Travaux de Bachelor : Analyse et conception de la gestion des sujets

Travail de Bachelor réalisé en vue de l’obtention du Bachelor HES

par :

Angelo ARGELLI

Conseiller au travail de Bachelor :

David BILLARD, professeur HES

Genève, le 09.12.2020

Haute École de Gestion de Genève (HEG-GE)

Filière : Informatique de Gestion

Déclaration

Ce travail de Bachelor est réalisé dans le cadre de l’examen final de la Haute école de gestion de Genève, en vue de l’obtention du titre Bachelor of Science dans le domaine de l’Informatique de Gestion.

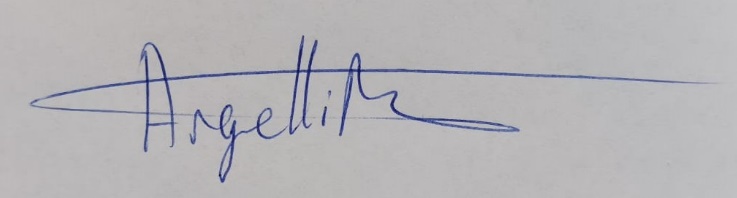
L’étudiant a envoyé ce document par e-mail à l'adresse remise par son conseiller au travail de Bachelor pour analyse par le logiciel de détection de plagiat URKUND, selon la procédure détaillée à l’URL suivante : [https://www.urkund.com](https://www.urkund.com/fr/student/) .

L’étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommandations formulées dans le travail de Bachelor, sans préjuger de leur valeur, n'engage ni la responsabilité de l'auteur, ni celle du conseiller au travail de Bachelor, du juré et de la HEG.

« J’atteste avoir réalisé seul le présent travail, sans avoir utilisé des sources autres que celles citées dans la bibliographie. »

Fait à Genève, le 09.12.2020

Angelo ARGELLI



Remerciements

En premier lieu, je tiens à remercier mon directeur de mémoire M. David Billard qui m’a suivi et conseillé tout au long de ce projet.

Je souhaite également remercier ma famille ainsi que mes amis (et collègues diplômés avant moi) pour leur soutien et leurs précieux conseils.

Ensuite, je remercie la direction de la HEG qui a eu l’indulgence de me laisser un délai supplémentaire de deux semaines après ma contamination au virus COVID-19 dont les symptômes m’ont empêché de travailler un certain temps et beaucoup affaibli.

Et enfin un immense merci à tous les enseignants, professeurs, et mentors qui m’ont enseigné l’informatique tout au long de ma formation, au CFPT ou à la HEG.

Résumé

Le but de ce travail est d’améliorer la gestion des sujets de travaux de bachelor, que ce soit du point de vue des étudiants ou du point de vue des professeurs qui les suivent.

L’objectif principal de ce travail est de trouver et de proposer une solution concrète d’amélioration de la gestion des sujets de bachelor.

Les points suivants seront abordés :

Une analyse du procédé actuel d’un étudiant pour trouver un sujet de travail de bachelor ainsi qu’un directeur de mémoire.

Une analyse du point de vue des professeurs sur ce procédé et du point de vue du secrétariat.

Cet état des lieux du fonctionnement des différentes parties établi, nous ferons une analyse qui permettra de soulever les problèmes de la procédure actuelle.

S’en suivra l’élaboration d’une solution informatique adaptée à la résolution des problèmes soulevés par l’analyse et des différents moyens de l’appliquer.

Enfin, la réalisation d’un prototype de plateforme informatique mettant en œuvre ces solutions.

Nous conclurons par une analyse du processus et du résultat obtenu afin de déterminer si les objectifs ont été atteints, à savoir, résoudre les problèmes soulevés par l’analyse.

Table des matières

[Déclaration i](#_Toc59192304)

[Remerciements ii](#_Toc59192305)

[Résumé iii](#_Toc59192306)

[Liste des tableaux vii](#_Toc59192307)

[Liste des figures vii](#_Toc59192308)

[1. Introduction 1](#_Toc59192309)

[2. Analyse de l’existant 2](#_Toc59192310)

[2.1 Point de vue d’un étudiant 2](#_Toc59192311)

[2.1.1 Documents à sa disposition 2](#_Toc59192312)

[2.1.2 Démarche 3](#_Toc59192313)

[2.2 Point de vue d’un directeur de mémoire 4](#_Toc59192314)

[2.2.1 Démarche 4](#_Toc59192315)

[2.3 Point de vue du secrétariat 5](#_Toc59192316)

[2.4 Synthèse d’analyse 5](#_Toc59192317)

[3. Conception d’une solution 7](#_Toc59192318)

[3.1 La forme 7](#_Toc59192319)

[3.1.1 Application de bureau 7](#_Toc59192320)

[3.1.2 Application Mobile 8](#_Toc59192321)

[3.1.3 L’application Web 8](#_Toc59192322)

[3.2 Les fonctionnalités 9](#_Toc59192323)

[3.2.1 La publication 9](#_Toc59192324)

[3.2.2 La consultation 9](#_Toc59192325)

[3.2.3 La communication 9](#_Toc59192326)

[3.2.4 La documentation 9](#_Toc59192327)

[4. Réalisation d’un prototype 10](#_Toc59192328)

[4.1 Technologies utilisées et Environnement de développement 10](#_Toc59192329)

[4.2 Principaux éléments 10](#_Toc59192330)

[4.2.1 Sujets 10](#_Toc59192331)

[4.2.2 Utilisateurs 11](#_Toc59192332)

[4.2.3 Filières 11](#_Toc59192333)

[4.3 Structure de l’application 12](#_Toc59192334)

[4.3.1 Connexion 12](#_Toc59192335)

[4.3.2 Navigation 13](#_Toc59192336)

[4.3.3 Liste des sujets – Index 14](#_Toc59192337)

[4.3.3.1 Affichage d’un sujet dans la liste 15](#_Toc59192338)

[4.3.3.2 Bouton de contact 15](#_Toc59192339)

[4.3.4 Ajout de sujet 16](#_Toc59192340)

[4.3.5 Liste de mes sujets 17](#_Toc59192341)

[4.3.6 Modifier un sujet 18](#_Toc59192342)

[4.3.7 Documents 19](#_Toc59192343)

[4.3.8 Gestion des sujets 19](#_Toc59192344)

[4.3.9 Connexion à la base de données 20](#_Toc59192345)

[4.3.10 Autres fonctions 21](#_Toc59192346)

[4.3.11 Structure de fichiers 21](#_Toc59192347)

[4.3.12 Les feuilles de style 21](#_Toc59192348)

[4.4 Modèles de données et diagrammes 22](#_Toc59192349)

[4.4.1 Diagramme 22](#_Toc59192350)

[4.4.2 Tables 22](#_Toc59192351)

[4.4.2.1 Table utilisateur 22](#_Toc59192352)

[4.4.2.2 Table personne 23](#_Toc59192353)

[4.4.2.3 Table sujet 24](#_Toc59192354)

[4.4.2.4 Table filière 24](#_Toc59192355)

[4.4.3 Données de Test 25](#_Toc59192356)

[4.4.4 Génération des données de test 25](#_Toc59192357)

[4.5 Vues 26](#_Toc59192358)

[4.5.1 Vue v\_user 26](#_Toc59192359)

[4.5.2 Vue v\_sujet 26](#_Toc59192360)

[4.5.3 Vue v\_personne 27](#_Toc59192361)

[4.5.4 Vue v\_edit\_sujet 28](#_Toc59192362)

[4.5.5 Vue v\_insert\_sujet 28](#_Toc59192363)

[5. Développement futur et Améliorations possibles 29](#_Toc59192364)

[5.1 Sécurité 29](#_Toc59192365)

[5.2 Déploiement 29](#_Toc59192366)

[5.3 Interface 29](#_Toc59192367)

[5.4 Efficience du code 29](#_Toc59192368)

[6. Conclusion 30](#_Toc59192369)

[Bibliographie 31](#_Toc59192370)

[Annexe 1 : Fichiers du prototype d’application web 32](#_Toc59192371)

Liste des tableaux

[Tableau 1 : Table utilisateurs (suj\_user) 23](#_Toc58423296)

[Tableau 2 : Table personnes (suj\_personne) 23](#_Toc58423297)

[Tableau 3 : Table sujets (suj\_sujet) 24](#_Toc58423298)

[Tableau 4 : Table Filières (suj\_filiere) 25](#_Toc58423299)

Liste des figures

[Figure 1 : Exemple de proposition de sujet à tous les étudiants 4](#_Toc58423314)

[Figure 2 : Maquette application de bureau 7](#_Toc58423315)

[Figure 3 : Maquette d’application Web 8](#_Toc58423316)

[Figure 4 : Logo PHP 10](file:///C:\Users\Naheu\Desktop\TB\TB_Angelo_Argelli.docx#_Toc58423317)

[Figure 5 : Page « Connexion » 12](file:///C:\Users\Naheu\Desktop\TB\TB_Angelo_Argelli.docx#_Toc58423318)

[Figure 6 : Comparaison de barre de navigation Admin-User 13](#_Toc58423319)

[Figure 7 : Comparaison des barres de recherche selon le point de vue 15](#_Toc58423320)

[Figure 8 : Page principale « Sujets de Travaux de Bachelor Proposés » 15](#_Toc58423321)

[Figure 9 : Page « Ajouter un sujet » 17](#_Toc58423322)

[Figure 10 : Page « Mes sujets » 17](#_Toc58423323)

[Figure 11 : Page « Modifier un Sujet » 18](#_Toc58423324)

[Figure 12 : Page « Gestion des Sujets » 20](#_Toc58423325)

[Figure 13 : Diagramme de Base de données 22](#_Toc58423326)

# Introduction

Il est souvent difficile pour un étudiant de trouver un sujet de travail de bachelor et un directeur de mémoire à qui le sujet parle pour le suivre.

C’est une épreuve par laquelle tous les étudiants aspirants à un Bachelor of Science passent ou sont passés, c’est pourquoi il est intéressant de tenter d’améliorer la gestion de ce procédé.

Cependant, l’étudiant n’est pas aussi seul qu’il ne le pense dans cette épreuve. C’est pourquoi dans ce travail je m’intéresserai également au point de vue des professeurs pouvant mettre à disposition des sujets et pas seulement aux étudiants en recherche.

D’abord, je proposerai une analyse de l’existant, afin d’établir un état des lieux et de déterminer comment fonctionnent actuellement les différents partis impliqués : les étudiants en recherche d’un sujet et/ou d’un directeur de mémoire, les professeurs pouvant suivre des étudiants en tant que directeur de mémoire et le secrétariat qui doit gérer les documents et enregistrer les différents sujets choisis par les étudiants.

Cette analyse permettra de soulever les problèmes de la procédure actuelle, problèmes auxquels il sera possible de trouver une solution dans la deuxième partie de ce travail, soit la conception d’une solution informatique. Celle-ci permettant de faciliter la mise en relation des trois parties et l’accès aux documents nécessaires aux étudiants pour avancer dans leur recherche de sujet. Cette partie inclut la modélisation des différents aspects de la solution, notamment les structures de données, le fonctionnement de la plateforme ou encore le choix d’un langage et d’une plateforme pour appliquer cette solution.

Enfin, la réalisation d’un prototype de cette solution incluant le développement et la documentation du code.

L’objectif principal de ce travail est donc de trouver et de proposer une solution concrète d’amélioration de la gestion des sujets de bachelor.

# Analyse de l’existant

## Point de vue d’un étudiant

### Documents à sa disposition

Les étudiants disposent de documents relatifs au Travail de bachelor pour les orienter. Ces différents documents sont disponibles en accès libre sur plusieurs plateformes, aux adresses suivantes :

* La fiche modulaire :
  + [https://www.hesge.ch/heg/formation-base/bachelors-science/[filière]/plan-des-modules](https://www.hesge.ch/heg/formation-base/bachelors-science/%5bfilière%5d/plan-des-modules) et y chercher le module de Travail de Bachelor (en anglais sous program > Bachelor Thesis pour la filière IBM).
* Les directives du travail de bachelor :
  + (IG) Directives directement mentionnées sur la fiche modulaire
  + (ID) <https://www.hesge.ch/heg/etudier-heg/cursus-academique/travaux>
  + (EE) <https://cyberlearn.hes-so.ch/enrol/index.php?id=16290>
  + (IBM) Sur un cours cyberlearn uniquement accessible aux membres de la filière.
* La convention du travail de bachelor (pour la filière IG) :
  + <https://www.hesge.ch/heg/etudier-heg/cursus-academique/travaux>
* Formulaire de proposition de sujet (pour la filière ID) :
  + <https://www.hesge.ch/heg/etudier-heg/cursus-academique/travaux>
* La charte graphique sous la forme d’un chablon de document WORD :
  + [intranet.hesge.ch/intranet/enseignementrecherche/formation-de-base](https://intranet.hesge.ch/intranet/enseignementrecherche/formation-de-base)
* Un guide de rédaction de bibliographie :
  + <https://www.hesge.ch/heg/infotheque/citations-et-references-bibliographiques>

### Démarche

Obtention des documents

De lui-même ou en demandant de l’aide, un étudiant prêt à commencer son travail de bachelor doit trouver les documents le concernant parmi tous ceux cités ci-dessus. On peut noter que ces différents documents sont disponibles à trois endroits différents du site de la HEG et que le lien de la charte graphique n’est disponible qu’à travers la fiche modulaire. De plus, les filières IBM et EE doivent consulter un cours supplémentaire sur cyberlearn pour accéder au reste des instructions.

Définition de son idée

Une fois qu’il a rassemblé les documents le concernant et qu’il en a pris connaissance, un étudiant peut se lancer dans la recherche d’un sujet de travail de bachelor. Il peut bien évidement s’en choisir un lui-même s’il a des idées qui peuvent convenir, ou bien partir à la recherche d’un directeur de mémoire qui pourra lui proposer un sujet.

Directeur de mémoire

Une fois qu’il a un sujet qui lui plait, l’étudiant peut le proposer à un professeur, par   
e-mail ou en personne, pour voir s’il est disponible pour le suivre et si le sujet est suffisant ou s’il convient de le modifier.

Sinon, l’élève peut faire la démarche inverse et aller voir les professeurs qu’il aimerait avoir comme directeur de mémoire sans idée de projet et leur demander s’ils ont, eux, des projets disponibles et du temps pour les suivre.

Occasionnellement, le secrétariat peut envoyer à tous les étudiants des e-mails d’idées de sujets de travail de bachelor susceptibles de les intéresser, ainsi que la personne à contacter en cas d’intérêt pour le sujet de la part d’un étudiant.

Figure 1 : Exemple de proposition de sujet à tous les étudiants



(Mail de Tuba Uygun, 2020 / capturé par Angelo Argelli, 2020)

## Point de vue d’un directeur de mémoire

De manière générale, un directeur de mémoire est sollicité par les étudiants et non l’inverse. Même s’il a souvent des projets sous la main, il ne les propose qu’aux étudiants faisant la démarche de lui demander ou avec qui il a eu l’occasion d’en parler spécifiquement. Sinon, les étudiants n’ont aucun moyen de savoir qu’un directeur de mémoire a des projets disponibles.

### Démarche

Si un professeur a un projet qu’il aimerait faire réaliser par un étudiant dans le cadre de son travail de bachelor, il n’a pas réellement besoin de partir à la recherche d’un étudiant car ce n’est pas son rôle. À moins qu’il ne discute du projet avec un étudiant dans un cadre autre, il devra attendre qu’il étudiant ne le contacte, en personne ou par e-mail, pour en parler, remplir la convention de travail de bachelor et voir ce projet se réaliser.

S’il sait que chaque année beaucoup d’étudiants lui demandent son suivi, il peut choisir de simplement attendre que des étudiants viennent à lui et proposer un sujet spécifique à un étudiant à qui il pense que le sujet peut convenir.

Si le projet lui tient particulièrement à cœur, il peut choisir d’envoyer un e-mail aux étudiants pour le proposer à tous et s’assurer que l’un d’entre eux se porte volontaire.

Si le mandant d’un projet est un externe à la HEG, il devra passer par le secrétariat ou par un professeur pour faire connaître son projet aux étudiants par le biais d’un e-mail.

## Point de vue du secrétariat

Le secrétariat de la filière s’occupe de recevoir les conventions de travail de bachelor et d’enregistrer les sujets attribués aux étudiants, leur directeur de mémoire ainsi que les dates de rendu dans un document unique, lui aussi rendu public. Ce dernier contient la liste des sujets de travaux de bachelor en cours et est accessible au même endroit que la convention de travail de bachelor (<https://www.hesge.ch/heg/etudier-heg/cursus-academique/travaux>).

Dans le cas où une personne voudrait faire connaître un projet aux étudiants, c’est aussi le secrétariat qui s’occupe d’envoyer un e-mail groupé aux étudiants concernés potentiellement intéressés.

## Synthèse d’analyse

On constate plusieurs problématiques auxquelles on peut facilement trouver une solution :

* L’accès aux documents est réparti sur différentes plateformes et les liens sont difficiles à trouver
* La mise en relation des étudiants avec des professeurs pouvant devenir leurs directeurs de mémoire est quasi non-existante
* La mise à disposition de l’information de l’existence des projets de sujets disponibles est très variable

On constate qu’un étudiant prêt à faire son travail de bachelor est très livré à lui-même. Ce qui est bien normal, l’un des buts principaux du travail de bachelor étant de « démontrer sa capacité à mener à bien un projet dans son intégralité, c’est-à-dire depuis le choix d’un sujet jusqu’à la défense de son travail, en respectant des contraintes temporelles. » comme le stipule le document « Directive générale pour la réalisation du Travail de Bachelor ». On peut alors argumenter que le but du travail de bachelor est de prouver la capacité d’un étudiant à mener à bien un projet de A à Z et que l’organisation est loin d’être parfaite dans chaque situation dans la vie. Alors, la difficulté à regrouper tous les documents nécessaires au Travail de Bachelor serait un petit obstacle prouvant cette capacité d’autonomie. Mais une plateforme commune permettrait à tous de n’avoir qu’un seul endroit à regarder.

De même, les méthodes actuelles de communication sur la disponibilité de sujets de travaux de bachelor sont moindres : des mails occasionnels et des discussions.

Être capable à tout moment de savoir quels sont les projets disponibles des professeurs grâce à cette plateforme permettrait à beaucoup d’étudiants de saisir des occasions de projets desquels ils n’auraient même pas entendu parler car ils sont noyés dans un flot d’e-mails quotidiens.

Les étudiants pourraient y retrouver tous leurs documents et les sujets disponibles proposés par des enseignants. Les enseignants pourraient y proposer leurs sujets directement sans passer par des e-mails. Le secrétariat n’aurait plus besoin de relayer d’e-mails non plus pour faire connaître des sujets mais les publierait directement sur cette plateforme commune.

Une plateforme telle n’existe pas à ce jour et permettrait de standardiser le processus hasardeux qu’est le choix d’un sujet de travail de bachelor tout en résolvant les trois points soulevés dans cette analyse.

Ce qu’il faut c’est une plateforme de rencontre pour sujets de travail de bachelor.

# Conception d’une solution

## La forme

Le besoin d’une plateforme de rencontre entre les étudiants, les mandants et les directeurs de mémoire étant bien présent, il faut identifier la forme qu’elle pourrait prendre. Penchons-nous sur trois formes les plus communes d’applications :

* L’application de bureau
* L’application mobile
* L’application web

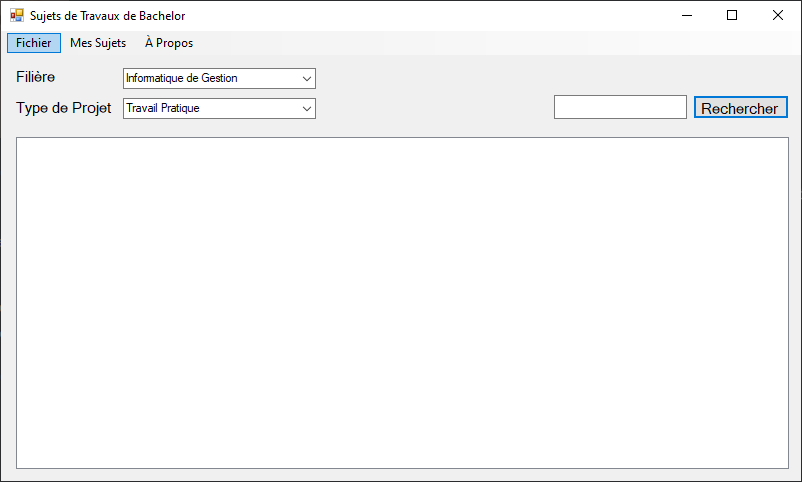
Il faut aussi déterminer quelles sont les qualités importantes que cette plateforme devra avoir pour qu’elle soit utilisée par les étudiants. Pour une telle population, qui est toujours connectée grâce aux smartphones et qui a une capacité de temps d’attention de plus en plus réduite par les stimuli quotidiens toujours plus nombreux, les qualités les plus importantes sont, selon moi, la facilité d’accès et la simplicité.

### Application de bureau

Une première option serait une application de bureau standard. L’avantage en serait la possibilité de la déployer sur tous les postes de la HEG. Mais les étudiants ont besoin, comme souligné précédemment, d’une solution plus accessible. S’ils font leur travail de bachelor après avoir validé tous leurs modules, ils ne reviendront pas forcément souvent sur les postes de la HEG. Pour convenir à des étudiants, cette plateforme se doit d’être disponible hors de la HEG.

Une application de bureau ne peut être envisagée que pour la partie gestion et validation des idées de sujets du côté du secrétariat, mais son accès restreint aux postes de la HEG rendrait son utilisation par les étudiants très rare.

Figure 2 : Maquette application de bureau

  
(Angelo Argelli, 2020)

### Application Mobile

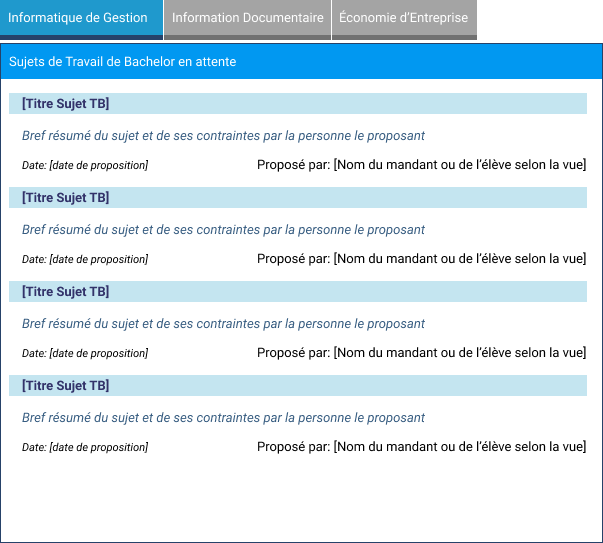
L’accès de plus en plus courant, d’autant plus auprès d’une population d’étudiants, à des smartphones oblige à se poser la question de la pertinence d’une application mobile. Contrairement à l’application de bureau, l’application mobile serait téléchargeable et donc accessible partout. Cependant, la télécharger rien que pour consulter les sujets de travaux de bachelor ou proposer son idée puis la désinstaller, est plutôt contraignant.

Une application mobile de ce genre a donc une disponibilité moins restreinte qu’une application de bureau, puisqu’elle est téléchargeable sur mobile, mais offre un accès quasi identique à l’application de bureau. Si l’étudiant n’est plus obligé de se rendre sur place pour utiliser un poste HEG, il est tout de même obligé d’installer sur son espace privé une application à l’utilité temporaire.

### L’application Web

Tout naturellement, l’application web est la solution la plus versatile dans son accès. Sur mobile ou sur un ordinateur (de la HEG ou personnel), l’utilisateur pourrait y accéder d’où il veut. Elle pourrait incorporer les fonctions de recherche de sujet, de proposition de sujets et aussi de gestion des propositions de sujets par l’administration. Elle est le juste milieu entre les deux options précédemment citées et pourrait facilement être intégrée à des services web déjà existants comme par exemple le portail des étudiants « age » ou le site de la HEG.

Figure 3 : Maquette d’application Web



(Angelo Argelli, maquette faite sur [www.figma.com](http://www.figma.com))

## Les fonctionnalités

Maintenant que la forme a été choisie, nous pouvons en venir au cœur du sujet : les fonctionnalités de cette plateforme. De quoi avons-nous besoin pour répondre aux exigences du processus et résoudre ses problèmes ?

### La publication

La plateforme devra bien évidemment permettre aux étudiants de publier les sujets qu’ils souhaitent faire et permettre aux professeurs d’en faire de même. La publication devra être simple et rapide pour permettre de communiquer son idée de façon claire et concise en un minimum de place.

### La consultation

Il faudra aussi que l’utilisateur, qu’il soit professeur ou étudiant, puisse accéder aux différents sujets disponibles pour lui, et ce sur une interface épurée, sans être noyé d’informations. L’utilisateur devra pouvoir rechercher et filtrer ce qui l’intéresse.

### La communication

D’une façon ou d’une autre, il faudra aussi mettre en contact les deux interlocuteurs pour qu’ils sachent que l’un est intéressé par le projet de l’autre.

Cela peut être par un système de chat en direct ou par des messages interposés. La plateforme pourrait aussi être une sorte de forum de discussions ou les différents utilisateurs pourraient proposer des pistes de développement aux autres.

### La documentation

Nous l’avons relevé plusieurs fois, l’accès aux documents est un problème par son inconsistance et son manque de clarté, tantôt sur une plateforme tantôt sur une autre, et pas identiques selon les filières. La nouvelle plateforme ne pourra pas uniformiser les différentes gestions choisies par les filières. Ce qu’elle pourra faire, toutefois, c’est de proposer aux étudiants un seul et unique endroit ou retrouver tous les documents nécessaires.

# Réalisation d’un prototype

## Technologies utilisées et environnement de développement

Figure 4 : Logo PHP

Pour me lancer dans le développement de ce projet, j’ai choisi le PHP. J’ai choisi ce langage plutôt qu’un autre pour sa versatilité et pour l’expérience que j’ai avec. Malgré sa popularité en déclin, il est simple d’utilisation et plus rapide d’exécution que d’autres alternatives comme le Python, particulièrement pour de petites applications.

De plus, utiliser PHP m’a permis de gagner le temps que j’aurais utilisé à apprendre un nouveau langage.

J’ai utilisé comme serveur de développement l’application EasyPhp Devserver 17.0 pour faire tourner PHP dans sa version 7.1.3.

Et pour la partie base de données, j’ai utilisé l’application MySQL Workbench 7.0 qui faisait tourner MySQL v 8.0.19.

Pour l’écriture du code j’ai utilisé l’éditeur de texte ATOM v 1.53.0 et directement l’interface de MySQL Workbench pour la base de données.

## Principaux éléments

### Sujets

Les sujets sont le point central de l’application destinée à leur gestion.  
Un sujet est défini par 7 attributs principaux :

* Son propriétaire
* La filière que le sujet concerne
* Son statut
* Son titre
* Sa description
* Son type
* La date de sa dernière mise à jour

Le propriétaire d’un sujet est celui, en plus de l’administrateur, qui sera amené à le gérer. Il peut à tout moment le supprimer ou le modifier à sa guise. Un sujet peut être proposé par n’importe quel utilisateur.

La filière que le sujet concerne déterminera avec quels autres sujets il sera affiché.

Le statut d’un sujet est d’abord « en attente » avant d’être validé ou non par un administrateur et de passer au statut « accepté » ou « refusé ». Une fois que le propriétaire d’un sujet a trouvé preneur et n’a plus besoin de le faire connaître, il peut le rendre indisponible en le faisant passer au statut « pris ». Les sujets proposés par les étudiants sont les seuls à devoir passer un processus de validation. Ceux proposés par des professeurs sont directement acceptés, partant du principe qu’il en va de leur responsabilité de proposer un sujet adéquat.

Le titre est comme son nom l’indique le titre choisi pour le sujet, simple et concis.

La description est un texte court allant jusqu’à 1000 caractères, servant à se faire une idée sur le sujet proposé, son contexte, ses limites, quelques détails importants avant de le choisir.

Le type d’un sujet désigne la nature du travail à fournir. Il peut être de type « travail pratique », « travail de recherche » ou « travail hybride » s’il comporte une part des deux.

La date correspond à la dernière mise à jour du sujet. S’il n’a pas encore été modifié, il s’agit de sa date de création.

### Utilisateurs

L’application Web fonctionne de plusieurs points de vue différents afin que tous ne puissent accéder qu’à ce qui les concerne. Il y a donc un point de vue administrateur, qui peut tout voir et gérer, et un point de vue utilisateur, séparé en mode professeur ou en mode étudiant. Les étudiants ne voyant que les sujets proposés par des professeurs et vice-versa.

### Filières

Pour une meilleure lisibilité, les sujets sont séparés selon la filière qu’ils concernent. Ce sont évidemment les quatre filières de la HEG : Informatique de Gestion, Information Documentaire, Economie d’Entreprise et International Business Management. Cependant, n’importe quel utilisateur peut proposer un sujet pour n’importe quelle filière ou bien proposer un sujet qui ne concerne pas de filière particulière.

## Structure de l’application

### Connexion

Cette page correspond au fichier login.php.

C’est ce que l’on peut attendre de n’importe quelle page de connexion. L’utilisateur entre son identifiant et son mot de passe dons un formulaire et aura accès au reste de l’application s’il entre la bonne combinaison. Toutes les autres pages sont sécurisées de manière à ne pas pouvoir les atteindre si l’utilisateur n’est pas connecté et à le rediriger vers la page de connexion.

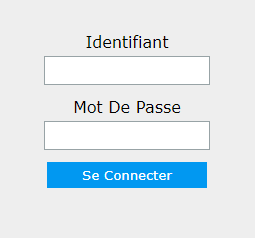


Figure 5 : Page « Connexion »

Une fois le bouton appuyé, le formulaire en méthode POST rafraichit la page. Avant d’effectuer le traitement de la connexion, les champs sont vérifiés, en plus de la validation par l’attribut HTML « required ». Le traitement de la connexion est autorisé uniquement si les valeurs des champs du $\_POST correspondant aux différents inputs ne sont pas vides.

Après la vérification de la présence de données dans les champs, le traitement de la connexion se fait en deux temps. D’abord une requête est faite sur la vue v\_user pour savoir s’il existe une entrée dans la base de données possédant le nom d’utilisateur saisi. Si la requête ne trouve pas de correspondance, le traitement s’arrête là et un message d’erreur est affiché indiquant que le nom d’utilisateur entré est erroné.

Dans un second temps, si le nom d’utilisateur existe, le mot de passe saisi est « hashé » avec la fonction PHP md5() puis comparé au hash de mot de passe enregistré dans la base de données pour le nom d’utilisateur entré. Ici aussi, si la requête ne trouve pas de correspondance entre le mot de passe entré et celui stocké, le traitement est arrêté et un message d’erreur est affiché en indiquant que l’utilisateur a saisi une mauvaise combinaison identifiant/mot de passe.

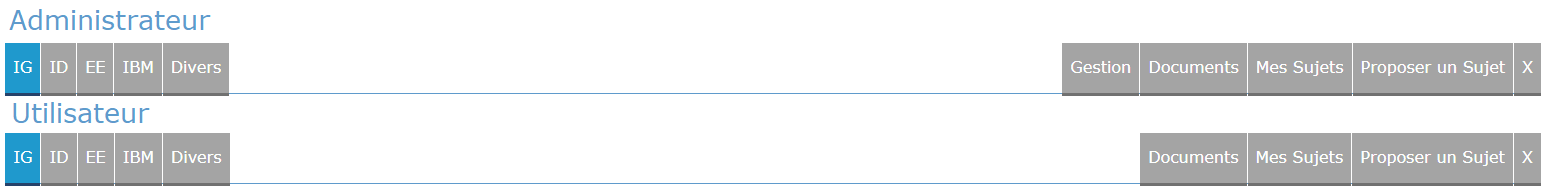
Si le mot de passe entré est correct, une session est ouverte et les données d’utilisateur sont stockés dans le $\_SESSION. Parmi ces données se trouvent l’id d’utilisateur et le type d’utilisateur mais aussi l’id de personne, l’id de la filière à laquelle la personne est affiliée et le type de personne. Ces données sont utilisées pour la suite du traitement sur les autres pages.

Il n’y a pas la possibilité de créer un compte utilisateur et ce pour plusieurs raisons. D’abord, le contexte de l’application étant la proposition et la recherche de sujets de travaux de bachelor, si le projet est effectivement déployé, les comptes devraient être préexistants, créés par l’administration s’ils sont propres à la plateforme ou reliés aux identifiants AAI. Ensuite, si cette possibilité existait, n’importe qui, réellement affilié à la HEG ou non aurait accès à toutes ces données sans vérification d’identité.

### Navigation

La navigation entre les pages, à une exception près, se fait par le biais d’une barre de navigation située en haut de l’interface. Elle est visible sur toutes les pages une fois l’utilisateur connecté et est composée d’un élément par lien vers une page de la plateforme.

Figure 6 : Comparaison de barre de navigation Admin-User



L’onglet actif est toujours affiché en bleu et les autres en gris pour que l’utilisateur puisse savoir à tout moment sur quelle page il se trouve.

Tout à droite de la barre de navigation se trouve également un bouton supplémentaire permettant de se déconnecter et de revenir à la page de connexion. L’utiliser redirige vers un script de destruction de session sur le fichier « logout.php » qui lui renvoie sur la page de connexion.

Hormis la page de connexion, la seule page qui ne se trouve pas sur la barre de navigation est la page de modification d’un sujet à laquelle on accède uniquement par le biais de la page « Mes Sujets ». Aussi, le bouton menant à la page « Gestion des sujets » n’est généré que si le compte actif dans la $\_SESSION est de type « administrateur ».

### Liste des sujets – Index

Une fois la connexion validée, l’utilisateur est dirigé sur la page principale de l’application. Il s’agit de la liste des sujets proposés. Quel que soit le point de vue de l’utilisateur (administrateur, professeur ou étudiant), la liste qu’il voit en arrivant est celle des sujets concernant la filière à laquelle il est affilié selon son enregistrement dans la base de données et peut naviguer entre les onglets correspondants aux différentes filières à sa guise. Il existe 5 onglets différents, 4 correspondants aux filières et un servant pour les sujets divers qui ne concernent pas une filière particulière. Ils sont en fait une seule et même page. Simplement, une requête différente est effectuée selon la valeur du paramètre $\_GET[‘filiere’] représentant l’id de la filière affichée. Ce paramètre servant de filtre déterminant de quelle filière l’on souhaite afficher les sujets affiliés et donc l’onglet virtuel qui est sélectionné. Si un utilisateur venait à supprimer cette variable dans l’URL, il serait redirigé vers l’onglet de la filière à laquelle il est affilié.

Sur cette page, l’utilisateur a également accès à une barre de recherche et de filtres pour affiner les résultats qu’il souhaite afficher. Un utilisateur de type étudiant pourra trier le sujet par type de travail et effectuer une recherche par mot clé. Un utilisateur de type professeur ou administrateur pourra également trier selon le statut du travail, étant donné qu’il a accès à tous les sujets indépendamment du statut contrairement à l’étudiant. Par défaut, tous les sujets accessibles par l’utilisateur sont visibles et les champs de recherche sont vides.

Une fois le bouton « Rechercher » validé, le formulaire de méthode GET englobant la barre de recherche est validé et la page rechargée avec donc les différents champs de saisie dans des paramètres GET de l’URL. Respectivement, rType contenant le type de travail, rStatus contenant son statut et rText contenant la recherche textuelle.

La recherche fonctionne en modulant la requête SQL à laquelle s’ajoute une condition après le WHERE afin de correspondre à ce qui a été choisi par l’utilisateur. Respectivement, les champs de type SELECT ajoutent un test pour filtrer exactement les types et statuts recherchés et le champ texte ajoute un test LIKE qui recherche les mots clés dans les titres et les descriptions des sujets. Ainsi, toutes les combinaisons de recherche sont possibles.

Figure 7 : Comparaison des barres de recherche selon le point de vue

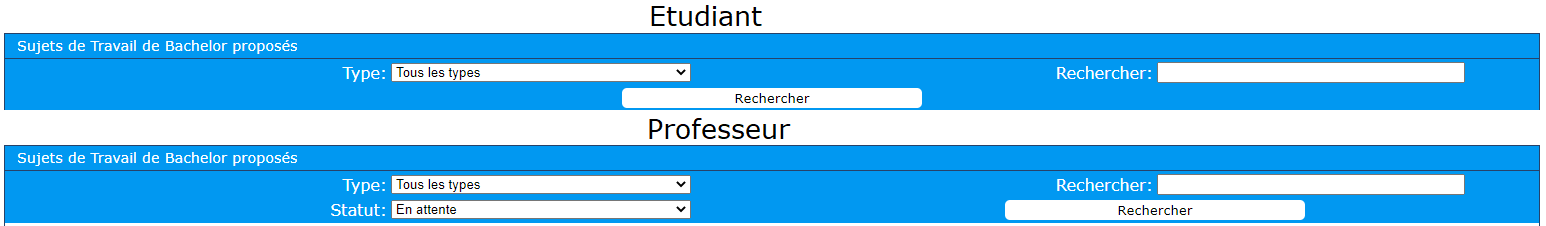
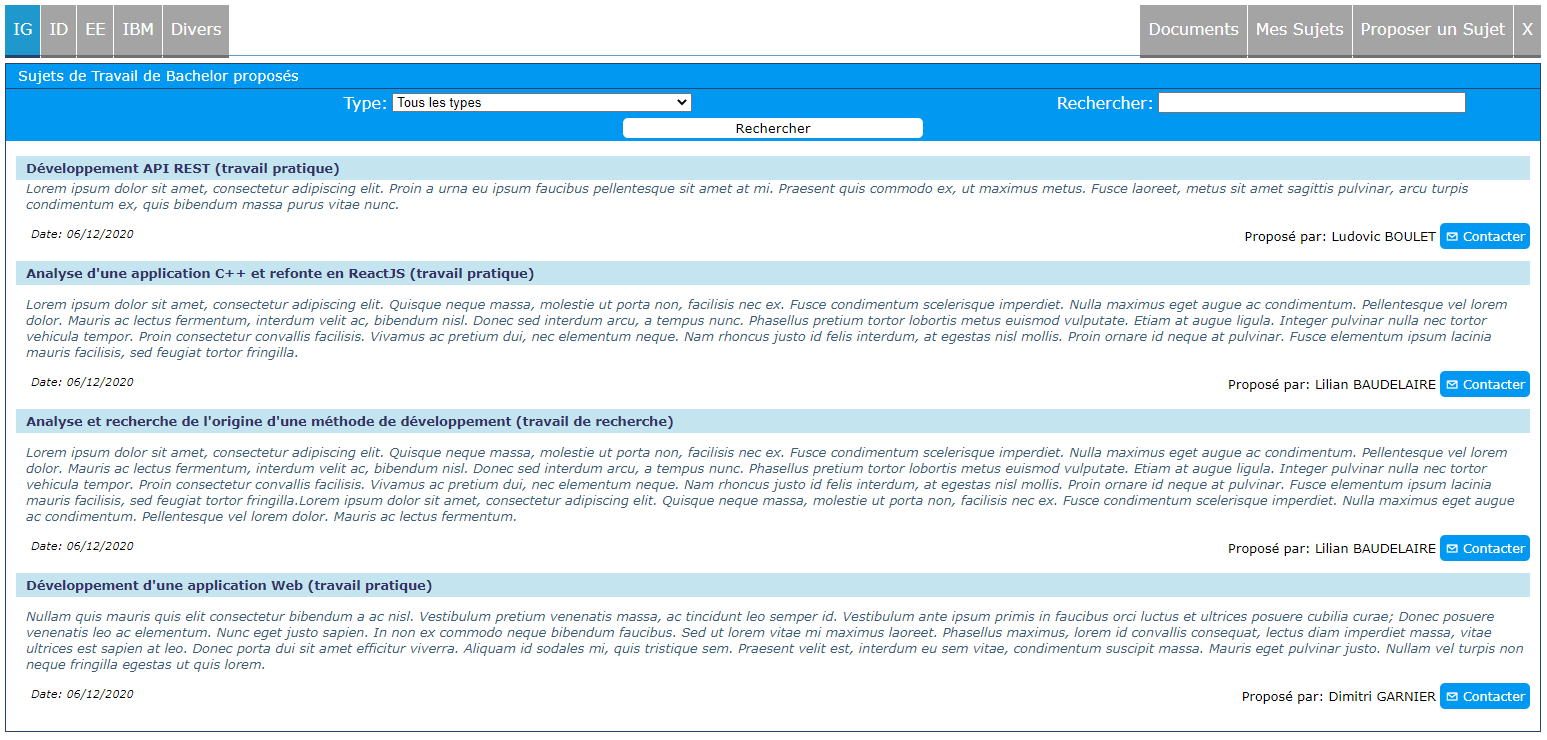


Figure 8 : Page principale « Sujets de Travaux de Bachelor Proposés »



#### Affichage d’un sujet dans la liste

Sur cette page on retrouve donc, pour chaque enregistrement de sujet retourné par la requête de recherche, un élément de liste affichant certaines de ses informations. Quel que soit le type d’utilisateur, s’y retrouvent, dans une barre bleue, le titre, avec entre parenthèses, le type de travail dont il s’agit. Au centre de l’objet de liste se trouve la description. En bas de l’objet, la date de dernière modification ainsi que le nom de la personne qui l’a proposé et un bouton de contact.

Dans le cas où l’utilisateur est un administrateur ou un professeur, il a le droit de voir tous les statuts. Alors, le statut de chaque type est également indiqué tout à droite de la barre de titre du sujet.

#### Bouton de contact

Pour chaque sujet, un bouton permet d’envoyer un mail à la personne concernée pour prendre contact directement avec elle. En cliquant dessus, on ouvre un lien de type « mailto » à l’adresse de la personne à contacter.

C’est l’alternative la plus simple que j’ai choisi. On aurait pu par exemple mettre en place un système de chat en ligne directement sur la plateforme. Mais en y réfléchissant, ça ne change pas grand-chose. Car même si les utilisateurs pouvaient communiquer par un système de chat direct, ils devraient être notifiés d’une certaine manière de la réception d’un message. Et comme cette application est une application web, les notifications auraient dû se faire par mail. On en reviendrait donc au même point.

### Ajout de sujet

La page d’ajout de sujet est accessible depuis toutes les pages de l’application par le lien dans la barre de navigation intitulé « Proposer un Sujet ». Tous les utilisateurs pouvant y accéder, la seule restriction d’accès est le simple fait d’être connecté.

La page propose un formulaire de méthode POST permettant de rentrer les informations du sujet à proposer : la filière, le type de travail, le titre du sujet ainsi que sa description. Une fois le formulaire validé, le sujet proposé peut être retrouvé dans la page « Mes Sujets » (voir plus loin).

Par défaut, les champs de sélection de type de travail et de filière concernée ont une valeur vide « -- Sélectionner une option -- » qui ne peut pas être resélectionnée par erreur.

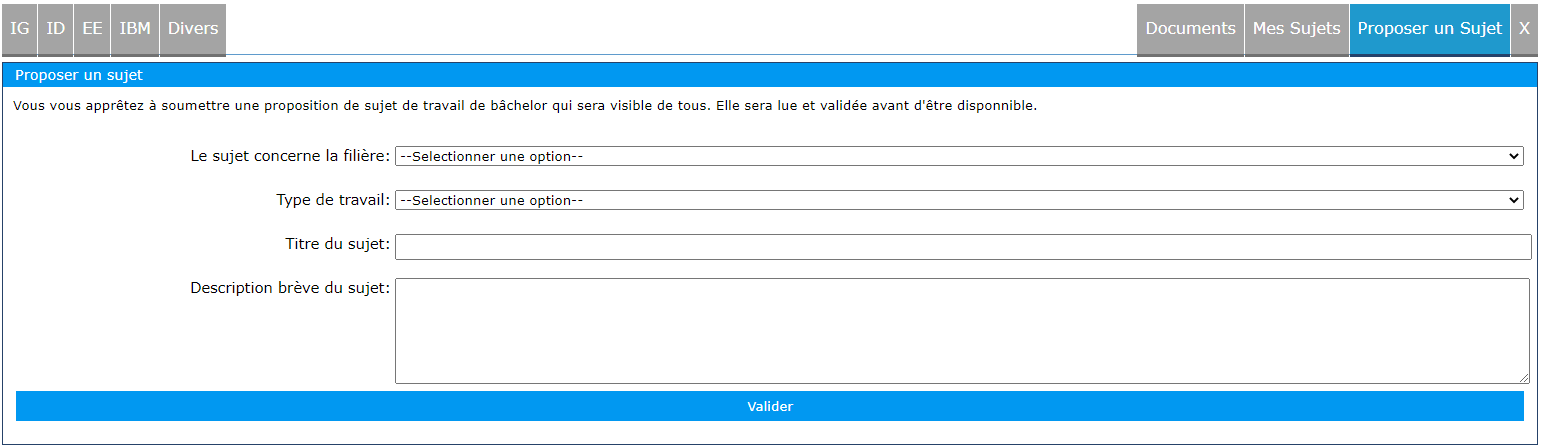
Le formulaire devrait, en principe être impossible à valider sans avoir rempli tous les champs, la mention HTML « required » étant attribuée à chaque champ. Dans le cas où quelqu’un viendrait à la retirer en inspectant le code, une seconde couche de sécurité en PHP viendra vérifier que tous les champs du $\_POST correspondant aux champs de texte ne soit vide et qu’aucune des sélections déroulantes ne soit nulle. Dans le cas d’une tentative de ce genre, un message d’erreur apparait indiquant une erreur de formulaire.

Une fois le formulaire validé, une requête INSERT est faite avec les informations rentrées, la date NOW() en SQL et le statut.

Le statut dépend du type de personne proposant un sujet. S’il s’agit d’un étudiant, il est « en attente » jusqu’à ce qu’un administrateur l’ait lu et validé. Si c’est un professeur, il est directement « accepté ».

Une fois le nouveau sujet inséré dans la table, par le biais d’une vue, l’utilisateur est redirigé vers la page « Mes sujets ».

Figure 9 : Page « Ajouter un sujet »



### Liste de mes sujets

La liste des sujets proposés par un utilisateur est elle aussi une page dont l’accès se fait depuis n’importe quel autre endroit du site dont la seule restriction d’accès est d’être un utilisateur connecté.

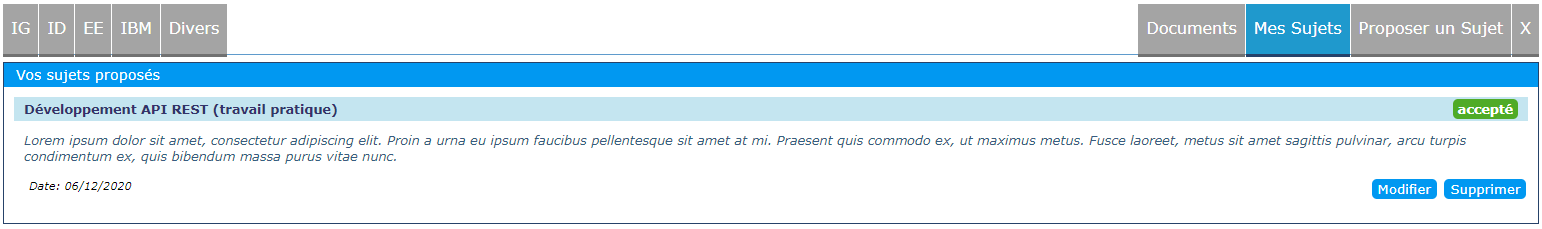
Elle propose une liste similaire à celle de la liste principale des sujets, à l’exception de l’affichage du statut, qui ici est disponible à tous les utilisateurs car les sujets affichés les concernent. Ainsi un utilisateur peut voir si son sujet est accepté ou toujours en attente.

Sont également présents des boutons permettant la modification ou la suppression de l’un des sujets.

Les boutons sont des liens qui utilisent l’identifiant d’un sujet dans leur valeur puis utilisent le $\_POST pour déterminer la fonction désirée. La valeur d’un bouton étant facilement accessible, une sécurité est en place utilisant la fonction isOwner() pour comparer l’id de l’utilisateur actif stocké dans la $\_SESSION avec l’identifiant de personne rattaché au sujet dont l’id est utilisé. Ceci afin de s’assurer que l’utilisateur essayant de modifier ou de supprimer un sujet soit bien propriétaire du sujet ou un administrateur.

Si l’action est de supprimer, elle est faite directement par cette page. Si au contraire l’utilisateur veut modifier le sujet, il est redirigé vers la page « Modifier un de sujet » avec l’id du sujet envoyé en $\_GET

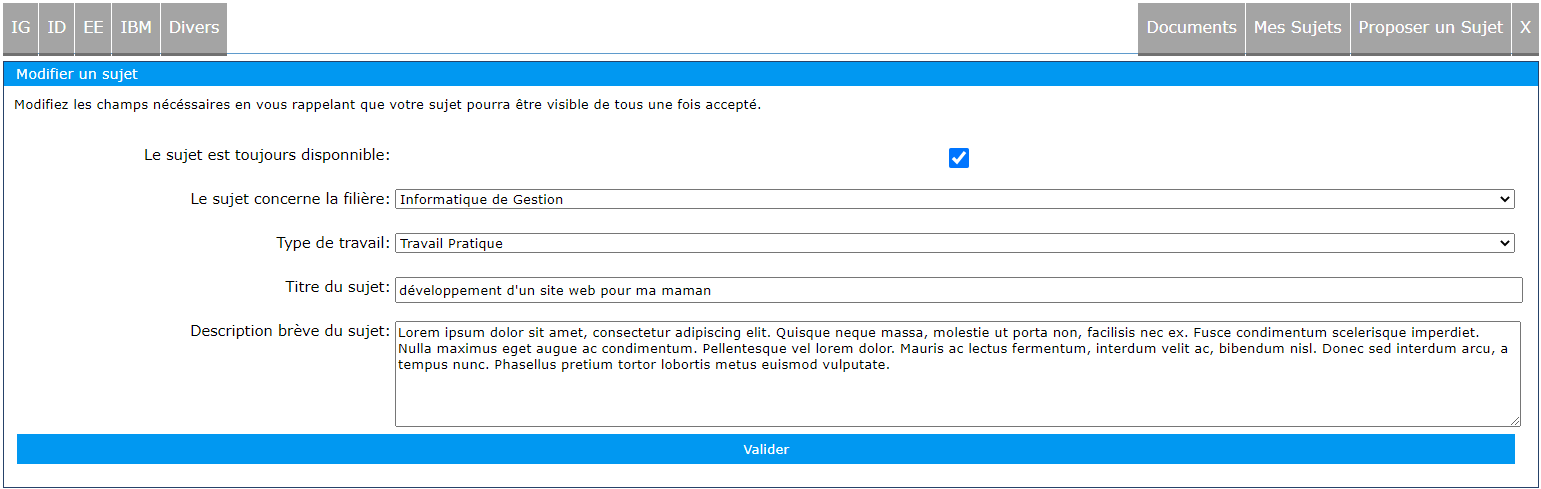
Figure 10 : Page « Mes sujets »



### Modifier un sujet

La page de modification des sujets est sensiblement identique dans son apparence à celle d’ajout d’un sujet. Elle est uniquement accessible par les boutons « Modifier » de chaque sujet sur la page « Mes Sujets ». La seule différence d’interface est la présence d’une case à cocher permettant de définir si le sujet est toujours disponible ou non.

Figure 11 : Page « Modifier un Sujet »



Le lien de cette page utilisant l’identifiant de sujet transmis par une variable $\_GET pour le cibler, une vérification est faite avant l’accès à la page, comme pour l’utilisation des boutons de la page « Mes Sujets » afin de s’assurer que l’utilisateur essayant d’accéder à la page est bien le bon et que l’identifiant n’a pas été remplacé par un autre dans l’URL. Il en est de même à la validation du formulaire.

Comme pour le formulaire d’ajout, ce formulaire ne peut pas être validé si certains de ces champs sont vides. La vérification est faite de la même manière, avec des attributs « required » couplés à une vérification du contenu du $\_POST.

Une fois le bouton « Valider » cliqué et les données validées, l’enregistrement correspondant à l’id est UPDATE par le biais d’une vue avec les nouvelles données entrés par l’utilisateur. La date est également modifiée avec un NOW() en SQL.

S’il s’agit d’un étudiant, le sujet, une fois modifié, sera de nouveau placé « en attente », indépendamment de son statut précédent, sauf bien sûr si la modification vise à changer le statut pour indiquer qu’il n’est plus disponible. Dans ce cas, le sujet prendra le statut « pris ». Ceci permet donc à un étudiant dont le sujet a été refusé de revenir dessus pour le faire examiner à nouveau.

### Documents

La page de documents, elle aussi, est disponible pour tous les utilisateurs par la barre de navigation. L’objectif de cette page est de rassembler tous les liens utiles permettant d’accéder rapidement aux documents d’informations et d’instructions nécessaires aux étudiants. Pour l’instant, afin d’être certains de garder accès aux documents, il s’agit de simples liens dirigeant vers les pages contenant les documents et non des liens vers les documents en eux-mêmes car ils sont susceptibles de changer de nom ou d’emplacement ou même d’être changés. Mais on peut imaginer un futur ou les documents seraient en accès direct depuis cette page par défaut et où c’est à cette page que les autres feraient référence pour l’accès aux documents d’instructions.

### Gestion des sujets

La page de gestion des sujets est celle qui réunit le plus d’informations sur les différents sujets et qui permet d’effectuer le plus d’actions. Elle sert de panneau de commande pour les comptes administrateurs dont la tâche est de gérer les propositions de sujets.

Son accès se fait par un nouveau bouton sur la gauche de celui des documents, visible uniquement quand l’utilisateur actif est un administrateur.

De la même manière que pour les autres pages dont l’accès est restreint, une vérification est faite pour s’assurer que l’utilisateur dont l’id est stocké dans la $\_SESSION est bien administrateur avant de le laisser accéder à la page. Même si en principe un utilisateur autre ne doit pas voir le bouton sur la barre de navigation, l’application n’est jamais à l’abri de quelqu’un qui a deviné ou aperçu quelque part le lien. Il en est de même pour la validation de chaque action effectuée sur cette page.

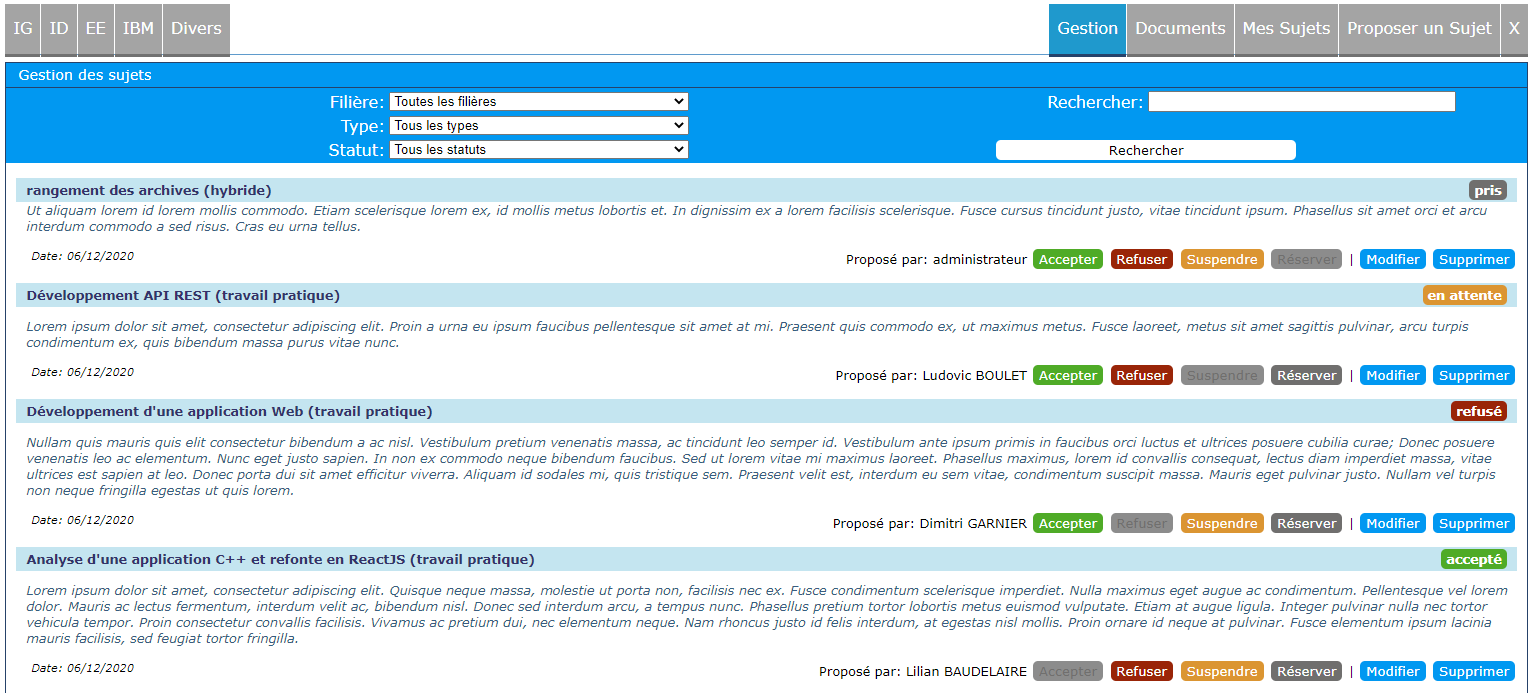
Ici l’administrateur a à sa disposition une liste de tous les sujets et peut les filtrer selon leur statut, leur filière, leur type, ou encore faire une recherche textuelle, comme sur la page de liste principale.

Pour chacun des sujets de la liste l’administrateur a une vue semblable à celle de la page « Mes sujets » avec donc une indication du statut et un accès aux boutons « Modifier » et « Supprimer ». En plus de cela, il a des boutons permettant de changer directement le statut des sujets.

Ces nouveaux boutons sont tous les quatre affichés mais celui du statut actuel du sujet est grisé et inutilisable. Leur utilisation est similaire à celle des boutons « Modifier » et « Supprimer ».

Chacun d’entre eux utilise un attribut « name » différent mais une « value » identique, celle de l’id du sujet à altérer. L’utilisation de l’un de ces boutons provoque la validation d’un formulaire de type POST rafraichissant la page. Si l’utilisation de l’un des boutons est confirmée par sa présence dans le $\_POST, une dernière vérification de droits est faite avant d’envoyer un UPDATE avec le statut correspondant.

Figure 12 : Page « Gestion des Sujets »



### Connexion à la base de données

Chacune des pages de l’application faisant une requête à la base de données fait appel à la même fonction du fichier « fonctions.php », la fonction connect().

Elle crée et retourne un PDO pour se connecter à la base de données sur le serveur MySQL. Elle utilise des identifiants stockés sur un fichier db.ini hors du dossier racine de l’application web et donc inaccessible aux utilisateurs.

Pour le développement du prototype, le serveur de données tournait à l’adresse 127.0.0.1:3306. Si déployé autrement, l’adresse à changer se trouve dans la fonction connect().

Chaque requête, peu importe sa nature, utilise la fonction PDO->prepare() permettant de préparer la requête en s’assurant qu’elle ne contienne pas d’injections SQL.

Voici les données contenues dans le fichier db.ini, utilisées pour le développement du prototype.

username=TBAdministrator  
password=InDev2020  
db=tb\_sujets\_db

### Autres fonctions

Le fichier « fonctions.php » permet de regrouper les fonctions récurrentes dans l’application. On peut y retrouver pour l’instant la fonction connect(), vue plus haut, et la fonction isOwner.

isOwner($personId, $subjectId) a pour but de faire une requête à la base de données afin de vérifier une correspondance du $personId donné avec l’identifiant de personne enregistré dans l’enregistrement du sujet correspondant au $subjectId donné. Elle retourne une valeur booléenne correspondant au résultat de ce test.

### Structure de fichiers

Les sont mis en place dans la structure suivante :

Prototype

├───db

│ db.ini 🡨 Fichier d’identifiants de bdd

│

└───public 🡨 Racine du serveur HTTP

│ addSubject.php

│ documents.php

│ editSubject.php

│ functions.php

│ index.php

│ login.php

│ logout.php

│ mySubjects.php

│ subjectsManager.php

│

└───style 🡨 Dossier des feuilles de style

documents.css

formSubject.css

index.css

login.css

mySubjects.css

searchBar.css

style.css

Si cette structure n’est pas respectée pour des tests supplémentaires sur le prototype, des erreurs de chemins d’accès pourraient se produire.

### Les feuilles de style

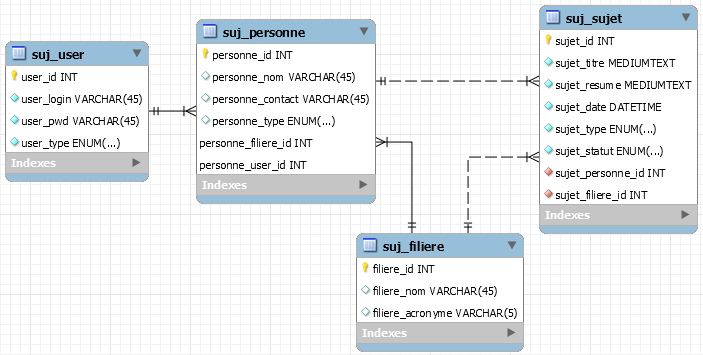
Presque chaque fichier php possède sa propre feuille de style pour les éléments qui lui sont propres. La plupart des fichiers php reprennent également les fiches index.css et style.css qui contiennent la plupart des styles récurrents. Pour plus de détails, voir les liens <link rel="stylesheet"> dans les balises <head> de chaque fichier php, tous disponibles sur le Github en Annexe 1.

## Modèles de données et diagrammes

### Diagramme

Voici le diagramme de la base de données de l’application web généré sur MySQL Workbench. Présentant les tables et leurs relations.

Figure 13 : Diagramme de Base de données



Bref résumé des relations établies : un utilisateur (suj\_user) est relié à une seule personne (suj\_personne) et inversement, donc les tables sont en relation 1,1. Une personne peut proposer plusieurs sujets et un sujet n’est proposé que par une seule personne, donc sa relation avec la table sujet (suj\_sujet) est de type 1,n. Une personne n’est affiliée qu’à une seule filière et une filière regroupe plusieurs personnes, la relation entre la table personne (suj\_personne) et la table filière (suj\_filière) est de 1,n. Il en va de même pour les sujets, qui ne concernent qu’une filière ou aucune filière. Cependant, la relation est de 1,n pour à des fins de facilitation de recherche. La table filière contenant un enregistrement représentant « toutes les filières » pour signifier non pas qu’un sujet ne concerne personne, mais concerne toutes les filières.

### Tables

#### Table utilisateur

La table utilisateurs contient les données de connexion ainsi que le type d’utilisateur. Les champs de login et de mot de passe sont utilisés pour la connexion et le type d’utilisateur détermine les droits du compte sur la gestion des sujets et donc son accès à la modification de données dont il n’est pas auteur. Les mots de passes ne sont pas stockés en clair mais c’est leur hash md5 qui est stocké. À terme, cette table pourrait être modifiée ou remplacée pour relier l’application à un système de connexion autre comme le login AAI.

Tableau 1 : Table utilisateurs (suj\_user)

|  |  |  |
| --- | --- | --- |
| **Champ** | **Type** | **Description** |
| user\_id | Int | Identifiant et clé primaire de la table |
| user\_login | Varchar(45) | Identifiant de connexion à saisir |
| user\_pwd | Varchar(45) | Hash md5 de mot de passe |
| user\_type | ENUM(‘user’,’admin’) | ‘user’ si le compte n’a pas les droits d’administration, ‘admin’ si il les a |

#### Table personne

La table personne contient les données des personnes associées à des comptes utilisateurs. Le nom et le prénom sont utilisés pour afficher la personne ayant proposé un sujet dans l’affichage des sujets. L’adresse e-mail est utilisée pour le bouton contact. Le champ type de personne est présent sur cette table et n’est pas une extension du type d’utilisateur pour deux raisons. Il ne désigne pas les droits d’un compte mais bien le point de vue d’une personne sur le processus de choix d’un sujet : étudiant ou professeur.

Tableau 2 : Table personnes (suj\_personne)

|  |  |  |
| --- | --- | --- |
| **Champ** | **Type** | **Description** |
| personne\_id | Int | Identifiant et clé primaire de la table |
| personne\_nom | Varchar(45) | Nom et prénom de la personne |
| personne\_contact | Varchar(45) | Adresse e-mail de la personne |
| personne\_type | ENUM(‘professeur’, ’étudiant’) | ‘professeur’ s’il est du côté des directeurs de mémoire proposant des sujets ou ‘étudiant’ si il est un étudiant |
| personne\_filiere\_id | Int | Clé étrangère correspondant à la filière affiliée à la personne |
| personne\_user\_id | int | Clé étrangère correspondant au compte utilisateur relié à la personne |

#### Table sujet

La table des sujets contenant toutes les informations concernant les sujets.

Tableau 3 : Table sujets (suj\_sujet)

|  |  |  |
| --- | --- | --- |
| **Champ** | **Type** | **Description** |
| sujet\_id | Int | Identifiant et clé primaire de la table |
| sujet\_titre | mediumtext | Titre du sujet |
| sujet\_resume | mediumtext | Description du sujet |
| sujet\_date | Datetime | Date de dernière mise à jour du sujet |
| sujet\_type | ENUM  (‘travail pratique’, ‘travail de recherche’, ‘travail hybride’) | Type de travail désignant sa nature |
| sujet\_statut | ENUM(‘en attente’, ‘accepté’, ‘refusé’, ‘pris’) | Statut du sujet, déterminant parfois son affichage |
| sujet\_personne\_id | Int | Clé étrangère correspondant à la personne ayant proposé le sujet. |
| sujet\_filiere\_id | Int | Clé étrangère correspondant à la filière concernée par le sujet |

#### Table filière

La table filière contient toutes les informations nécessaires à l’affichage des différentes filières d’études de la HEG. Leur nom ainsi que leur acronyme correspondant. À noter l’existence d’une filière à l’identifiant 0 qui représente toutes les filières dans les cas où une personne ou un sujet ne serait affilié à aucune en particulier.

Tableau 4 : Table Filières (suj\_filiere)

|  |  |  |
| --- | --- | --- |
| **Champ** | **Type** | **Description** |
| filiere\_id | Int | Identifiant et clé primaire de la table |
| filiere\_nom | Varchar(45) | Nom de la filière |
| filiere\_acronyme | Varchar(5) | Acronyme du nom de la filière |

### Données de Test

Voici quelques exemples de données de test à utiliser dans le prototype.

|  |  |  |
| --- | --- | --- |
| Identifiant | Mot de passe | Type de compte |
| administrateur | pwd | Administrateur |
| LudovicBOULET | profPassword | Professeur |
| ÉmilienneGAUTHIER | etuPassword | Étudiant |

Les mots de passe de tous les comptes professeurs sont identiques, de même pour les étudiants.

### Génération des données de test

Pour éviter d’utiliser des noms réels, j’ai généré des données d’utilisateurs fictives à l’aide du site internet <https://www.fantasynamegenerators.com/french_names.php> avec lesquels j’ai pu remplir ma base de données.

J’ai aussi utilisé le site internet <https://www.lipsum.com/> pour générer du texte aléatoire pour prendre la place des descriptions de sujets.

## Vues

À des fins de sécurité, pour préserver l’intégrité de leurs données, les tables ne sont évidemment pas directement accessibles par l’application. Leurs données sont relayées par des vues avec des droits différents selon la situation.

### Vue v\_user

Elle relaye l’entièreté des données de la table suj\_user en lecture seule peu importe qui y accède afin d’éviter toute altération des données de la table.

CREATE

ALGORITHM = UNDEFINED

DEFINER = `TBAdministrator`@`localhost`

SQL SECURITY DEFINER

VIEW `v\_user` AS

SELECT

`suj\_user`.`user\_id` AS `id`,

`suj\_user`.`user\_login` AS `login`,

`suj\_user`.`user\_pwd` AS `pwd`,

`suj\_user`.`user\_type` AS `type`

FROM

`suj\_user`

### Vue v\_sujet

Cette vue est la vue utilisée pour appeler une liste des sujets à afficher. Que ce soit pour la page principale affichant les résultats de recherche, la page « Mes Sujets » ou la page de « Gestion des sujets » et la page « Modifier un Sujet », toutes les données concernant un sujet récupérées par un SELECT viennent de cette vue qui est en lecture seule.

Elle contient aussi des données jointes d’autres tables. Les données de la personne ayant proposé le sujet y sont aussi représentées ainsi que les données de la filière que le sujet concerne. Ceci pour faciliter l’affichage en liste en faisant une seule requête.

### Vue v\_personne

CREATE

ALGORITHM = UNDEFINED

DEFINER = `TBAdministrator`@`localhost`

SQL SECURITY DEFINER

VIEW `v\_sujet` AS

SELECT

`suj\_sujet`.`sujet\_id` AS `id`,

`suj\_sujet`.`sujet\_titre` AS `titre`,

`suj\_sujet`.`sujet\_resume` AS `resume`,

DATE\_FORMAT(`suj\_sujet`.`sujet\_date`, '%d/%m/%Y') AS `date`,

`suj\_sujet`.`sujet\_type` AS `typeSujet`,

`suj\_sujet`.`sujet\_statut` AS `statut`,

`suj\_personne`.`personne\_id` AS `idPersonne`,

`suj\_personne`.`personne\_nom` AS `nomPersonne`,

`suj\_personne`.`personne\_type` AS `typePersonne`,

`suj\_personne`.`personne\_contact` AS `contactPersonne`,

`filierepersonne`.`filiere\_acronyme` AS `filierePersonne`,

`filieresujet`.`filiere\_id` AS `idFiliere`,

`filieresujet`.`filiere\_acronyme` AS `acrFiliere`,

`filieresujet`.`filiere\_nom` AS `nomFiliere`

FROM

(((`suj\_sujet`

JOIN `suj\_personne` ON ((`suj\_personne`.`personne\_id` = `suj\_sujet`.`sujet\_personne\_id`)))

JOIN `suj\_filiere` `filieresujet` ON ((`filieresujet`.`filiere\_id` = `suj\_sujet`.`sujet\_filiere\_id`)))

JOIN `suj\_filiere` `filierepersonne` ON ((`suj\_personne`.`personne\_filiere\_id` = `filierepersonne`.`filiere\_id`)))

Cette vue comprend les données de la table suj\_personne en lecture seule pour y accéder au besoin. Elle est notamment utilisée par la page de connexion quand elle enregistre les données de l’utilisateur connecté.

CREATE

ALGORITHM = UNDEFINED

DEFINER = `TBAdministrator`@`localhost`

SQL SECURITY DEFINER

VIEW `v\_personne` AS

SELECT

`suj\_personne`.`personne\_id` AS `idPersonne`,

`suj\_personne`.`personne\_nom` AS `nom`,

`suj\_personne`.`personne\_contact` AS `contact`,

`suj\_personne`.`personne\_type` AS `type`,

`suj\_personne`.`personne\_filiere\_id` AS `idFiliere`,

`suj\_personne`.`personne\_user\_id` AS `userId`

FROM

`suj\_personne`

### Vue v\_edit\_sujet

C’est une vue utilisée pour les modifications de la table suj\_sujet. Elle est disponible en écriture et est utilisée par les pages altérant le sujet : « Mes sujets », « Modifier un Sujet » et « Gestion des Sujets ».

CREATE

ALGORITHM = UNDEFINED

DEFINER = `TBAdministrator`@`localhost`

SQL SECURITY DEFINER

VIEW `v\_edit\_sujet` AS

SELECT

`suj\_sujet`.`sujet\_id` AS `id`,

`suj\_sujet`.`sujet\_titre` AS `titre`,

`suj\_sujet`.`sujet\_resume` AS `description`,

`suj\_sujet`.`sujet\_date` AS `date`,

`suj\_sujet`.`sujet\_type` AS `type`,

`suj\_sujet`.`sujet\_statut` AS `statut`,

`suj\_sujet`.`sujet\_personne\_id` AS `idPersonne`,

`suj\_sujet`.`sujet\_filiere\_id` AS `idFiliere`

FROM

`suj\_sujet`

### Vue v\_insert\_sujet

C’est la sœur de la vue v\_edit\_sujet. Elle est disponible pour l’insertion de nouvelles données, utilisée par la page « Ajouter un Sujet »

CREATE

ALGORITHM = UNDEFINED

DEFINER = `TBAdministrator`@`localhost`

SQL SECURITY DEFINER

VIEW `v\_inserting\_sujet` AS

SELECT

`suj\_sujet`.`sujet\_id` AS `id`,

`suj\_sujet`.`sujet\_titre` AS `titre`,

`suj\_sujet`.`sujet\_resume` AS `resume`,

`suj\_sujet`.`sujet\_date` AS `date`,

`suj\_sujet`.`sujet\_type` AS `type`,

`suj\_sujet`.`sujet\_statut` AS `statut`,

`suj\_sujet`.`sujet\_personne\_id` AS `idPersonne`,

`suj\_sujet`.`sujet\_filiere\_id` AS `idFiliere`

FROM

`suj\_sujet`

# Développement futur et Améliorations possibles

Ce prototype, dans l’éventualité où il serait adopté par la HEG, devrait être retravaillé sur plusieurs points. Par exemple en utilisant un framework dont la structure serait moins vulnérable que celle créée pour le développement du prototype.

## Sécurité

Les mesures de sécurité mises en œuvre dans le développement du prototype sont moindres et certainement contournables assez facilement. L’usage par exemple du framework Laravel permettrait une meilleure sécurité. En effet, le framework fournit de nombreuses fonctionnalités de sécurité, couvrant, l’authentification, l’autorisation, la vérification d’e-mails, le cryptage et hashage et autres.

De plus, si comme mentionné précédemment, le système de login est relié aux identifiants AAI des étudiants, il faudrait complètement remplacer le système de connexion par cette méthode.

## Déploiement

Si on veut déployer cette application, il serait bien de la relier à une plateforme comme AGE ou le site de la HEG. Dans le cas de AGE, elle deviendrait un nouvel onglet et la partie connexion sauterait complètement, étant donné la nécessité d’être connecté pour accéder à AGE. Cela faciliterait également l’accès aux données des personnes, déjà connectées.

## Interface

Un effort serait à faire sur la clarté de l’interface, qui sert son but mais n’est pas bien poussée. Une interface plus moderne serait de mise, et si l’application est intégrée à une autre plateforme, il serait nécessaire d’accorder l’interface sur celle de la plateforme existante pour ne pas dénoter.

## Efficience du code

Le code, encore une fois, sert son but mais n’est pas bien optimisé, du PHP simple pour une plateforme ou l’utilisateur sera mené à faire des recherches multiples n’est pas le plus efficace. Peut-être faudrait-il coupler ce PHP avec un autre langage dynamique ne nécessitant pas un rechargement de page pour accéder aux données de la recherche comme par exemple du JavaScript.

# Conclusion

Tout au long de ce projet, nous avons appris bien des choses. Nous avons découvert et analysé le processus complet de choix d’un sujet de travail de bachelor, du point de vue de trois parties : l’étudiant, le professeur et le secrétariat. De cette analyse nous avons relevé trois problématiques.

* L’accès aux documents difficile
* La mise en relation des étudiants et des professeurs
* L’accès à l’information de sujets disponibles par les étudiants

La solution proposée par ce travail répond à ces trois problématiques. Le prototype fournit un accès groupé aux documents, garantissant aux étudiants de tous les trouver du premier coup. Il est une plateforme de rencontre entre des personnes proposant des sujets et des personnes en recherche de sujets. L’information est donc facilement disponible par les étudiants, par une recherche pouvant être affinée par filière, type de travail et mots clés. Mais également aux professeurs, qui voient aussi les sujets proposés par les étudiants. La plateforme propose un lien direct vers un envoi de mail à la personne concernée par un sujet, réalisant la mise en relation entre les deux parties intéressées.

Les problématiques relevées lors de l’analyse sont donc toutes trois résolues et le prototype fournit une nouvelle plateforme, inexistante auparavant pour faciliter le processus de choix d’un sujet de travail de bachelor.

Bibliographie

HIGGINS Sean, 2020. *How to build a Secure Login Page in PHP*. In : GitConnected [en ligne] [Consulté le 10.10.2020] Disponible à l’adresse : <https://levelup.gitconnected.com/how-to-build-a-secure-login-page-in-php-954f51d08701>

BROTHERTON Claire, 2020. *Les frameworks PHP les plus populaires à utiliser en 2020*. In : Kinsta [en ligne] [Consulté le 1.12.2020] Disponible à l’adresse : <https://kinsta.com/fr/blog/frameworks-php>

LARAVEL, 2020. *Laravel Homestead.* In : Laravel [en ligne] [Consulté le 1.12.2020] Disponible à l’adresse : <https://laravel.com/docs/7.x/homestead>

Annexe 1 : Fichiers du prototype d’application web

Vous trouverez sur Github les fichiers de l’application prototype, les fichiers de base de données et les figures.

<https://github.com/Naheufan/tb_Angelo_Argelli>