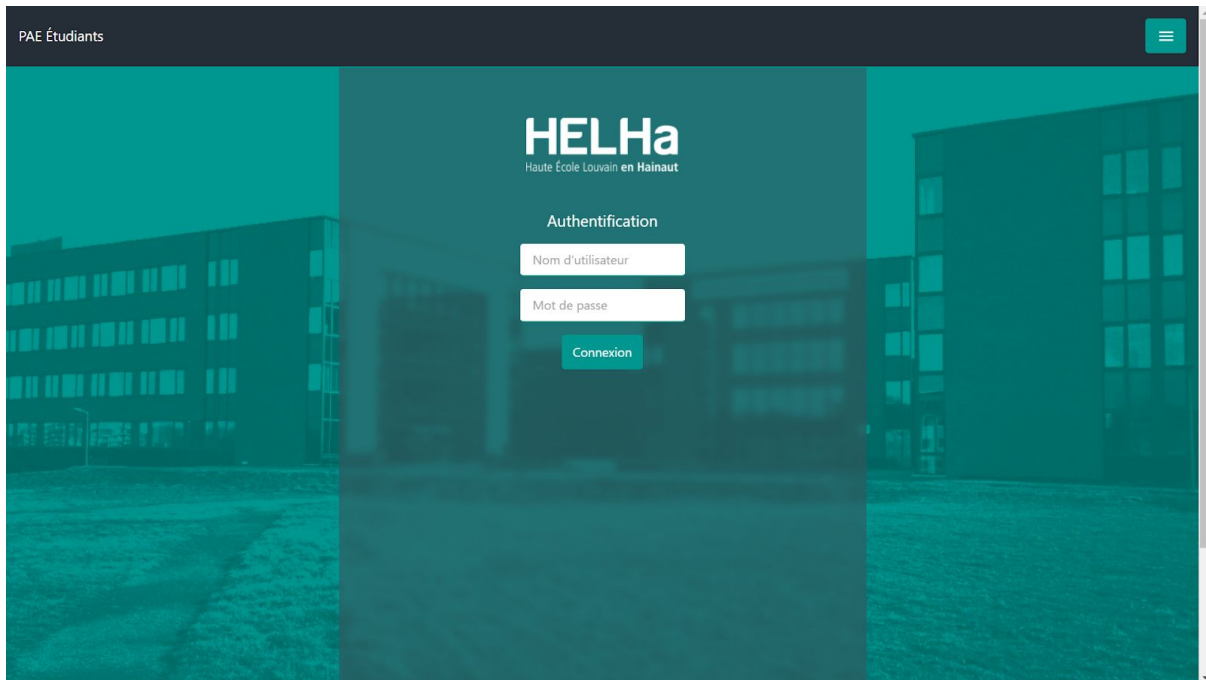
/**
 * Remove an entity
 * @param <TEntity> : type of the data
 * @param klass : name of the variable, which represent the class used
 * @param entity : the entity that'll be updated
 * @return : a boolean [true or false]
 */
public <TEntity extends IEntity> boolean remove(Class<TEntity> klass, TEntity entity) {
    TEntity isIn = findById(klass, entity.getId());

    if (isIn == null) {
        return false;
    }

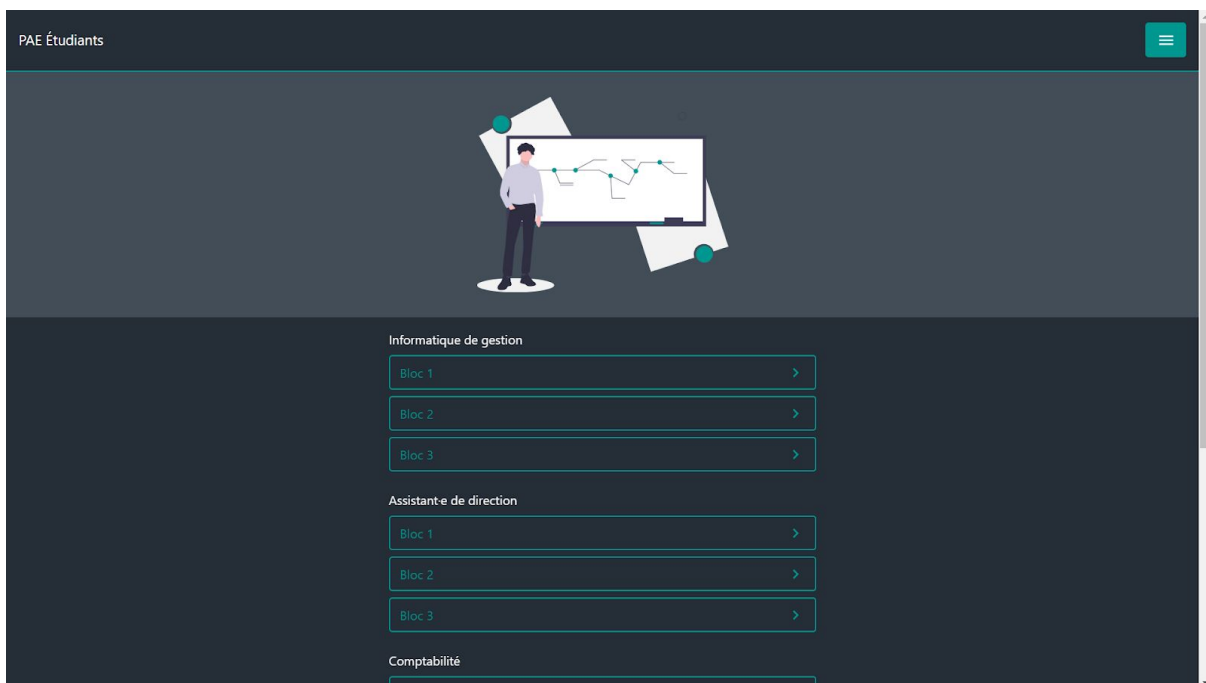
    manager.remove(isIn);
    return true;
}
```

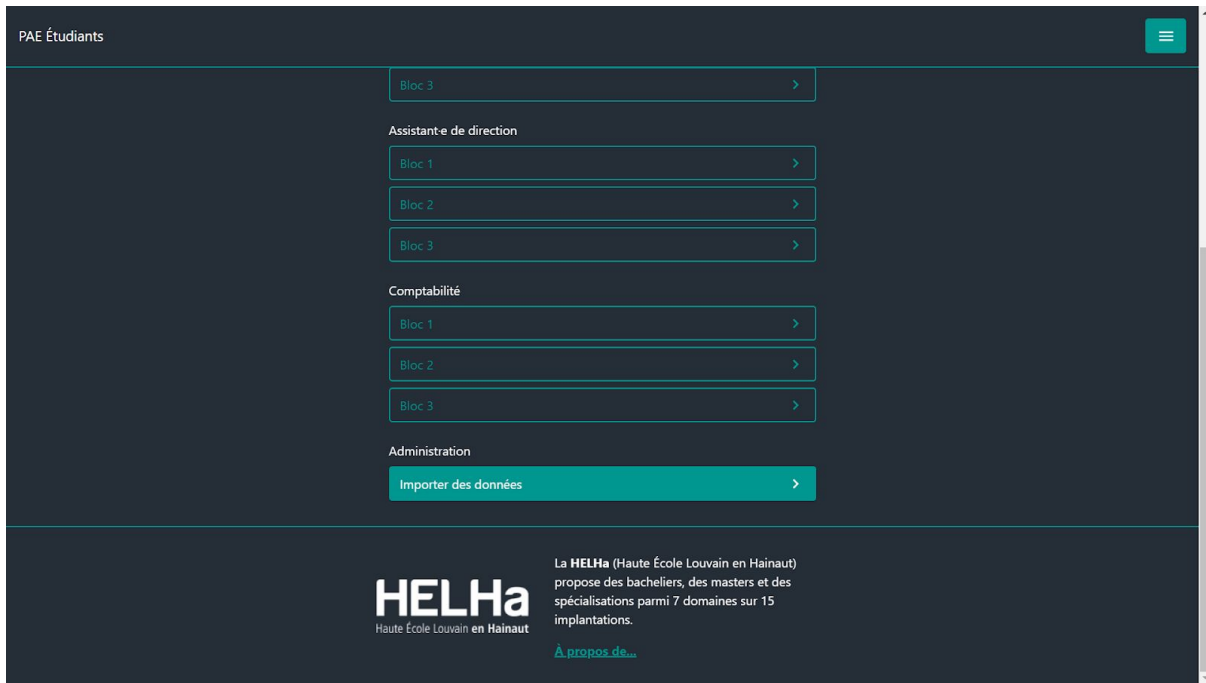
Implémentation

Interface de login

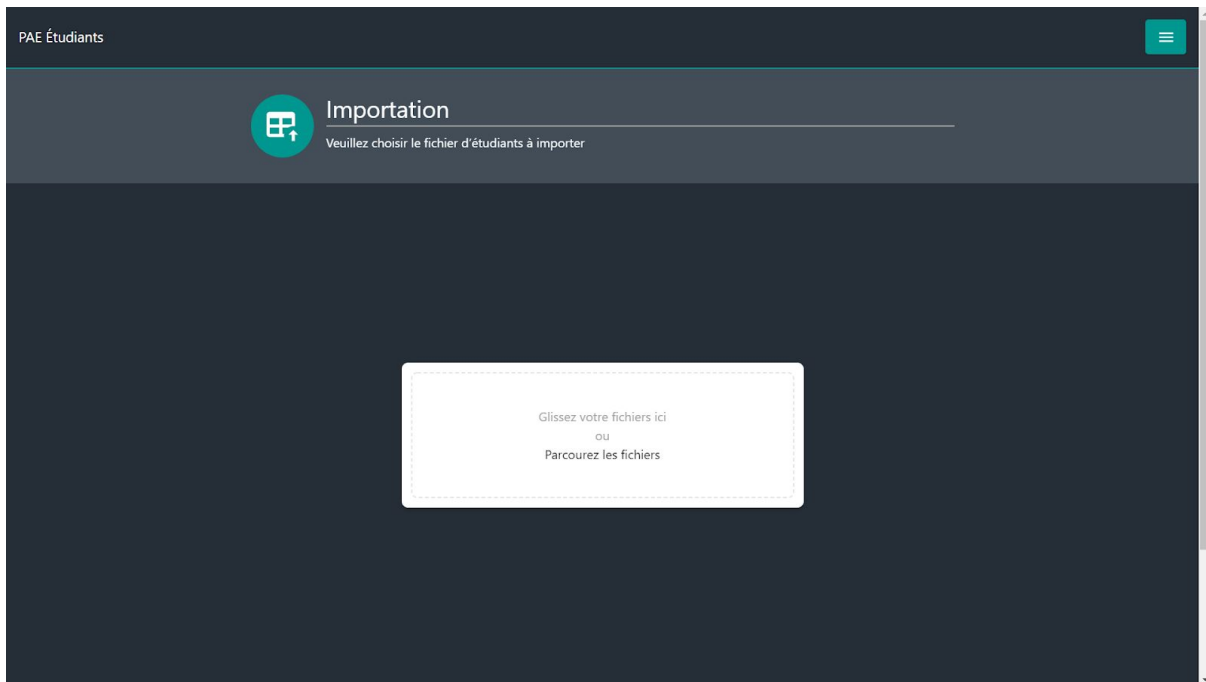


Accueil

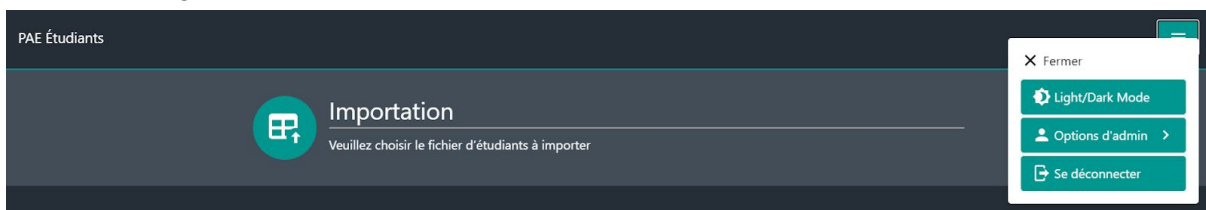




Importation



Barre de navigation et menu



Interface administrateur

PAE Étudiants

Administration
Modification des utilisateurs

[Créer un compte >](#)

Veuillez sélectionner le compte à modifier

Id	Login	Rôle	
1	user	SECRETAIRE	>
2	admin	DIRECTEUR	>

Modification d'un utilisateur et suppression

PAE Étudiants

Compte de l'utilisateur user
Modifiez les informations ou supprimez le compte

[Mettre à jour les informations](#) [Supprimer](#)

Nom d'utilisateur :

user

Mot de passe :

Mot de passe

Interface de création d'utilisateur

PAE Étudiants

HELHa
Haute École Louvain en Hainaut

Enregistrement d'un/e secrétaire

Nom d'utilisateur

Mot de passe

☐ Afficher le mot de passe

Créer un compte

[Revenir à la page de gestion d'utilisateurs](#)

Affichage des étudiants par bloc

PAE Étudiants


Bloc 1
Informatique de gestion

Télécharger tous les PAE

Étudiants Informatique de gestion	Progression			
	Bloc 1	Bloc 2	Bloc 3	
Belepuwe Vewehe	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	✓
Beyafuyo Dupoxu	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	✓
Beyixja Zukusi	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	○
Bolanuni Savaji	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	○
Boxunudi Vizobi	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	○
Buxacora Qemume	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	○
Cajudapo Roxido	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	○
Calafewu Gefosi	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	○
Cuwacola Pofozu	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	○
Dedavihe Jacoto	<div><div></div></div>	<div><div></div></div>	<div><div></div></div>	○

Affichage d'un étudiant + création de son PAE

PAE Étudiants



Veweho Belepuwe
Bachelier en informatique de gestion - IG1

Prévisualiser

Télécharger

BLOC 1

5 Crédits


48 Crédits - Validée

↕ Développement 1	4 Crédits	
↕ Pr. lang. prog 2	12 Crédits	<input checked="" type="checkbox"/>
↕ Gestion 1	6 Crédits	
↕ Développement 2	5 Crédits	
↕ Tech. Internet 1	5 Crédits	
↕ Fichiers et B.D.	2 Crédits	
↕ Langues étr. 1	4 Crédits	
↕ Mat. et t. de d.	6 Crédits	

Confirmer

5 Crédits

PAE Étudiants



Veweho Belepuwe
Bachelier en informatique de gestion - IG1

Prévisualiser

Télécharger

BLOC 3

22 Crédits

9 Crédits - Validée


↕ Prat. de la Cpt. 2	5 Crédits	<input checked="" type="checkbox"/>
↕ Fis. eu. lg. étr.	3 Crédits	
↕ Dim. hum. manag.	1 Crédit	
<div> <div>×</div> <div>Spé. lg. des aff.</div> <div>LF: anglais</div> <div>Projet ling.: anglais</div> <div>LF</div> <div>Projet ling.: néerlandais</div> </div>	7 Crédits	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
↕ Spéc. Fiscalité 2	5 Crédits	
↕ Spé. en gest. 3	3 Crédits	<input checked="" type="checkbox"/>
↕ Act. d'int. prof.	15 Crédits	<input type="checkbox"/>
↕ Compta. 3	2 Crédits	<input type="checkbox"/>
↕ Log. comptables	6 Crédits	

Confirmer

22 Crédits

Interface de prévisualisation d'impression

PAE Étudiants




Veweho Belepuwe

Bachelier en informatique de gestion - IG1

Télécharger

UE - IF108 : Pr. lang. prog 2	12
INF12 - Méthodes et pr. de prog.	2
INF13 - Lang. de prog. procédural	5
INF14 - Projet	5
UE - IF202 : Langues étr. 2	3
INF20 - Langues étrangères	3
UE - IF204 : Tech. Internet 2	6
INF22 - Technologies Internet	3
INF23 - Fichiers et B.D.	3




La **HELHa** (Haute École Louvain en Hainaut) propose des bacheliers, des masters et des spécialisations parmi 7 domaines sur 15 implantations.

[À propos de...](#)

Interface "À propos"


PAE Étudiants




À propos

v1.0.0 publiée le 18/01/2021 code-name 'pinpin'.


Application réalisé avec ❤️ par




Sasha Dhaeyer



Mathieu Trempont
[mathieu-trempont-a6b60117b](#)



Guillaume Charlier
[charlier-guillaume-b750b21ab](#)



Nicolas Van Bossuyt
[nicolas-van-bossuyt](#)

Étudiants en informatique de gestion à la HELHa Campus Mons, dans le cadre du cours d'Applications entreprise multi-tiers de Mr. V. Altares.

Stats pour les nerds 🤖

React v17.0.1
TailwindCSS v2.0.0

Travail de groupe

Afin de travailler plus efficacement, nous avons mis en place une série de tâches sur GitHub. Ainsi, chaque membre de groupe a pu choisir la partie qu'il serait chargé de concevoir. Cela nous a permis de voir plus clair et de mieux savoir quoi faire. Le travail étant bien réparti, nous avons donc pu commencer à coder efficacement.

L'entraide a été essentielle durant ce projet. En effet, pour la plupart nous ne savions pas comment utiliser React. Ainsi, en cas de soucis, nous avons pu compter sur l'aide de chacun d'entre nous pour répondre aux questions, essayer de résoudre une erreur, ...

Aussi, étant donné que pour tous les projets que nous avons dû réaliser dans le cadre des cours de Technologies Internet, Technologies Mobile, Développement de Jeu Vidéo, nous étions dans le même groupe, cela a facilité la compréhension des autres et la cohésion, ce qui a rendu le travail plus "fluide".

Répartition :

6 CHARLIER Guillaume	5 DHAAYER Sasha	5 TREMPONT Mathieu	6 VAN BOSSUYT Nicolas
<ul style="list-style-type: none"> Implementation du token de connection Suppression du menu modal quand on clique en dehors Cryptage des mots de passe des utilisateurs Connexion entre backend et frontend Interface d'importation d'un fichier excel Implémentation de la suppression d'utilisateur 	<ul style="list-style-type: none"> Interface du feedback du PAE Interface des étudiants par bloc Barres de progression de bloc d'un étudiant Conversion du PAE en PDF Génération de tout les PAE sous format PDF 	<ul style="list-style-type: none"> Implémentation des classes entités dans le backend Interface des options d'administrateur Implémentation de la création de compte Implémentation de la modification d'utilisateur Ajout des feedbacks pour l'utilisateur 	<ul style="list-style-type: none"> implémentation de la redirection au login si pas connecté Implementation d'un thème Dark Organisation du fichier Excel après lecture Implémentation des dao et ejb génériques Ajout de la sélection des cours pour le PAE Ajout des crédits des PAE

Conclusion

En conclusion, cette semaine de développement, et plus particulièrement ce projet nous a permis d'approfondir nos connaissances en Java EE tout en améliorant nos compétences en développement. Aussi, il nous a permis d'apprendre de nouveaux langages comme React, qui, j'en suis sûr, nous sera utile à l'avenir. Toutes ces connaissances acquises au cours de ce projet complètent bien notre formation, afin d'être le plus polyvalent possible.

Aussi, il y a vraiment une force dans notre groupe c'est l'entraide, nous aidions toujours, dans la mesure du possible, les membres du groupe qui avait le plus de difficulté. Au début du projet, nous nous sommes répartis les tâches en fonction des facilités et/ou des difficultés de chacun.

Nous sommes fiers et satisfaits du travail accompli, malgré les erreurs pouvant survenir lors du développement et qui pourrait nous bloquer pour la suite.

Bibliographie

Stack Overflow - Where Developers Learn, Share, & Build Careers. (2020). Stack Overflow. <https://stackoverflow.com/>

Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. (2020, 15 novembre). Tailwindcss. <https://tailwindcss.com/>

React – A JavaScript library for building user interfaces. (2013). React. <https://reactjs.org/>

Adam Bien. (2019, 1 octobre). Authentication and Authorization with Payara and MicroProfile JWT. YouTube. <https://www.youtube.com/watch?v=ozQzsVxFfXU&list=RDCMUCksTNgiRyQGwi2ODBie8HdA&index=8>

Devoxx FR. (2017, 24 avril). Mettre en place sa sécurité et sa gestion d'identité en 2017 (Sebastien Blanc). YouTube. <https://www.youtube.com/watch?v=tC7G7cjHGrE>

M. (2020). MrRio/jsPDF. GitHub. <https://github.com/MrRio/jsPDF>

npm : jsPDF. (2021, 15 janvier). npm. <https://www.npmjs.com/package/jsPDF>

Home - Documentation. (2021). githack. <http://raw.githack.com/MrRio/jsPDF/master/docs/>

Fiche technique JEE

	Pas utilisé	Partiellement opérationnel : nom qualifié de la classe ou de la page web	Complètement opérationnel : nom qualifié de la classe
Architecture 3-tiers, 4-tiers			3-tiers
Noms de 2 patterns utilisés:			X
-1			Singleton (services)
-2			React component pattern
Entités			AABlock AASStudent Block Role Section Student UEBlock UEStudent User
Hiérarchie d'entités			X
Associations 1 - 1	X		
Associations 1 – n			Section => Block Block => UEBlock UEBlock => AABlock Student => UEStudent Student => AASStudent
Associations n - n	X		
Requêtes dynamiques			X
Requêtes nommées			X
orphanRemoval			X

JTA			X
Stateless Session bean			GenericDAO en @Stateless
@Remote	X		
@Local	X		
Named beans	X (a voir)		
JAAS	X		
fileRealm	X		
jdbcRealm	X		
WebService - API REST			X (Api Rest)
GET			X
POST			X
DELETE			X
UPDATE			X
CSS			X
AJAX			X
HTML5			X
JavaScript			X
JQuery	X		
JSF, indiquez ci-dessous les	X		

principales balises:			
JUnit	X		