

INFORMATIQUE

Mémoire de fin d'études

en vue de l'obtention du titre de

Bachelier en Développement d'Applications

Année académique 2023 - 2024

My Skills Logbook

HEC-Ecole de Gestion de l'université de Liège,
cellule OEP

Rue Louvrex 14, 4000, Liège

Présenté par

Lucas BODET



Mémoire de fin d'études

en vue de l'obtention du titre de

Bachelier en Développement d'Applications

Année académique 2023 - 2024

My Skills Logbook

HEC-Ecole de Gestion de l'université de Liège,
labo AR/VR SIG

Rue Louvrex 14, 4000, Liège

Présenté par

Lucas BODET



Remerciements

Tout d'abord, je tiens à témoigner toute ma gratitude à mon maître de stage, Stéphane DESPLANQUE ainsi qu'aux membres de l'équipe que j'ai intégrée pour m'avoir aidé et encadré.

Ensuite, je tiens à exprimer toute ma reconnaissance à mon promoteur Jean JADOT et à mes professeurs pour m'avoir accompagné lors de la rédaction de ce mémoire et de mon parcours.

Enfin, je tiens à remercier mes proches pour m'avoir soutenu tout au long de mes études.

Table des matières

Introduction	1
1. Présentation de l'entreprise	2
2. Présentation du TFE	4
2.1. Contexte	4
2.2. Les besoins et raisons qui ont poussé à proposer ce TFE.....	6
2.3. Objectifs du TFE	7
3. Méthodologie.....	9
3.1. Organisation du travail	9
3.2. Outils.....	9
3.3. Réflexions personnelles.....	10
4. Analyse	11
4.1. Fonctionnalités	11
User Stories	12
4.2. Modèle de données.....	14
4.3. Architecture logicielle.....	15
4.4. Technologies.....	16
4.4.1. PHP	16
4.4.2. PHPUnit.....	16
4.4.3. MySQL	17
4.4.4. HTML.....	17
4.4.5. CSS.....	17
4.4.6. Bootstrap	17
4.4.7. Javascript	17
4.4.8. JQuery	17
4.4.9. PDF Make	18
4.4.10. HTML2Canvas	18
4.4.11. DOM to Image.....	18
4.5. Outils de développement	19
4.5.1. Visual studio code	19
4.5.2. PHPMyAdmin	19
4.5.3. MySQL Workbench	19
4.5.4. GIMP	20
5. Réalisation.....	21

5.1.	Création de la structure de la base de données	21
5.2.	Authentification	21
5.3.	Intégration des interfaces	21
5.4.	Gestion de la navigation et de l'historique	22
5.5.	CRUD	23
5.6.	API	23
5.6.1.	Contrôleurs.....	24
5.6.2.	Fichier d'entrée	24
5.6.3.	Redirection vers les contrôleurs.....	25
5.6.4.	Utilisation des contrôleurs	26
5.6.5.	Utilisation des endpoints.....	27
5.6.6.	Sécurisation de l'API.....	28
5.6.7.	Fichier config.php.....	28
5.6.8.	Uniformisation des données étrangères à l'application	29
5.6.9.	Avantages / désavantages de cette pratique.....	30
5.7.	Création d'un store	30
5.8.	Exports	31
5.9.	Mise en production.....	34
5.10.	Compte-rendu.....	34
	Conclusion	35

Introduction

La rédaction de ce travail de fin d'études est basée sur le développement d'une solution informatique sur laquelle j'ai eu l'opportunité de travailler pour HEC Liège. Il s'agit d'une application destinée aux étudiants. Avant sa création, il n'existait aucune solution permettant aux étudiants de stocker et mettre en valeur leurs activités. L'objectif de cette dernière est d'améliorer les services de l'école en offrant aux étudiants une plateforme où ils peuvent consigner toutes leurs réalisations, participations à des activités et séjours à l'étranger. Ainsi, ils pourront valoriser ces expériences lors de leur recherche d'emploi. Elle permettra également à HEC Liège de renforcer son attractivité, et sera destinée à tous les étudiants, indépendamment de leur domaine d'étude. Il s'agit d'une nouvelle solution développée spécifiquement pour répondre à ce besoin.

Auparavant, les étudiants n'avaient aucun moyen de consigner leurs activités durant leur cursus, ce qui les empêchait de valoriser pleinement leur expérience. Les membres de l'équipe pédagogique ont constaté cette lacune et ont donc imaginé cette application afin d'aider les étudiants à garder une trace de leurs activités et à les mettre en valeur lors de leur recherche d'emploi. Les recruteurs recherchent de plus en plus de candidats avec des compétences transversales qui peuvent être développées aussi grâce à des activités extrascolaires. Grâce à cette application, les étudiants pourront mettre en avant leurs compétences et expériences, ce qui les distinguera des autres candidats. En outre, cela contribue également à renforcer la réputation de l'école en permettant aux étudiants de valoriser leur expérience. Cela améliorera la perception des recruteurs quant à la qualité de l'enseignement dispensé à HEC Liège, ce qui attirera davantage d'étudiants et renforcera l'image de l'école. Enfin, cette application répond également au besoin de transparence en regroupant toutes les activités proposées par l'école en un seul endroit. Les étudiants pourront ainsi mieux s'organiser et participer plus facilement aux activités qui les intéressent.

Dans les chapitres suivants, je commencerai par vous présenter l'entreprise et le service dans lequel j'ai travaillé. Ensuite, je donnerai plus de détails sur le sujet de mon TFE ainsi que sur le contexte qui a poussé HEC Liège à proposer un tel projet.

Une fois que vous aurez pris connaissance du contexte, je décrirai plus amplement mon expérience en détaillant la manière dont s'est organisé mon travail au sein de l'équipe mais également les technologies que j'ai utilisées pour réaliser ce projet ainsi que les problèmes que j'ai rencontrés et les solutions que j'y ai apportées.

Je conclurai en dressant le bilan des objectifs atteints ainsi que des améliorations et futures évolutions dont le travail que j'ai fourni pourrait faire l'objet. Pour terminer, je vous partagerai mes réflexions personnelles sur l'expérience que j'ai vécue durant mon insertion professionnelle au sein de HEC Liège.

1.Présentation de l'entreprise

J'ai réalisé mon stage à HEC Liège qui est une école de commerce et de gestion fondée en 1897. Depuis quelques années, elle fait partie de l'Université de Liège et offre une gamme complète de programmes de baccalauréats, masters et doctorats en commerce, économie et gestion.



Figure 1 : Logo de HEC Liège

Le département OEP (Operational Excellence Program) auquel j'ai été intégré dans le cadre de mon stage a pour rôle de développer des solutions informatiques pour le système d'informations de HEC Liège. Il travaille en étroite collaboration avec les autres départements de l'école pour identifier les besoins en matière de technologies de l'information et de communication et pour proposer des solutions efficaces et adaptées. Pour ce faire, l'équipe à laquelle j'ai été intégré est composée de trois développeurs dont un étant le chef de projet, ainsi que d'une analyste des données qui assure la cohérence et la cohésion des données au sein du système d'information de l'école.

Sa mission est de concevoir, développer, tester et mettre en œuvre des applications informatiques pour soutenir les processus de l'école. Elle est chargée de garantir la qualité, la sécurité, la stabilité du système d'information de l'école. L'équipe est également responsable de la maintenance des applications informatiques existantes et de l'optimisation des processus pour améliorer l'efficacité et la rentabilité de l'école. Elle garantit une intégration harmonieuse des solutions informatiques dans les processus de l'école et offre une assistance technique aux utilisateurs.

En somme, son rôle est de contribuer à l'amélioration de la qualité de l'enseignement et de la recherche de HEC Liège en fournissant des solutions informatiques efficaces et adaptées aux besoins de l'école.

Plus précisément, voici quelques exemples d'interactions que peut avoir l'équipe avec les autres services/départements de l'école :

Les enseignants et les chercheurs de l'école peuvent travailler avec l'équipe OEP pour développer des applications informatiques spécifiques à leurs besoins, telles que des outils d'analyse de données, des systèmes de gestion, etc.

Elle peut collaborer avec le département administratif pour développer des applications informatiques pour la gestion des ressources humaines, la gestion financière, la gestion des partenariats, etc.

Enfin, elle peut travailler avec le département de la recherche pour développer des applications informatiques spécifiques pour la gestion des projets de recherche, la gestion des données de recherche, etc.

Dans l'ensemble, l'équipe OEP est un département clé de HEC Liège et collabore avec de nombreux autres départements pour assurer le bon fonctionnement des processus de l'école grâce à l'utilisation efficace des technologies de l'information et de la communication.

2. Présentation du TFE

Dans ce chapitre, je vais plonger plus en profondeur dans la description du contexte de départ et des objectifs du service à développer. Alors que j'ai déjà donné une présentation succincte dans l'introduction, je vais maintenant explorer les différents aspects de manière plus détaillée.

2.1. Contexte

L'application sur laquelle j'ai travaillé est un projet qui a été pensé en amont par HEC Liège mais qui n'existait pas encore. Le but de ce système est d'améliorer les services de l'école en offrant aux étudiants une plateforme leur permettant de garder une trace de toutes les activités auxquelles ils ont participé durant leur cursus.

En effet, l'application permettra aux étudiants de collecter et d'exporter leurs différentes réalisations, participations à des activités, séjours à l'étranger, afin de pouvoir les valoriser lorsqu'ils chercheront un emploi. Car avant l'existence de cette application, les étudiants ne disposaient pas de traces des activités effectuées durant leur cursus. Ils n'avaient donc pas de moyen de valoriser ces dernières lors de leurs entretiens d'embauche. Ainsi, ce service contribuera à ajouter de la matière à leur bagage, et donc à augmenter leurs chances d'être embauchés. Concernant les bénéfices apportés pour l'établissement, l'application va permettre de renforcer l'attractivité de l'enseignement de HEC Liège et d'augmenter ses chances de conserver ou d'obtenir de nouvelles accréditations.

En ce qui concerne le contexte avant la mise en place de l'application, certaines activités étaient déjà enregistrées dans différentes bases de données en fonction de leur nature (ateliers, séjours...).

L'application que je développe est donc vouée à s'intégrer à ce système d'information plus complexe, et à en exploiter certaines données. En termes de classe d'utilisateurs, l'application est destinée à tous les étudiants de l'école, quel que soit leur domaine d'étude.

Parmi les activités déjà présentes dans les bases de données de l'entreprise, il existe :

- Les ateliers : chaque atelier organisé par HEC Liège figurait dans la base de données SkiWi (Skills and workshops information), celle-ci étant une plateforme qui permet de gérer les ateliers de compétences. Ce sont des workshops organisés comme des cours et qui abordent différents thèmes comme « Booster sa créativité », « Se motiver en toutes circonstances », etc. C'est dans différentes catégories comme le digital, les compétences relationnelles, etc. La plateforme permet de gérer les ateliers, d'assigner les formateurs, de renseigner les étudiants, de permettre aux formateurs de faire des évaluations, etc...
- Les séjours : soit Erasmus ou à double diplôme, les séjours figurent dans la base de données IRIS (International Relations information system) qui, elle, est la plateforme de gestion des relations internationales au sein d'HEC. Elle permet de gérer les inscriptions des étudiants aux séjours

ERASMUS, de gérer la liste des partenaires ainsi que les contrats qui lient HEC à ceux-ci, de permettre une communication vers les étudiants via un système d'envoi de mails.

- Les participations au conseil : pendant leur cursus, les étudiants sont parfois amenés à être représentant de leur section, ou délégués de leurs classes, par exemple. Ce type d'activités est renseigné sur la plateforme SIRH (Système d'information des ressources humaines). C'est la plateforme qui gère les membres du personnel au sein d'HEC; elle reprend les informations des personnes, leur attribution au sein des services, leur contrat, leur historique. Elle permet également de gérer les différents conseils de l'école ainsi que les départements et les services via un système d'organigramme, de gérer les responsabilités institutionnelles de chaque membre du personnel, de faire leur évaluation également. Elle permet aussi, aux membres du personnel académique et scientifique, de demander des valorisations pour leur charge de cours, des attributions de cours supplémentaires etc... Et finalement, elle permet aux responsables des services de gérer les feedbacks des membres qu'ils ont chez eux.

Il faut noter que ces plateformes sont interconnectées sur différents aspects. SKIWI est alimentée, par exemple, avec une partie des membres du SIRH pour gérer la liste des formateurs; le profil international du SIRH sera alimenté par IRIS et inversement. Voici un schéma illustrant ces interactions :

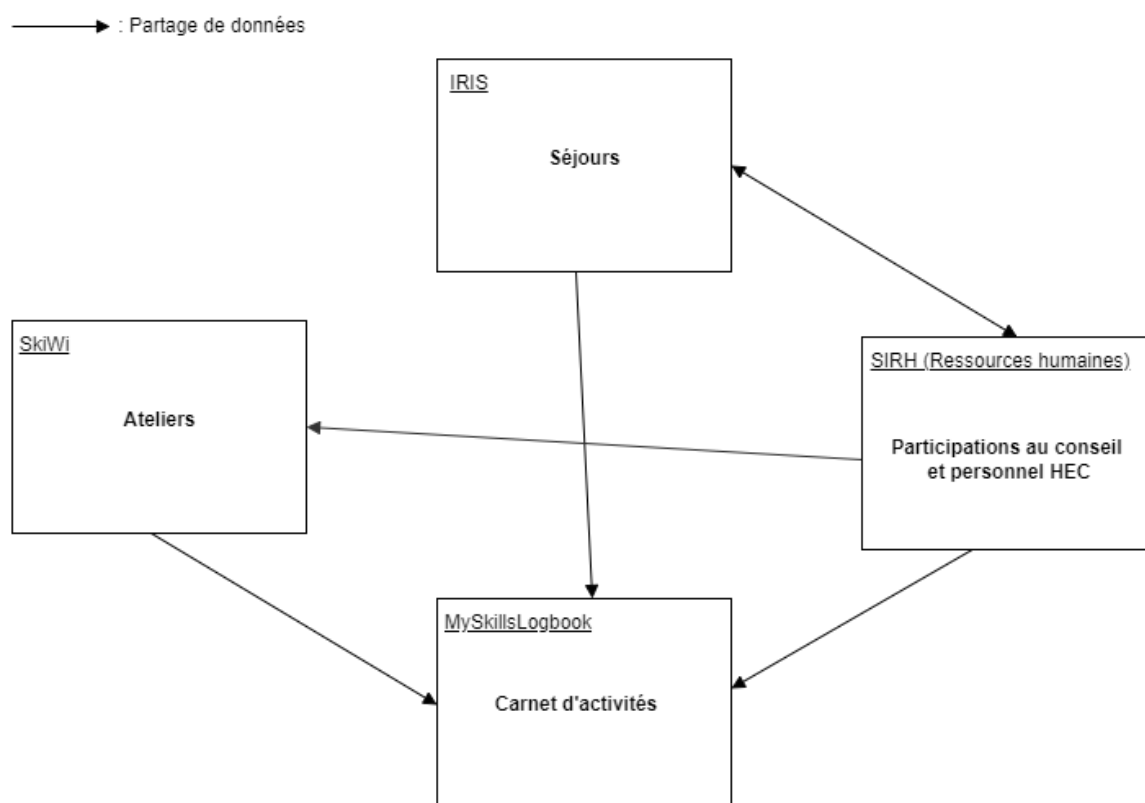


Figure 2 : Interactions entre les différents systèmes d'informations

En effet, les étudiants ont la possibilité de s'inscrire à des séjours Erasmus via d'autres plateformes déjà existantes. Ces inscriptions, elles, sont ensuite

encodées dans la base de données IRIS. Il en est de même pour les inscriptions aux workshops et aux activités, qui elles, sont encodées dans la base de données SkiWi.

Sur le plan technique, l'application a été développée en utilisant les technologies de développement appliquées sur les plateformes déjà existantes à HEC Liège. Les technologies utilisées au sein du système d'information de l'entreprise comprennent LAMP (Linux, Apache, MySQL, PHP). Cette application devra également intégrer des données déjà présentes dans le système d'information de l'école et ainsi, les récupérer en accédant directement (en lecture seule) aux différentes bases de données existantes. De plus, elle devra intégrer le système d'authentification ULiège préexistant afin de permettre aux utilisateurs de s'y connecter à partir de leur compte HEC Liège.

En résumé, l'application que j'ai développée est un nouvel outil qui permettra aux étudiants de HEC Liège de stocker, exporter et valoriser leurs réalisations et projets. Le but étant de renforcer la qualité de l'enseignement de l'école et de créer une application conviviale, intuitive et facile à utiliser pour les étudiants.

2.2. Les besoins et raisons qui ont poussé à proposer ce TFE

Le développement de cette application est issu d'un besoin de l'école HEC Liège. En effet, avant la mise en place de ce projet, les étudiants n'avaient pas de moyen de garder une trace de toutes les activités auxquelles ils avaient participé durant leur cursus. Ce constat a été relevé par plusieurs membres de l'équipe pédagogique de l'école qui ont remarqué que les étudiants ne disposaient pas d'un outil pratique pour valoriser l'ensemble de leurs expériences.

Par conséquent, l'objectif premier de cette application est de permettre aux étudiants de garder une trace de leurs activités, ce qui leur permettra de mieux valoriser leur expérience lorsqu'ils chercheront un emploi. En effet, les recruteurs recherchent de plus en plus de candidats ayant des compétences transversales pouvant être acquises lors de la participation à des activités extrascolaires. Grâce à cette application, les étudiants pourront mettre en avant toutes leurs compétences et toutes leurs expériences, ce qui leur permettra de se démarquer des autres candidats.

De plus, l'application répond également à un besoin de l'école HEC Liège. En effet, en permettant aux étudiants de valoriser leur expérience, elle contribue à renforcer la réputation de l'école. Les recruteurs auront ainsi une meilleure perception de l'enseignement dispensé à HEC Liège, ce qui pourra contribuer à attirer de nouveaux étudiants et à renforcer l'image de l'école.

Le sujet de mon TFE portera donc sur cette application nommée "My skills logbook". Son principal objectif est de développer l'outil en se concentrant sur les besoins décrits plus tôt. Plus précisément, il s'agira de mettre en place les fonctionnalités suivantes :

Création d'activités : Les étudiants peuvent créer des activités telles que des conférences, des visites, des séminaires, etc. Les activités peuvent être créées

par l'étudiant lui-même ou importées directement depuis les différentes bases de données de l'école.

Édition d'activités : Les étudiants ont la possibilité d'éditer les activités qu'ils ont créées ou importées en y ajoutant des informations supplémentaires telles que leurs impressions concernant l'activité, une note sur 5 pour les compétences acquises et des documents tels que des images, des vidéos ou des documents texte.

Agenda des Workshops : Les étudiants peuvent consulter un agenda dans lequel figurent les différents « Workshops » (activités de groupes permettant aux étudiants de faire des comptes-rendus des différentes activités auxquelles ils ont participé) auxquels ils peuvent s'inscrire et consulter les places encore disponibles.

Profil de l'étudiant : Les étudiants peuvent consulter leur profil où seront listées toutes leurs activités. Ce profil est également exportable.

Skills profile : Les étudiants peuvent consulter leur « Skills profile » qui est un rapport de toutes les compétences acquises lors de leurs cursus sous forme de graphiques. Ce rapport est également exportable.

Export de données : Les étudiants peuvent exporter leurs activités et leur « Skills profile » sous forme de fichiers PDF pour les utiliser dans leur CV ou lorsqu'ils cherchent un emploi.

Administration : Les enseignants et le personnel de l'école peuvent utiliser une interface d'administration pour ajouter ou supprimer des activités et pour gérer les inscriptions aux Workshops. Les administrateurs ont aussi la possibilité d'exporter les listes d'inscriptions aux différents workshops sous forme de CSV.

2.3. Objectifs du TFE

En ce qui concerne les objectifs du TFE, plusieurs éléments sont attendus à la fin du travail.

Tout d'abord, une application complète et fonctionnelle qui répond aux besoins et exigences de l'école HEC Liège.

Ensuite, un rapport de TFE complet et détaillé qui explique les différentes étapes du développement de l'application, les choix technologiques effectués, les problèmes rencontrés et les solutions apportées. Ce rapport devra également mettre en avant les bénéfices et les avantages de l'application pour l'école HEC Liège, notamment en termes d'amélioration de la qualité de l'enseignement, de la visibilité de l'école et de la valorisation des compétences des étudiants.

Enfin, une présentation claire et concise de l'application et de ses fonctionnalités qui pourra être utilisée par l'école HEC Liège pour promouvoir et présenter l'application à ses différents publics (étudiants, enseignants, partenaires, etc.). Cette présentation devra être attractive, convaincante et efficace afin de susciter l'intérêt et l'adhésion des utilisateurs potentiels.

Concernant les bénéfices pour l'école HEC Liège, plusieurs points sont à souligner :

En premier lieu, l'application permettra à l'école de mieux suivre et de valoriser les activités des étudiants, ce qui contribuera à améliorer la qualité de l'enseignement et à renforcer la visibilité de l'école auprès de ses différents publics.

Ensuite, l'application permettra aux étudiants de mieux structurer leur parcours de formation, en identifiant les compétences acquises et les activités réalisées tout au long de leur cursus. Cela facilitera leur recherche d'emploi et leur insertion professionnelle, en mettant en avant leurs compétences et leur expérience.

Enfin, l'application permettra à l'école de mieux valoriser ses partenariats et ses collaborations, en mettant en avant les différents événements et activités organisés en collaboration avec des entreprises et des organisations externes. Cela contribuera à renforcer la notoriété et la réputation de l'école, en montrant son dynamisme et son ouverture sur le monde professionnel.

En conclusion, ce TFE vise à développer une application nouvelle et utile pour l'école HEC Liège, en répondant à des besoins précis et en offrant des avantages significatifs pour l'école et ses étudiants.

3. Méthodologie

Dans ce chapitre, je vais décrire comment notre travail s'est organisé au sein de l'entreprise et les outils qui ont été utilisés pour faciliter cette organisation.

3.1. Organisation du travail

Lors de mon stage, j'ai été confronté à une organisation du travail qui combinait le travail solitaire avec le soutien de mon équipe. J'ai bénéficié d'une grande autonomie dans l'exécution de mes tâches, sans pour autant suivre une méthodologie stricte. Les principes agiles ont néanmoins guidé notre approche de travail. Les voici :

- **Accent sur la satisfaction client dans nos processus** : L'application que j'ai développée était destinée aux étudiants et nous avons veillé à ce qu'elle soit attrayante et agréable pour ses utilisateurs. Tout au long du développement, j'ai maintenu une communication constante avec les différentes parties prenantes du projet afin de prendre en compte leurs attentes et d'apporter des ajustements si nécessaire. Il me faut préciser que parmi ces parties prenantes ne figuraient que des membres du personnel en charge du projet. Les retours provenant des étudiants, eux, ne seront entendus qu'une fois l'outil présenté et rendu accessible par l'établissement, cela plusieurs mois après la fin de mon stage.
- **Intelligence collective** : Notre équipe était composée de membres aux compétences diverses, favorisant ainsi l'intelligence collective. Outre les développeurs, nous avons également une spécialiste en analyse de données. J'ai également eu l'occasion de collaborer avec des personnes d'autres services, par le biais de réunions et d'échanges de courriers électroniques, telles qu'un responsable de la communication, ainsi que les décideurs clés du projet. Cette diversité nous a permis d'obtenir des perspectives variées et d'enrichir notre approche.
- **Développement évolutif** : Le développement de l'application était une amélioration continue. Certaines fonctionnalités et modifications ont été ajoutées au projet même si elles ne figuraient pas dans le cahier des charges initial. Nous avons adopté un cycle de développement itératif, incrémental et adaptatif en découpant le projet en deux grandes phases : d'abord, l'implémentation des interfaces utilisateurs puis, l'implémentation de l'API répondant aux requêtes. Celles-ci étant elles-mêmes découpées en plusieurs sous-phases.

3.2. Outils

Pour organiser nos tâches, nous avons utilisé un tableau interactif Trello. Initialement édité par mon équipe, j'ai également contribué à sa mise à jour de temps à autre.

Les maquettes d'interfaces utilisateur ont été fournies via l'outil Miro. Bien que je n'aie pas participé à leur réalisation, j'ai pu adapter l'interface au fur et à mesure du développement pour intégrer de nouvelles fonctionnalités.

3.3. Réflexions personnelles

Au cours de mes études, j'ai appris l'existence de méthodologies de travail plus strictes lors du développement d'outils informatiques, comme par exemple la méthodologie SCRUM¹ vue lors du cours d'informatique managériale. Cependant, étant le seul à travailler sur ce projet, mon expérience lors de ce stage a été caractérisée par une plus grande liberté dans l'organisation du travail. Cela m'a donné l'opportunité d'organiser mes tâches à ma guise tout en respectant le déroulement du développement tel qu'imaginé par mon équipe. En revanche, n'ayant pas été confronté à la réalité d'une organisation plus stricte, il me restera cette part du métier à découvrir lors de ma future vie professionnelle, bien que j'ai théoriquement connaissance de ces types d'organisations.

En conclusion, cette expérience m'a permis d'appréhender la réalité du travail en entreprise, où différentes méthodologies, outils et formes d'organisation coexistent. J'ai pu expérimenter les principes agiles dans un contexte pratique, en adaptant mon approche en fonction des besoins du projet et des attentes des parties prenantes. Cette expérience a renforcé ma capacité à travailler de manière autonome et en équipe, en intégrant les aspects de réactivité, flexibilité et amélioration continue dans ma pratique professionnelle future.

¹ Wikipedia. Sur le site Wikipedia. Consulté le 25/05/2022.
[https://fr.wikipedia.org/wiki/Scrum_\(d%C3%A9veloppement\)](https://fr.wikipedia.org/wiki/Scrum_(d%C3%A9veloppement))

4. Analyse

Ce chapitre va me permettre de décrire en détails les fonctionnalités et/ou les attentes qui ont été définies pour notre projet.

4.1. Fonctionnalités

Les fonctionnalités ont été décomposées en plusieurs user stories, représentées ci-dessous par ce diagramme de cas d'utilisation :

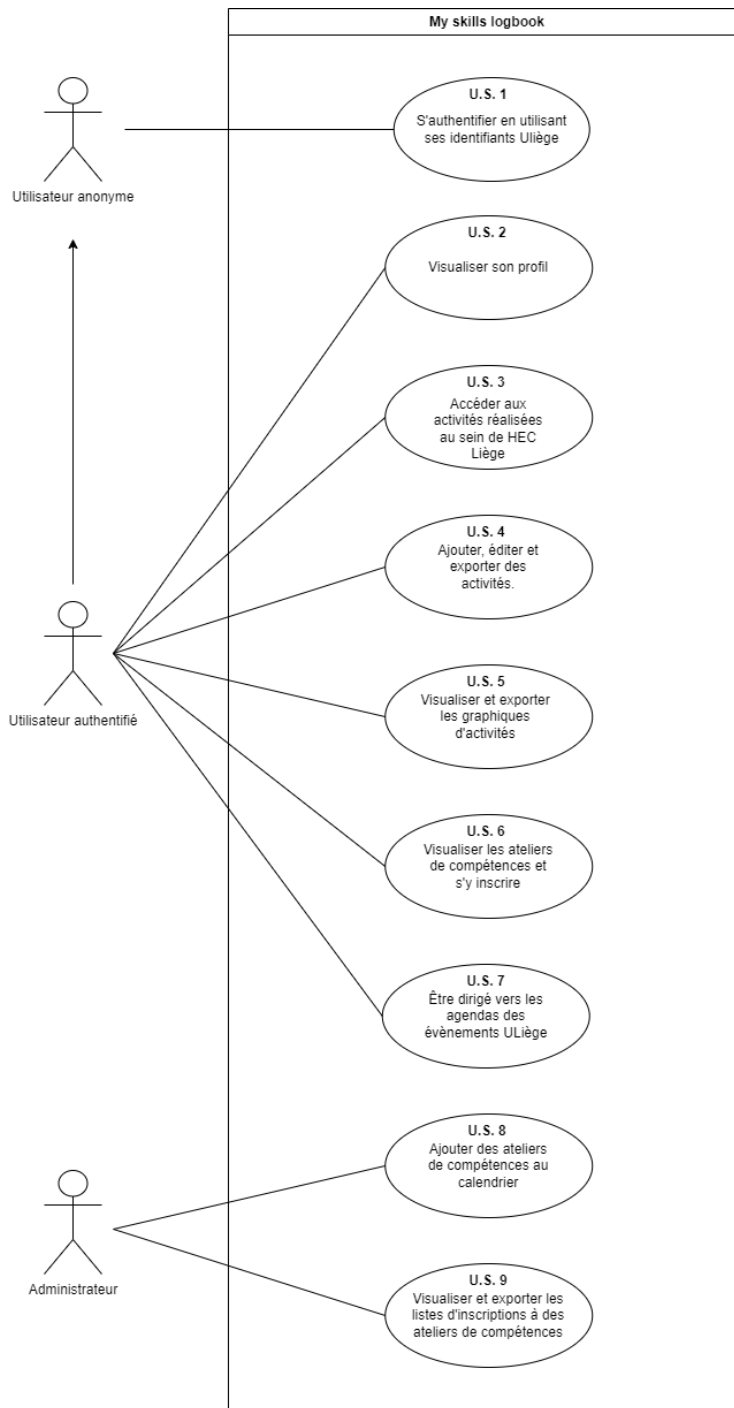


Figure 3 : diagramme de cas d'utilisation

User Stories

1. En tant qu'utilisateur, je souhaiterais pouvoir m'authentifier via le système d'authentification ULiège afin d'accéder à l'application.

Préconditions : L'utilisateur doit disposer d'un matricule dans le domaine ULiège.

Postconditions : L'application coté serveur dispose des informations de l'utilisateur et une session lui est créée.

2. En tant qu'utilisateur authentifié, je souhaiterais avoir accès à mes données personnelles afin de visualiser mon profil.

Préconditions : L'utilisateur est un étudiant sinon l'application présentera un mode dédié à la démonstration et/ou à l'administration.

Postconditions : L'utilisateur peut visualiser son profil avec ses données personnelles (nom, prénom, photo de profil et cursus suivi).

3. En tant qu'utilisateur authentifié, je souhaiterais avoir accès aux activités auxquelles j'ai participé au sein de HEC Liège afin de les visualiser, les éditer et ensuite, les exporter.

Préconditions :

- Afin de pouvoir les visualiser et les éditer, l'utilisateur doit avoir été inscrit à des activités organisées par HEC Liège, recensées dans les bases de données de l'école. Ces activités seront des ateliers auxquels ils auraient participé, des séjours Erasmus / à double diplôme ou encore des rôles qu'ils auraient endossés au sein du conseil HEC.
- Afin de pouvoir les exporter, l'utilisateur doit avoir, au préalable, édité ces activités.

Postconditions :

- L'utilisateur peut visualiser ses activités dans son profil et éditer les champs dont les informations n'ont pas été importées depuis les différentes bases de données.
- L'utilisateur a la possibilité d'exporter ses activités sous forme de fichier PDF une fois qu'elles ont été éditées.

4. En tant qu'utilisateur authentifié, je souhaiterais pouvoir ajouter des activités à mon profil afin de les visualiser, les éditer et les exporter.

Préconditions : L'utilisateur doit être authentifié.

Postconditions : L'utilisateur peut visualiser les activités qu'il a ajoutées, les mettre à jour et les supprimer depuis son profil.

N.B. : Concernant l'édition, l'utilisateur doit pouvoir renseigner des informations sur une activité (nom, date, type, catégories et dimensions), donner ses impressions concernant celle-ci, donner une estimation sur

les compétences qu'elle lui aurait apportées, ainsi que lui assigner des images, des documents ou encore des liens.

5. En tant qu'utilisateur authentifié, je souhaiterais pouvoir visualiser mes activités éditées sur des graphiques afin de les exporter et de visualiser les compétences qu'elles m'auront apportées au fil du temps.

Préconditions : L'utilisateur doit avoir évalué au moins une compétence concernant une activité pour qu'elle soit représentée sur les graphiques.

Postconditions :

- L'utilisateur peut visualiser son évolution sur des graphiques.
- L'utilisateur peut exporter ceux-ci sous forme de fichier PDF.

6. En tant qu'utilisateur authentifié, je souhaiterais pouvoir visualiser les ateliers de compétences à venir afin de m'y inscrire.

Préconditions :

- Des ateliers de compétences doivent avoir été encodés au préalable par un administrateur.
- Afin de s'y inscrire, l'utilisateur doit avoir ajouté de lui-même un minimum de 5 activités dans son profil. En effet, il doit avoir participé à 5 activités ne provenant pas de l'application pour commencer à profiter des activités relayées par celle-ci. De plus, il doit rester des places disponibles aux activités permettant encore l'inscription.

Postconditions :

- Les ateliers sont affichés avec leur nom, leur date ainsi que le nombre de places restantes.
- Après une inscription de l'utilisateur, l'interface lui signalera qu'il est inscrit et il recevra un mail de confirmation.

7. En tant qu'utilisateur authentifié, je souhaiterais pouvoir être dirigé vers les agendas des différentes plateformes de l'Université de Liège afin de me tenir au courant des différents événements à venir.

Préconditions : L'utilisateur s'est rendu dans l'onglet « Agenda ».

Postconditions : L'utilisateur peut se diriger vers les agendas des différentes plateformes en suivant les liens vers ceux-ci.

8. En tant qu'administrateur, je souhaiterais pouvoir ajouter des ateliers de compétences à la liste des événements à venir afin de permettre aux étudiants de s'y inscrire.

Préconditions : L'utilisateur doit disposer des droits d'administrateur sur la plateforme.

Postconditions : L'utilisateur a la possibilité d'ajouter des ateliers de compétences qui seront renseignés dans le calendrier, étant la page

listant les ateliers de compétences à venir et où les étudiants auront la possibilité de s'inscrire.

9. En tant qu'administrateur, je souhaiterais avoir accès aux listes des inscriptions aux ateliers de compétences afin de pouvoir les exporter.

Préconditions : Il doit exister des ateliers de compétences et l'utilisateur doit être administrateur.

Postconditions : L'utilisateur peut exporter la liste des inscriptions à un atelier de compétences sous forme de fichier CSV.

4.2. Modèle de données

Voici mon modèle de données. Comme j'étais seul sur un projet en début de développement, cette tâche a été ma première priorité. Je l'ai affiné au cours de mon stage, en tenant compte des problèmes rencontrés et avec l'aide de mon équipe. Je vais ici me concentrer sur certaines particularités qu'il présente.

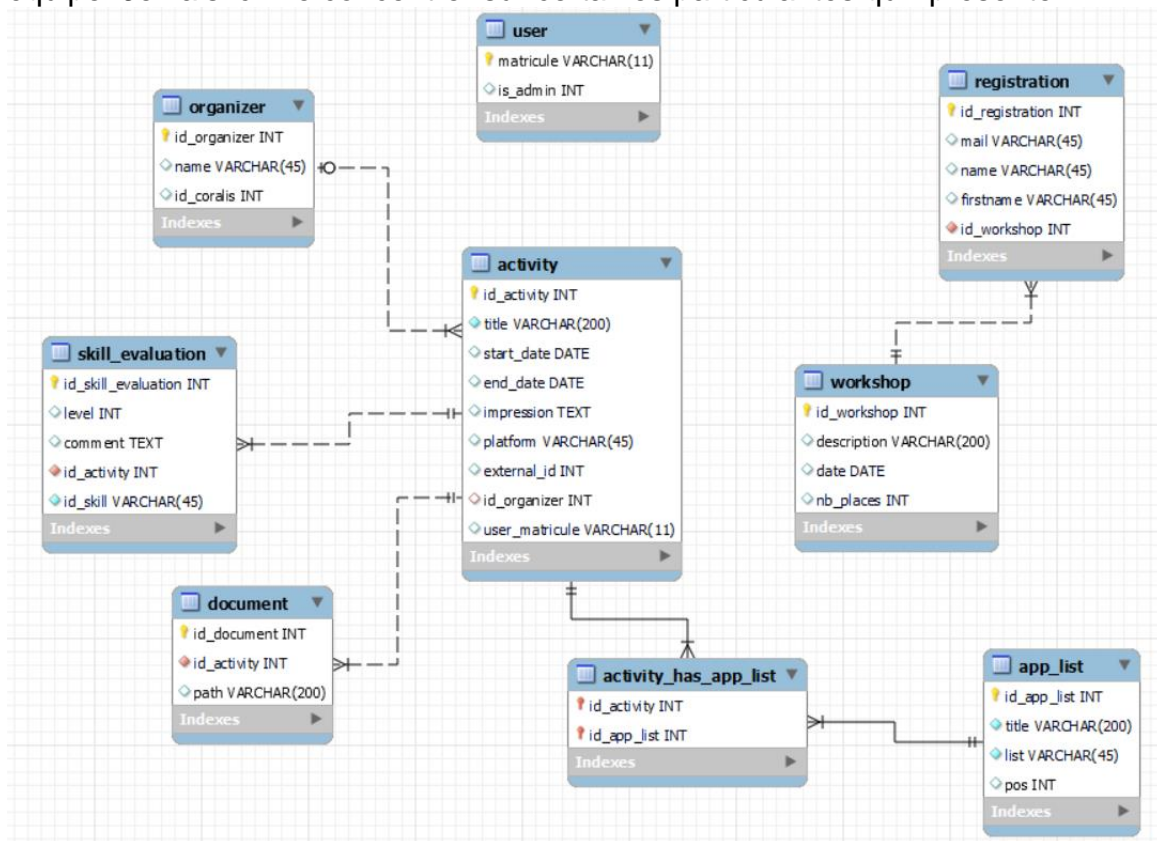


Figure 4 : modèle de données

Au cours de la conception, plusieurs problèmes se sont posés :

L'enregistrement des utilisateurs :

À première vue, un détail pourrait sembler troublant : les utilisateurs ne sont liés à aucune table. En réalité, toutes les données des utilisateurs proviennent d'une autre base de données du système. On remarque que la table "activité" comporte une colonne "user_matricule". Cela permet d'enregistrer uniquement les utilisateurs ayant le statut d'administrateur dans la base de données de mon

application. Ainsi, nous économisons de l'espace sur le serveur en évitant la duplication de données inutiles.

Les activités provenant d'autres plateformes :

Comme on peut le constater, la table "activity" comporte une colonne "platform" et une colonne "external_id". Ces deux propriétés permettent d'indiquer l'origine des activités importées depuis d'autres plateformes. Elles représentent respectivement le nom de la plateforme d'origine et l'identifiant de l'activité au sein de cette dernière.

Les organisateurs externes :

Dans l'application, une activité peut avoir deux types d'organiseurs. D'une part, ceux encodés manuellement dans la base de données. D'autre part, ceux provenant d'une autre plateforme nommée "CORALIS". Cette dernière répertorie un grand nombre de partenaires de HEC Liège. Un mécanisme d'autocomplétion a été intégré à l'application, permettant de récupérer et suggérer tous les partenaires de la plateforme CORALIS. Lorsqu'un utilisateur enregistre un organisateur via ce mécanisme, l'organisateur est alors enregistré avec son nom et son identifiant externe dans la base de données de l'application.

4.3. Architecture logicielle

En ce qui concerne le diagramme de déploiement de l'application, en voici une représentation :

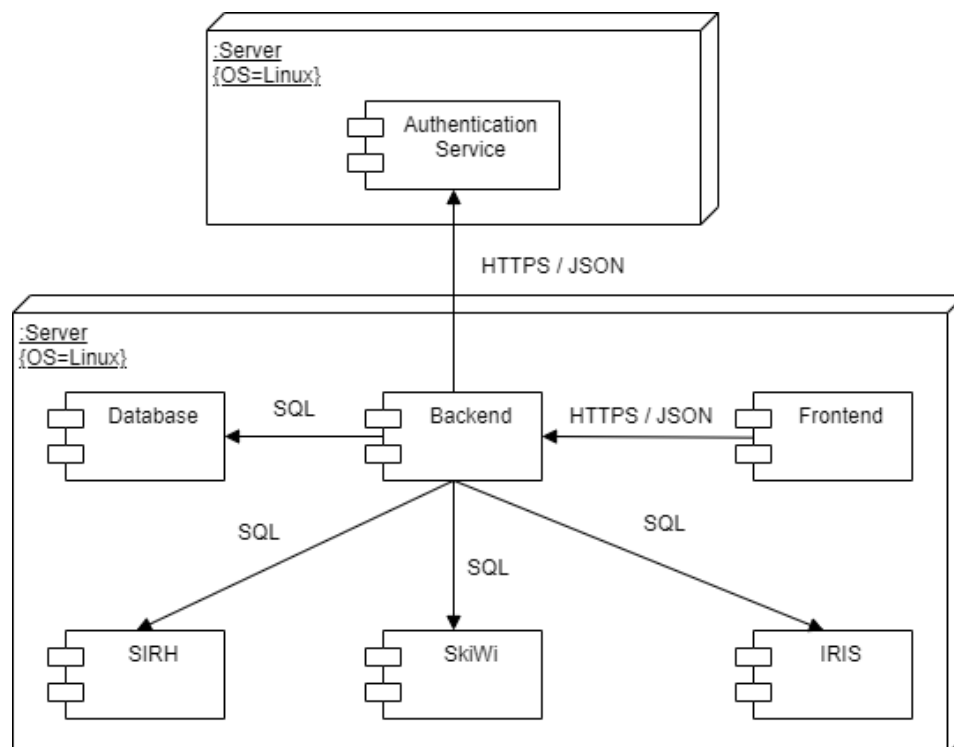


Figure 5 : diagramme de déploiement

L'architecture est de type 3-tiers². En effet, le tout étant déployé sur un serveur Linux hébergé par l'entreprise, la couche de logique (API backend PHP) s'y exécute et les données y sont enregistrées et récupérées (bases de données MariaDB). La couche de présentation (frontend HTML/CSS alimenté par du javascript), elle, s'exécute sur le navigateur web du client et est déployée sur le serveur web Apache lui-même déployé sur le serveur Linux.

Le backend transmet les données au frontend grâce à des requêtes http qui retournent le résultat au format JSON.

Le service d'authentification, lui, est déployé sur un serveur Linux à l'Université de Liège.

4.4. Technologies

Les technologies que j'ai utilisées pour développer cette solution m'ont été imposées par mon entreprise, bien que j'aie pris la liberté d'en ajouter certaines pour mieux répondre à mes besoins. Il est important de préciser qu'aucun framework n'a été utilisé pour le développement de cette application. Après en avoir discuté avec mon équipe, qui n'était pas encore formée à l'utilisation de frameworks, nous avons convenu qu'un projet vanilla serait plus simple à maintenir pour eux par la suite.

4.4.1. PHP

Cette technologie est au centre du système informatique que mon équipe a développé pour HEC Liège. En effet, toutes les plateformes déployées sur le serveur sont en PHP. Même si cela m'a été imposé, il était pertinent de garder la même ligne directrice afin de faciliter la maintenance de mon travail au cas où d'autres développeurs de l'équipe devaient s'y atteler.

4.4.2. PHPUnit

PHPUnit³ est un framework de test unitaire pour PHP, largement utilisé pour écrire et exécuter des tests automatisés afin de vérifier le bon fonctionnement des différentes parties d'une application développée en PHP.



Figure 6 : logo PHPUnit

Avec l'accord de mon équipe, j'ai pris la liberté d'ajouter cette technologie à ma solution pour garantir et démontrer son bon fonctionnement. Bien que cela ne m'ait pas été explicitement demandé, cette technologie me permettait de tester localement certaines fonctionnalités sans avoir à redéployer constamment mes

² Jean Jadot. « Introduction » [« Architecture 3-tiers »]. Consulté le 25/05/2022. <https://learn-technique.helmo.be/course/view.php?id=246>

³ PHPUnit. Sur le site PHPUnit. Consulté le 25/05/2022 <https://phpunit.de/>

fichiers sur le serveur. De cette manière, j'ai également pu faire découvrir une nouvelle technologie à mon équipe.

4.4.3. MySQL

De la même manière que pour PHP, cette technologie m'a été imposée pour permettre à mon application de coexister de façon harmonieuse avec les autres plateformes existantes.

4.4.4. HTML

Allié à du PHP et à du JavaScript vanilla, le HTML était donc un incontournable pour développer mon application web.

4.4.5. CSS

Allié à Bootstrap, le CSS m'a permis d'intégrer les interfaces utilisateur de manière plus flexible et personnalisée, notamment pour intégrer des graphiques conçus par les designers, pour lesquels aucune librairie existante n'aurait permis un rendu équivalent. Son utilisation m'a également permis de maîtriser des détails que Bootstrap aurait imposés, et d'assurer une meilleure évolution de l'interface dans les futures itérations du développement de la solution.

4.4.6. Bootstrap

Il est très utilisé par l'équipe de développement avec laquelle j'ai travaillé et m'a évité de recréer des classes CSS. Ainsi, mon équipe n'a pas eu besoin de se familiariser avec de nouvelles classes lors de la maintenance de la solution.

4.4.7. Javascript

Afin de faciliter la maintenance de la solution après mon départ, il a été convenu que l'utilisation d'un framework frontend n'était pas adaptée au contexte dans lequel je me trouvais. J'ai donc développé toute la partie front-end à l'aide de JavaScript vanilla, en utilisant aussi jQuery. Cela m'a permis de plonger profondément dans le langage et ses subtilités, notamment lorsque j'ai dû implémenter mon propre système de routage dans l'application.

4.4.8. JQuery

jQuery est largement utilisé par mon équipe et m'a permis de rendre mon code plus lisible et concis. De plus, j'ai utilisé sa méthode `$.ajax()` pour chaque requête vers le backend. Bien que j'aie eu quelques notions de jQuery grâce à un cours de JavaScript, j'ai pu approfondir mes connaissances sur cette bibliothèque pendant mon stage.

4.4.9. PDF Make

PdfMake⁴ est une bibliothèque JavaScript qui permet de générer des fichiers PDF côté client en utilisant une syntaxe simple et déclarative.



Figure 7 : logo PDF Make

Cette librairie m'a permis de générer des fichiers PDF lors de l'implémentation des exports que la solution propose.

4.4.10. HTML2Canvas

Html2canvas⁵ est une bibliothèque JavaScript qui capture le rendu d'une page web et génère une image à partir de son contenu HTML et CSS.

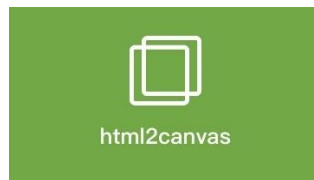


Figure 8 : logo html2canvas

Cette bibliothèque m'a permis de convertir des éléments DOM en images directement insérables dans les fichiers PDF. Elle est couramment utilisée pour ce type d'opération.

4.4.11. DOM to Image

Dom-to-image⁶ est une bibliothèque JavaScript qui permet de convertir le contenu d'un élément DOM en une image.



Figure 9 : logo dom-to-image

Cette bibliothèque est très similaire à la précédente car elle offre presque les mêmes fonctionnalités. Elle m'a été utile car la précédente ne prenait pas en

⁴ pdfmake. Sur le site pdfmake. Consulté le 25/05/2023

<http://pdfmake.org/#/>

⁵ html2canvas. Sur le site html2canvas. Consulté le 25/05/2023

<https://html2canvas.hertzen.com/>

⁶ Tsayen. Sur le site npm. Consulté le 25/05/2023

<https://www.npmjs.com/package/dom-to-image>

charge certaines règles CSS, comme les ombrages, qui étaient indispensables pour le rendu de l'export des graphiques.

4.5. Outils de développement

4.5.1. Visual studio code



Figure 10 : logo Visual Studio Code

C'est un choix personnel d'avoir utilisé Visual Studio Code. C'est un outil avec lequel j'ai l'habitude de travailler et dans lequel je me sens à l'aise. Léger et offrant de nombreuses extensions, c'est un environnement de travail que j'apprécie particulièrement.

4.5.2. PHPMyAdmin

PHPMyAdmin⁷ est une interface web qui permet de gérer les bases de données MySQL de manière visuelle et conviviale en facilitant les tâches d'administration telles que la création de tables, la gestion des utilisateurs et l'exécution de requêtes SQL.



Figure 11 : logo phpMyAdmin

Auparavant, je n'avais jamais utilisé cette interface web pour gérer mes bases de données. Elle m'a été imposée, mais je l'ai trouvée instinctive et facile d'utilisation dès ma première journée de stage. Elle m'a permis de naviguer aisément entre les différentes bases de données de l'entreprise.

4.5.3. MySQL Workbench

MySQL Workbench⁸ est un outil graphique de modélisation, développement et administration de bases de données MySQL, permettant de concevoir, gérer et exécuter des requêtes SQL de manière conviviale et intuitive.

⁷ PHPMyAdmin. Sur le site PHPMyAdmin. Consulté le 25/05/2023

<https://www.phpmyadmin.net/>

⁸ MySQL. Sur le site MySQL. Consulté le 25/05/2023



Figure 12 : logo MySQL Workbench

Comme PHPMyAdmin, je n'avais jamais utilisé cet outil pour modéliser et générer des bases de données auparavant, mais sa prise en main a été rapide et agréable. Il m'a accompagné tout au long du développement lorsque j'adaptais la structure de ma base de données.

4.5.4. GIMP

GIMP⁹ est un logiciel de manipulation d'images gratuit et open source offrant des fonctionnalités avancées pour l'édition et la retouche d'images. Il est utilisé par les designers et les développeurs pour créer et modifier des éléments graphiques tels que des logos, des icônes et des images pour les sites web.



Figure 13 : logo GIMP

Ne possédant pas de licence pour Photoshop, j'ai découvert cet outil qui m'a beaucoup aidé dans mon travail lorsque j'ai dû manipuler des images PNG avec transparence. Il m'a également permis de générer des images que j'ai pu réutiliser lors de l'export des fichiers PDF.

<https://www.mysql.com/products/workbench/>

⁹ GIMP. Sur le site GIMP. Consulté le 25/05/2023

<https://www.gimp.org/>

5. Réalisation

Dans ce chapitre, je vais vous partager les étapes clés de la réalisation de mon projet. Je vous dévoilerai comment j'ai créé ma base de données, géré l'authentification, intégré les interfaces et surmonté les défis liés à la manipulation de l'historique sans l'aide d'un framework. De plus, je vous expliquerai comment j'ai mis en place mon API backend sans l'utilisation d'un framework également. Enfin, je vous livrerai un compte rendu complet de cette expérience.

5.1. Création de la structure de la base de données

Ma première journée sera consacrée à la création de ma base de données, bien qu'elle sera modifiée tout au long de mon stage. En effet, je commencerai par la modéliser avec MySQL Workbench pour ensuite en exporter le script et l'exécuter sur phpMyAdmin.

5.2. Authentification

Lors de ma deuxième journée, j'intégrerai l'authentification ULiège à mon projet. Ce service m'a été fourni dès le début du projet par le biais d'une librairie ayant été développée par l'Université de Liège. Je n'ai donc pas eu accès aux détails de son implémentation. Celle-ci me permettra de récupérer les informations de l'utilisateur authentifié dans ma session PHP et de créer un cookie de session dans le navigateur.

5.3. Intégration des interfaces

La suite du programme consistait à intégrer les interfaces conçues préalablement par les différentes parties prenantes du projet, partagées via l'outil en ligne Miro.

Le premier défi était de rendre les interfaces responsives en suivant l'approche "Mobile first". Cela signifie que toutes les maquettes fournies étaient initialement prévues pour les smartphones. Lors de l'adaptation à des écrans plus grands et lors de rotations, il a été nécessaire de trouver un moyen de remplir l'espace vide. Pour cela, j'ai utilisé des flexboxes et j'ai choisi une approche courante en laissant des espaces blancs aux bords de l'écran pour compenser l'excès de largeur.

Le second défi a été d'implémenter des graphiques représentant le niveau de compétences acquises lors de chaque activité. Ces graphiques permettaient d'avoir une représentation visuelle directe des compétences acquises par l'étudiant au cours de son parcours.

Voici l'interface en question :

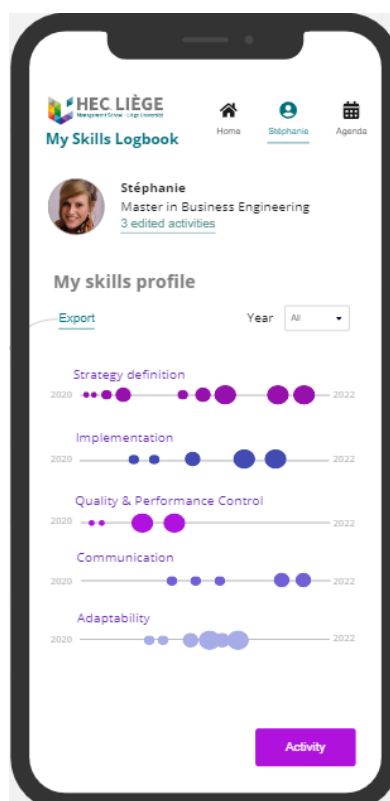


Figure 14 : Graphiques d'évaluation de compétences

En effet, ces graphiques ont été conçus par l'équipe pédagogique lors de la création des maquettes de l'application. Il n'existait pas de bibliothèques capables de fournir ce type de résultat, alors j'ai pris plaisir à les concevoir entièrement en utilisant CSS et Bootstrap. Bootstrap, avec son système de grille à 12 colonnes, m'a permis de représenter les 12 mois d'une année sur la ligne du temps. Le reste du design a été réalisé avec CSS.

5.4. Gestion de la navigation et de l'historique

Une fois les interfaces intégrées, il était temps de les rendre fonctionnelles. J'ai d'abord travaillé sur la navigation de la barre et des boutons en général.

Cependant, j'ai rencontré un problème : mon historique de navigation ne fonctionnait pas, ce qui signifiait qu'il n'était pas possible de revenir en arrière. Habituellement, PHP gère à la fois le frontend et le backend, mais j'ai décidé de ne pas l'utiliser pour le frontend afin de maintenir une séparation claire. Ainsi, tout le frontend a été géré avec JavaScript et jQuery pour assurer une cohérence maximale.

Lors de la navigation entre les pages, j'ai chargé dynamiquement mes templates PHP (les structures HTML de mes pages) à l'aide de JavaScript. Cependant, cela n'avait aucun impact sur l'historique ni sur l'URL de ma page.

Pour résoudre ce problème, j'ai travaillé avec mon équipe pour trouver une solution sans utiliser de framework frontend. J'ai essayé de manipuler l'historique à l'aide de l'API History de JavaScript, mais je me suis rendu compte que cela ne fonctionnait pas comme prévu. Je n'arrivais pas à contrôler facilement des éléments tels que l'URL actuelle, qui restait constamment la même (par exemple : `.../index.php`).

C'est alors que j'ai eu l'idée d'implémenter mon propre routeur, inspiré par ceux que j'avais utilisés dans des frameworks comme ReactJS avec `react-dom-router`. J'ai choisi d'implémenter un HashRouter. L'idée derrière ce routeur est d'utiliser le hash de l'URL, c'est-à-dire tout ce qui suit le hashtag dans l'URL (par exemple : `https://monUrl/index.php#nomDeMaPage`).

Un hashtag ajouté au début de l'attribut "href" d'une balise "`<a>`" annule le rechargement de la page lorsque ce lien est cliqué. Par exemple, l'utilisation de `` permet de modifier l'URL de la page lors d'un clic sans recharger la page. J'ai ensuite écouté l'événement `window.onhashchange` pour détecter les changements de hash et j'ai associé chaque hash au template HTML correspondant. Une fois cela fait, j'ai chargé le template approprié au bon endroit à l'aide de JavaScript.

J'ai choisi d'insérer les vues dans une balise avec l'identifiant "router", juste en dessous de ma barre de navigation. Cette balise contenait tout le contenu de la page, à l'exception de la barre de navigation qui restait fixe.

Plus tard, j'ai amélioré ce système en ajoutant des sous-routes pour avoir un niveau supplémentaire dans l'arborescence de navigation. J'ai légèrement ajusté l'algorithme pour prendre en charge les nouveaux paramètres d'URL et pour charger les templates appropriés dans des balises avec l'identifiant "subRoute".

5.5. CRUD

Avec la partie frontend fonctionnelle, j'ai pu me concentrer sur la partie backend.

J'ai commencé par créer toutes les classes du modèle correspondant aux tables de la base de données. Ensuite, j'ai implémenté progressivement les répertoires avec les opérations CRUD de base.

Pour m'assurer que toutes mes requêtes fonctionnaient correctement, j'ai utilisé la bibliothèque `phpUnit`. Cela m'a permis d'automatiser des tests unitaires pour chacune des opérations. `phpUnit` m'a également permis d'exécuter les requêtes directement à partir d'une ligne de commande dans le terminal, sans avoir besoin de redéployer mes fichiers sur le serveur.

5.6. API

Les opérations de base de manipulation de données étaient maintenant implémentées. Il a donc fallu architecturer mon backend d'une façon maintenable et extensible.

Dans mon parcours, la dissociation entre backend et frontend s'est surtout faite lorsque nous avons commencé à utiliser des frameworks, tant backend que frontend. J'ai donc pris l'habitude d'implémenter des API's lorsqu'il s'agissait de réaliser ce genre de tâches. Or, les frameworks backend que j'ai pu utiliser nous fournissent un cadre de travail où il nous suffit de créer des contrôleurs dans lesquels nous créons des endpoints¹⁰ qui eux, sont testables directement depuis des interfaces web telles que Swagger¹¹ ou PostMan¹².

N'ayant pas beaucoup pratiqué le PHP avant de me lancer dans ce projet, j'ai appris que dans PHP, un endpoint classique pouvait être un simple fichier PHP qui exécuterait un script et renverrait la réponse à l'aide d'un echo. J'ai également découvert qu'on pouvait utiliser des headers pour gérer la réponse. Cela m'a permis de mieux comprendre et de faire le lien avec les APIs que je connaissais déjà.

Cependant, je n'avais pas envie de créer un fichier PHP distinct pour chaque requête vers mon API, ni d'imbriquer des conditions dans un script procédural qui aurait servi de contrôleur. Je voulais une structure d'URL de requête similaire à celle que l'on trouve généralement dans les APIs, comme par exemple `api/controller/action?parametre=valeur`. De plus, je voulais des contrôleurs orientés objet comme ceux que l'on trouve dans les frameworks modernes.

Pour réaliser cela sans utiliser de framework, j'ai fait appel à ma créativité et j'ai mené des recherches approfondies pour savoir comment implémenter une API orientée objet en PHP.

Je vais donc vous présenter le cheminement que j'ai suivi lors de la création de cette API.

5.6.1. Contrôleurs

Je voulais absolument avoir en guise de contrôleurs, des classes liées à des objets de mon modèle et à leurs répertoires. Mes contrôleurs disposeraient donc du répertoire de l'objet en question et s'en serviraient pour interagir avec la base de données. Dans ces contrôleurs, chacune des méthodes représenterait donc un endpoint qui permettrait de retourner le résultat par requête http au client.

5.6.2. Fichier d'entrée

Pour parvenir à cela, j'ai d'abord commencé par créer un fichier `api.php` à la racine du dossier contenant mon backend, c'est-à-dire le dossier "src" lui-même situé à la racine du projet. Ce fichier servira de point d'entrée pour toutes les requêtes envoyées vers mon API.

¹⁰ Bénédicte Brossault. Sur le site [blog.hubspot.fr](https://blog.hubspot.fr/website/endpointapi#:~:text=Sur%20une%20API%2C%20un%20endpoint,l'API%20envoie%20ses%20requ%C3%AAtes). Consulté le 26/05/2023
<https://blog.hubspot.fr/website/endpointapi#:~:text=Sur%20une%20API%2C%20un%20endpoint,l'API%20envoie%20ses%20requ%C3%AAtes>.

¹¹ Swagger. Sur le site Swagger. Consulté le 26/05/2023
<https://swagger.io/>

¹² Postman. Sur le site Postman. Consulté le 26/05/2023
<https://www.postman.com/>

Voici le fichier d'entrée de l'API :

```
src > api.php
1  <?php
2  /**
3   * Point d'entrée des requêtes ajax.
4   */
5  namespace Api;
6
7  require_once '../vendor/autoload.php';
8  require_once "../services/pdo/requetes.php";
9  require_once "config.php";
10
11 use Api\Controllers\MainController;
12
13 ini_set('display_errors', 1);
14 ini_set('display_startup_errors', 1);
15 error_reporting(E_ALL);
16
17 // Création du contrôleur principal sur base de l'url de la requête.
18 new MainController($_SERVER['REQUEST_URI'], $db);
19 ?>
```

Figure 15 : point d'entrée des requêtes ajax

Ces requêtes auront donc pour URL : `src/api.php/....`. Ce n'est pas tout à fait le type d'URL que l'on retrouve dans une API web classique, mais j'ai considéré que cela relevait du détail. L'essence même du mécanisme que je souhaitais était respectée. De plus, en suivant ce schéma, il serait possible de créer plusieurs APIs aux responsabilités différentes et de les utiliser en tant que microservices.

5.6.3. Redirection vers les contrôleurs

Désormais, j'avais besoin d'un système pour que chaque contrôleur soit instancié de façon adéquate suivant la requête émise par le client. Par exemple, une requête de type POST composée de « `src/api.php/activity/create` » devrait donc instancier la classe « `ActivityController` » et exécuter sa méthode « `create` ».

Pour cela, j'ai créé une classe nommée « `MainController` », qui sera dédiée à la redirection vers le contrôleur adéquat et qui exécutera la méthode souhaitée au sein de celui-ci. Cette classe « `MainController` » prendra en paramètres de son constructeur :

- L'url de la requête afin d'en extraire les paramètres, d'instancier le contrôleur approprié ainsi que d'en appeler la méthode souhaitée.
- La connexion à la base de données afin de la transmettre au contrôleur qui lui-même la transmettra aux répertoires qu'il utilise.

Voici la classe MainController :

```
src > Controllers > MainController.php
1  <?php
2
3  namespace Api\Controllers;
4
5  require_once '../vendor/autoload.php';
6
7  class MainController {
8
9      public function __construct($url, $db){
10         // Explosion de l'url en un tableau de paramètres
11         $urlSegments = explode("/", $url);
12         // Récupération du nom du contrôleur qui va gérer la requête (4e case du tableau)
13         $controller = $urlSegments[3];
14         // Récupération de l'action à exécuter dans le contrôleur (nom de la méthode et 5e case du tableau)
15         $action = explode("?", $urlSegments[4])[0];
16         $this->route($controller, $action, $db);
17     }
18
19     /**
20      * Fonction qui crée le contrôleur et fait appel à la méthode à exécuter.
21      */
22     private function route($controllerName, $actionName, $db){
23         // Nom complet de la classe du contrôleur (on passe la première lettre en majuscule)
24         $className = "Api\\Controllers\\".ucfirst($controllerName). "Controller";
25         // Instanciation du contrôleur, on lui donne la bd pour lui permettre de faire des requêtes
26         $controller = new $className($db);
27         // Appel de la méthode à exécuter dans le contrôleur
28         $controller->{$actionName}();
29     }
30 }
31 ?>
```

Figure 16 : Redirection vers les contrôleurs

5.6.4. Utilisation des contrôleurs

À partir de ce moment, j'avais une API dont le fonctionnement était similaire à celui généré par un framework backend. Cela m'a permis de commencer à implémenter mes contrôleurs. Pour cela, il suffisait de créer une classe contrôleur avec le nom souhaité et de m'assurer que les requêtes clients utilisent le même nom que mon contrôleur, en omettant le terme « Controller » et en utilisant le camelCase. Par exemple, pour instancier « ActivityController », la requête émise par le client devrait avoir cette forme : « src/api.php/activity/... ».

Voici un exemple de requête vers le backend :

```
40  /**
41   * Requête qui retourne une activité sur base de son id.
42   */
43  function getActivity(id){
44      return $.ajax({
45          url: `src/api.php/activity/get?id=${id}`,
46          type: 'GET',
47          async: false,
48          dataType: "json",
49          success: function(response) {
50              return response;
51          },
52          error: function(error) {
53              console.log(error);
54              return error;
55          }
56      });
57  }
```

Figure 17 : exemple de requête vers le backend

5.6.5. Utilisation des endpoints

Dès lors, il fallut ajouter des points de terminaisons à mes contrôleurs afin de les utiliser. Pour cela, il suffisait d'ajouter, au sein de ceux-ci, autant de méthodes que nous voulions de point de terminaisons; chaque méthode ayant sa propre tâche retournant un résultat par http, le plus souvent au format JSON.

Comme pour l'instanciation des contrôleurs, le paramètre d'action dans l'url de la requête doit fournir exactement le même nom que celui de la méthode à exécuter dans le contrôleur (sensible à la casse). Par exemple, si je souhaite exécuter la méthode « update » de mon contrôleur « ActivityController », la requête devra prendre cette forme : « src/api.php/activity/update ». Ceci permettra d'instancier le contrôleur et d'en appeler sa méthode.

Voici un exemple d'envoi de réponse depuis le backend :

```
73  /**
74   * Récupère une activité sur base de son id.
75   * url : src/api.php/activity/get?id=${id}
76   */
77  public function get()
78  {
79      $id = Utils::getParam($_SERVER['REQUEST_URI']);
80      $activity = $this->activityRepository->find($id);
81      $this->setCharacteristics($activity);
82      if(isOwnerOf($activity)){
83          header('Content-Type: application/json');
84          echo json_encode($activity);
85      } else echo "You are not the owner of this activity";
86  }
```

Figure 18 : exemple de point de terminaison

5.6.6. Sécurisation de l'API

Afin que les utilisateurs ne puissent pas interagir avec l'API sans être authentifié, ou agir sur des données qui appartiennent à d'autres utilisateurs, j'utilise le matricule récupéré depuis l'authentification ULiège. Le matricule étant injecté directement dans le tableau `$_SESSION` de PHP, il n'est pas possible, à ma connaissance, de le corrompre depuis le navigateur. En effet, il aurait été judicieux d'utiliser des web tokens tels que les JWT¹³. Avec plus de temps, j'aurais aimé les intégrer à l'application.

Voici donc la méthode alternative utilisée :

```
88      /**
89      * Vérifie l'appartenance de l'activité à l'utilisateur courant.
90      */
91      public function isOwnerOf($activity){
92          return constant("USER") === $activity->userMatricule;
93      }
```

Figure 19 : méthode vérifiant l'identité de l'utilisateur

5.6.7. Fichier config.php

À la racine de mon API, j'ai fourni un fichier de configuration qui permet de stocker :

- Des constantes qui pourraient changer au fil du temps, et qui modifieraient le fonctionnement de l'application.
- Des données de test à utiliser pendant la phase de développement.
- Des constantes à modifier permettant de passer de la phase de développement à celle de mise en production.

¹³ JWT. Sur le site JWT. Consulté le 27/05/2023
<https://jwt.io/>

En voici le contenu :

```
src > config.php
1  <?php
2
3  namespace Api;
4
5  /**
6   * Matricules de test, pour changer : remplacer la constante DEV_USER
7   */
8  $studentMatricule = "s180022";
9  $adminMatricule = "u193459";
10 $friendMatricule = "f074100";
11
12 /**
13  * Constantes du projet
14  */
15 // Constantes de développement
16 // 2e paramètre à remplacer pour changer de type d'utilisateur
17 define("DEV_USER", $studentMatricule);
18 define("DEV_USER_MAIL", "l.bodet@student.helmo.be");
19
20 // Constantes de production
21 define("PROD_USER", $_SESSION["phpCAS"]["user"]);
22 define("PROD_USER_MAIL", $_SESSION["phpCAS"]["attributes"]["mail"]);
23
24 // Constantes à modifier si nécessaire
25 // Dans ce cas, à modifier si changement du système d'année académique
26 define("DEBUT_AA", ["month" => 9, "day" => 15]);
27 define("FIN_AA", ["month" => 8, "day" => 31]);
28
29 // Constantes à modifier si switch entre développement et production
30 // (remplacer "DEV" par "PROD" et vice versa)
31 define("USER", constant("DEV_USER"));
32 define("USER_MAIL", constant("DEV_USER_MAIL"));
33 ?>
```

Figure 20 : fichier de configuration du backend

5.6.8. Uniformisation des données étrangères à l'application

Un des défis majeurs de mon projet était d'intégrer des données provenant d'autres plateformes internes de l'entreprise au sein de l'application. C'était une première pour moi d'interagir avec plusieurs bases de données dans une même solution.

L'application est principalement axée sur la gestion des activités des étudiants, qui sont répertoriées sur différentes plateformes en fonction de leur type (ateliers, séjours et participations au conseil). Il a donc été nécessaire de récupérer les données pertinentes et de les convertir en activités au sein de mon application. Chaque champ récupéré d'une activité externe devait être associé à un champ d'activité tel que défini dans ma solution. De plus, j'ai dû standardiser le format du matricule des étudiants, qui pouvait varier en fonction de la plateforme : soit sous forme abrégée (une lettre suivie de six chiffres), soit numérique (huit chiffres).

En plus des activités, j'ai dû intégrer des données provenant de la plateforme CORALIS (Corporate Alumni Information System), qui gère tous les partenaires de HEC Liège. Ces données étaient nécessaires car une fonctionnalité de mon

application propose un mécanisme d'autocomplétion pour suggérer des organisateurs lors de l'édition ou création d'activités.

5.6.9. Avantages / désavantages de cette pratique

Développer sa propre API backend présente à la fois des avantages et des désavantages.

D'un côté, l'un des avantages de coder mon API sans framework est la flexibilité. En évitant de dépendre d'un framework spécifique, j'ai pu avoir un contrôle total sur l'architecture et la conception de mon application. J'ai pu adapter mon code selon mes besoins et éviter les limitations imposées par certains frameworks. Cette approche m'a permis d'avoir une plus grande liberté dans le choix que j'ai dû effectuer.

Un autre avantage est l'apprentissage approfondi des concepts sous-jacents à un framework. En développant mon API sans framework, j'ai dû comprendre les principes fondamentaux de la création d'une API backend tels que la gestion des routes, la manipulation des données et les opérations CRUD (Create, Read, Update, Delete). Cela m'a permis d'approfondir mes connaissances en programmation et de mieux comprendre le fonctionnement interne d'une API.

Cependant, il est important de mentionner les désavantages potentiels de cette approche. En l'occurrence, le développement sans framework peut être plus chronophage. En effet, les frameworks offrent souvent des fonctionnalités prêtes à l'emploi qui accélèrent le processus de développement. En développant chaque composant de mon API à partir de zéro, j'ai dû consacrer plus de temps et d'efforts au développement et à la maintenance de l'application.

En conclusion, coder une API backend sans utiliser de framework présente des avantages en termes de flexibilité et d'apprentissage approfondi mais peut également être plus chronophage et nécessiter une recherche plus intensive de solutions.

5.7. Création d'un store

Maintenant que toutes les données nécessaires au bon fonctionnement de l'application étaient disponibles, il fallait trouver un moyen de les propager vers la partie frontend.

Pour cela, en m'inspirant des frameworks avec lesquels j'ai travaillé, j'ai pensé qu'il serait judicieux d'implémenter un magasin de données. L'idée était d'éviter d'appeler des requêtes AJAX un peu partout dans l'application. J'ai préféré les regrouper dans un magasin de données que je pourrais rafraîchir au besoin et accéder de manière plus lisible.

Ce magasin de données se traduit par un dossier "store" à la racine du projet, dans lequel se trouve un fichier "store.js". Ce fichier contient un objet JavaScript dont les propriétés permettent de récupérer les données des requêtes AJAX.

5.8. Exports

Un des défis proposés par ce projet était l'export de différentes données; voici les différentes sections de mon projet qui ont dû être exportées :




- La liste d'activités d'un étudiant sous forme de PDF.
- Son profil de compétences sous forme de PDF.
- La liste d'inscriptions à un atelier de compétences sous forme de CSV.


L'objectif principal de cet outil est de permettre aux étudiants d'avoir une version palpable du compte-rendu de leurs expériences au cours de leur cursus. Ainsi, ils pourront les valoriser lors de leur arrivée dans le monde professionnel. Ensuite, il allait de soi de permettre aux administrateurs l'export des inscriptions dont ils sont responsables afin de leur faciliter la tâche.

Pour réaliser cela, j'ai d'abord dû trouver des librairies appropriées et apprendre à m'en servir. Heureusement, elles étaient accompagnées d'une dense documentation et un de mes collègues a pu m'aiguiller car il s'en était déjà servi. Une fois prises en main, il ne restait plus qu'à fournir un design de qualité.


Pour cela, j'ai plusieurs fois échangé avec une des personnes en charge du projet afin de fournir un export digne d'être partagé et valorisé. Cette personne m'a donc fourni la charte graphique de l'Université de Liège et j'ai pu m'inspirer de leurs codes afin de produire quelque chose d'officiel qui corresponde à leurs attentes et à l'identité de l'entreprise.

Voici, ci-dessous, le rendu des exports de la liste d'activités que j'ai fournie, sachant que je ne me limite ici qu'à une page de PDF afin de rester concis.





**Demo Test**
Demonstration of studentbook
[1 edited activity](#)

My activities

**Activity Example**
May 16th 2023 - May 31st 2023

General information
Type : Academic
Organized by : HEC Liège
Categories : Lifelong training; Hard skills;
Dimensions / Topics : International;

Skills
Strategy definition

Skill level : Awareness, knowledge, ability to remember and understand information
Comment : None

Quality and performance control

Skill level : Application, ability to use, to put into practice
Comment : None




Figure 21 : exemple d'activité exportée sous forme de PDF

Enfin, voici, ci-dessous, un exemple de profil de compétence exporté sous forme de PDF.

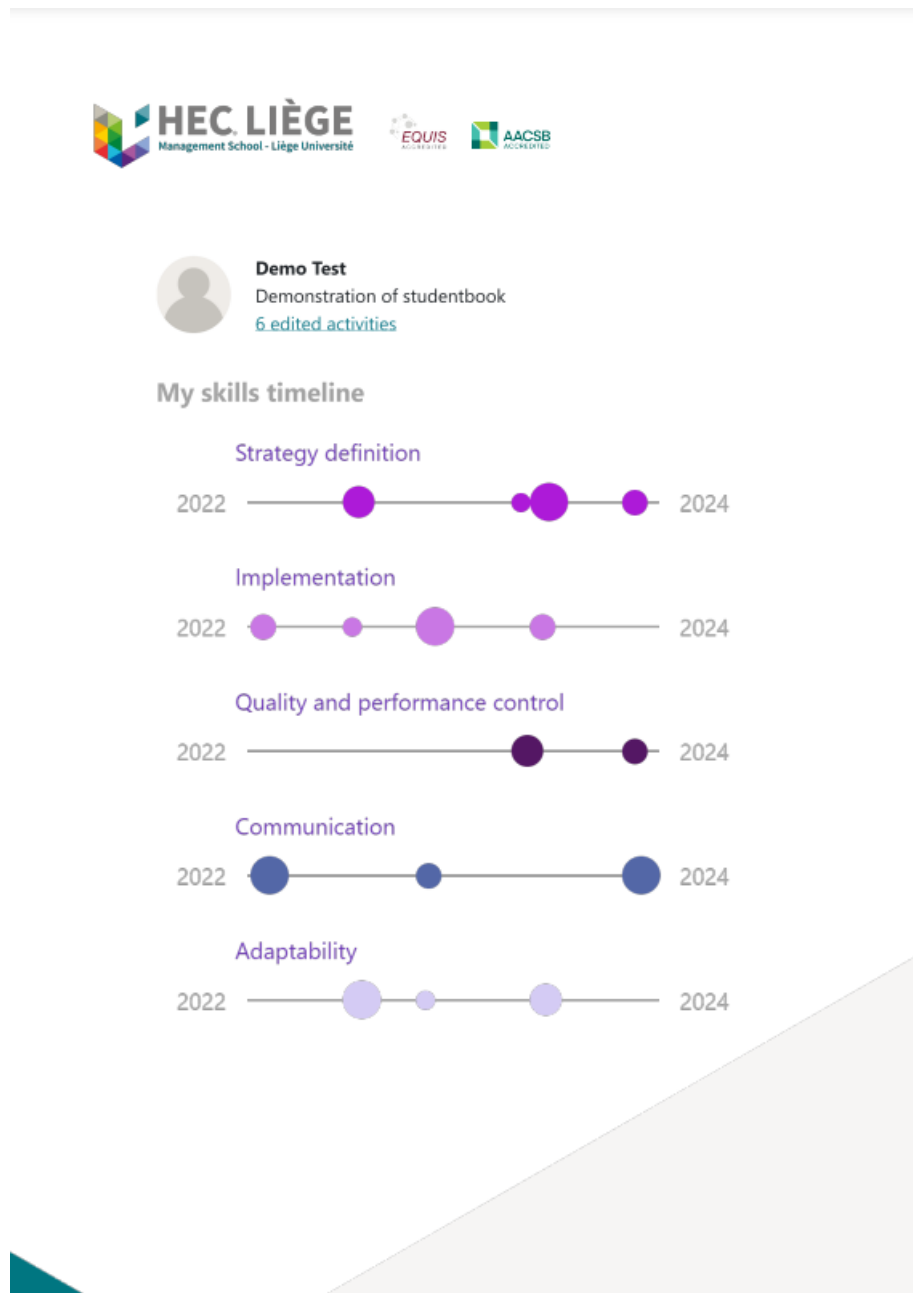


Figure 22 : exemple de profil de compétences exporté sous forme de PDF

5.9. Mise en production

Pendant la phase de développement, j'avais besoin de données pour tester la récupération d'activités à partir de diverses plateformes. J'ai donc effectué des requêtes SQL pour rechercher des étudiants ayant participé à tous les types d'activités. Avec les matricules ainsi récupérés, j'ai pu simuler l'utilisation de l'application par l'un de ces étudiants.

Lors de la mise en production, il suffit de remplacer le matricule de test par le matricule provenant de la session PHP dans le fichier config.php du backend.

5.10. Compte-rendu

L'implémentation de la solution que j'ai développée fut un succès selon les retours de l'entreprise. J'ai réussi à intégrer toutes les fonctionnalités requises dans l'application, et même à en ajouter certaines supplémentaires.

Pour l'instant, l'application n'a pas encore été publiée mais dès sa disponibilité, des statistiques sur son utilisation seront recueillies. Cela permettra d'évaluer son impact et de recueillir des informations précieuses sur la façon dont les utilisateurs interagissent avec elle. Ces données seront essentielles pour guider les futures améliorations et évolutions de l'application.

En parlant d'évolutions, il est important de souligner que l'application est destinée à évoluer en continu. De nombreuses autres fonctionnalités pourraient être ajoutées pour répondre aux besoins changeants des utilisateurs. Cette perspective ouvre la porte à de nouvelles opportunités d'innovation et de développement pour rendre l'application plus performante et utile.

Le processus de développement de cette application a été extrêmement enrichissant pour moi. J'ai pu mettre en pratique les différents aspects de la programmation que j'ai appris tout au long de mon parcours. Cette expérience m'a permis d'approfondir mes connaissances techniques et de renforcer mes compétences dans le domaine du développement d'applications.

Enfin, je suis vraiment heureux d'avoir pu contribuer à la création d'une application qui sera réellement utilisée. Cela donne un sentiment de satisfaction de savoir que mon travail aura un impact concret sur les utilisateurs, en leur offrant une solution pratique et efficace à leurs besoins.

Conclusion

En conclusion, le projet de développement de cette application par HEC Liège répond à un besoin réel des étudiants en offrant une plateforme pour consigner et valoriser leurs activités. Bien que l'application n'ait pas encore été lancée, il est prévu qu'elle permette aux étudiants de mieux mettre en valeur leurs compétences et expériences lors de leur recherche d'emploi, contribuant ainsi à renforcer leur employabilité. De plus, en centralisant les activités proposées par l'école, l'application devrait faciliter l'engagement des étudiants dans les activités qui les intéressent.

Il est important de noter que des ajustements pourraient être apportés au fonctionnement de l'application avant son lancement officiel. Il sera essentiel d'obtenir l'adhésion des étudiants à utiliser l'application et de promouvoir ses avantages de manière efficace. Des efforts supplémentaires pourraient être déployés pour sensibiliser et impliquer les étudiants afin d'assurer une adoption optimale de l'application et une participation active de leur part. Comme par exemple une mise en place de mécanismes d'évaluation et de suivi pour mesurer son impact réel sur leur parcours académique et leurs perspectives professionnelles.

En résumé, bien que l'application n'ait pas encore été lancée et que des améliorations supplémentaires puissent être apportées, les attentes résident dans sa capacité à permettre aux étudiants de valoriser leurs activités, à renforcer l'image de l'école et à favoriser une meilleure participation aux opportunités offertes.

En termes d'évolutions et d'améliorations futures, plusieurs perspectives prometteuses se dessinent pour cette application.

Tout d'abord, il serait envisageable d'intégrer les alumni de l'école dans l'utilisation de l'application. Cette extension permettrait aux anciens étudiants de continuer à tirer parti de la plateforme pour mettre en valeur leurs réalisations et expériences professionnelles post-diplôme. En facilitant le maintien de liens forts avec l'école, cette intégration renforcerait le réseau professionnel dynamique au sein de la communauté HEC Liège.

Par ailleurs, nous pourrions aussi imaginer la création d'une fonctionnalité de partage de conseils et mentorat. Les étudiants actuels pourraient ainsi bénéficier des connaissances et de l'expertise des alumni. Ces derniers pourraient prodiguer des conseils sur la recherche d'emploi, partager leurs expériences et offrir un soutien précieux dans différents domaines. Cette interaction intergénérationnelle favoriserait le développement professionnel des étudiants et renforcerait les liens entre les différentes promotions.

Une autre perspective intéressante consisterait à doter l'application d'un système intelligent de recommandations personnalisées. Grâce à une analyse pertinente des données, l'application pourrait suggérer aux étudiants des activités, opportunités de stages ou d'emplois, événements ou formations complémentaires correspondant à leurs intérêts et compétences. Cette fonctionnalité de recommandation personnalisée faciliterait la découverte de

nouvelles opportunités et encouragerait les étudiants à s'impliquer davantage dans des domaines qui les passionnent.

Enfin, afin de favoriser le réseautage professionnel, il serait envisageable d'intégrer des fonctionnalités de mise en relation entre les étudiants, les alumni et les entreprises partenaires de l'école. Celles-ci faciliteraient l'expansion du réseau professionnel des étudiants, leur permettant de découvrir des opportunités de carrière, d'établir des collaborations fructueuses et de renforcer leurs liens avec les acteurs clés du monde professionnel.

Ces perspectives d'évolutions et d'améliorations ouvrent de nouvelles opportunités pour renforcer la communauté éducative de HEC Liège, favoriser le développement professionnel continu des étudiants et créer des synergies fructueuses entre les différentes générations d'étudiants et d'anciens élèves.

En conclusion, mon expérience de travail sur ce projet a été à la fois enrichissante et gratifiante. J'ai eu l'opportunité de contribuer à la création d'une application innovante répondant à un réel besoin des étudiants de HEC Liège. Tout au long de mon stage, j'ai pu développer mes compétences techniques et mettre en pratique les connaissances acquises au cours de ma formation.

Sur le plan technique, l'entreprise m'a affirmé que j'ai relevé avec succès les défis qui se sont présentés à moi, en respectant les objectifs fixés dans mon cahier des charges. Néanmoins, avec davantage de temps, j'aurais pu explorer plus en profondeur certaines fonctionnalités et perfectionner certains aspects techniques. Cette auto-critique me pousse à continuer à me former et à chercher constamment à améliorer mes compétences dans le domaine du développement d'applications.

Du point de vue de mon intégration dans le milieu professionnel, j'ai été chaleureusement accueilli par l'équipe de HEC Liège. J'ai rapidement trouvé ma place au sein du groupe, bénéficiant de l'entraide et du soutien de mes collègues. Les interactions au sein de l'équipe ont ainsi favorisé un environnement de travail propice à la collaboration et à l'échange d'idées.

En somme, je suis très reconnaissant pour cette expérience enrichissante, car mon travail sur ce projet m'a permis de m'épanouir sur le plan professionnel en m'intégrant à une équipe et en contribuant à la réalisation d'une nouvelle application, tout en renforçant mes compétences techniques.