

Travail de Bachelor 2018

Skool



Étudiant : Julie Préperier  
Professeur : Jean-Pierre Rey  
Déposé le : 30 juillet 2018

## RÉSUMÉ

Le but de ce projet est de fournir une interface graphique utilisable par les membres d'une organisation scolaire. Le programme est conçu afin de pouvoir évaluer des élèves de primaires selon les objectifs instaurés par le Plan d'Étude Romand (PER). Les besoins de l'application ont été mis en place en collaboration avec un membre du centre ICT-VS de Sion.

Avant le développement concret du programme, une analyse des outils d'évaluations existants ainsi qu'un état de l'art sur la méthodologie de travail sont réalisés.

Grâce à un tableau comparant les fonctionnalités existantes et les demandes du client, il est possible de déterminer quelles sont les options les plus intéressantes pour le projet. À la suite de cette comparaison, une analyse des langages de programmation est conçue et un choix est effectué afin de réaliser au mieux le logiciel.

L'outil développé propose une solution aux professeurs des classes de primaire pour évaluer les objectifs des élèves qu'ils ont sous leur responsabilité. Les élèves peuvent se connecter à l'interface et avoir un aperçu de leur évolution au fil de leur scolarité. Les données conservées dans le programme sont entièrement modifiables grâce à des interfaces administratives disponibles uniquement pour les directeurs ainsi que pour le superadministrateur.

Dans ce travail, les mots de vocabulaire utilisés sont tirés majoritairement de l'anglais. Dans un but de compréhension, les définitions de ces mots sont répertoriées dans un glossaire au début du document. De plus, beaucoup d'acronymes sont utilisés dans les parties d'analyse. Ceux-ci sont énumérés dans la liste des acronymes également en début de travail.

Mots clés : Évaluation, interface, objectifs, programme, glossaire

## AVANT-PROPOS

En Suisse, la scolarité obligatoire d'un enfant dure onze ans. Elle est composée d'un degré primaire et d'un degré secondaire I (Confédération Suisse, s.d.). Tout au long de leur scolarité, les élèves sont évalués selon une notation numérique allant de un à six. Cette note, vecteur de stress dès le plus jeune âge, ne représente pas forcément l'atteinte des objectifs, mais une moyenne de ceux-ci.

L'idée du projet « Skool » – concaténation des mots anglais « skill » (compétence) et « school » (école) – est née lors du « Hackathon pédagogique » du mois de mai 2017 organisé par l'institut Informatique de gestion de la HES-SO Valais-Wallis. Son idéateur, Mathieu Moser, est venu avec une simple idée qui, durant une journée complète, a été concrétisée grâce à trois étudiants en informatique, une étudiante en tourisme ainsi qu'une intervenante externe.

C'est dans le cadre du travail de Bachelor, réalisé à la HES-SO Valais-Wallis que le projet « Skool » est repris. Le but de ce projet est de créer une plateforme web permettant aux enseignants, parents et enfants de visualiser la progression de l'élève et d'améliorer l'apprentissage ainsi que la collaboration « école-famille ».

## REMERCIEMENTS

Je tiens à remercier les personnes suivantes qui m'ont aidé dans le cadre de ce projet :

M. Jean-Pierre Rey, mon référent pour ce travail de Bachelor, pour avoir assuré le suivi de mon travail. Merci pour ses précieux conseils qui ont évité que je me perde dans la planification de ce travail. Merci également pour sa disponibilité et sa réaction rapide à mes mails.

M. Roger Schaer pour avoir pris du temps afin de m'aider à résoudre les problèmes rencontrés dans le développement Node.js de mon programme.

M. Mathieu Moser, collaborateur du centre ICT-VS de Sion et idéateur de Skool, pour m'avoir aidé à voir plus clair dans le nouveau système scolaire romand, pour avoir suivi le projet depuis le début et pour ses corrections de mon document écrit. Une tâche qui n'a pas dû être de tout repos.

Mes parents pour leur relecture de mon travail et pour leur soutien durant cette période.

Mes camarades de classe pour leur soutien, leur présence et leur bonne humeur tout au long de notre formation.

**TABLE DES MATIÈRES**

**RÉSUMÉ.....II**

**AVANT-PROPOS.....III**

**REMERCIEMENTS.....IV**

**TABLE DES MATIÈRES..... V**

**LISTE DES FIGURES.....IX**

**GLOSSAIRE.....XI**

**LISTE DES ABRÉVIATIONS.....XII**

**INTRODUCTION ..... 1**

**1. MÉTHODOLOGIE DE TRAVAIL ..... 2**

1.1. WATERFALL VS AGILE .....2

1.2. TRAVAIL PAR ITÉRATION.....4

1.3. ARTÉFACTS .....4

1.3.1. PRODUCT BACKLOG .....4

1.3.2. RÉSUMÉ D’ITÉRATION .....5

1.4. MÉTHODOLOGIE ETAT DE L’ART .....5

**2. ÉTAT DE L’ART ..... 6**

2.1. OUTILS EXISTANT .....6

2.1.1. EVALOG .....6

2.1.2. EDUMOOV.....7

2.1.3. ISM.VS.CH .....9

2.2. ANALYSE DU LANGAGE DE PROGRAMMATION ..... 11

2.2.1. BACKEND ..... 11

PHP..... 11

NODE.JS ..... 12

2.2.2. FRONTEND..... 14

HTML ..... 14

PUG..... 14

2.2.3. JAVASCRIPT..... 15

ANGULAR.JS ..... 15

REACT.JS ..... 15

EXPRESS.JS ..... 16

2.2.4. CSS ..... 16

BOOTSTRAP ..... 16

FOUNDATION .....	17
MATERIALIZED .....	17
2.2.5. BASE DE DONNÉES.....	18
MYSQL .....	18
MONGODB.....	18
<b>3. MODÉLISATION .....</b>	<b>20</b>
3.1. USE CASE .....	20
3.1.1. PROFESSEUR .....	21
3.1.2. PARENTS / ENFANTS .....	22
3.1.3. DIRECTEUR .....	23
3.1.4. SUPERADMINISTRATEUR .....	24
3.2. SCHÉMA BASE DE DONNÉES .....	25
<b>4. CHOIX.....</b>	<b>26</b>
4.1. TECHNOLOGIE .....	26
4.2. OUTILS.....	27
4.2.1. OUTILS DE DEVELOPPEMENT .....	27
4.2.2. OUTILS DE MODÉLISATION .....	28
<b>5. DEVELOPPEMENT .....</b>	<b>29</b>
5.1. MOCKUPS.....	29
5.1.1. LOGIN .....	29
5.1.2. ACCUEIL .....	29
5.1.3. RECHERCHE.....	30
5.1.4. FICHE D'INFORMATIONS.....	31
5.2. GUIDE TECHNIQUE.....	32
5.2.1. BASE DE DONNÉES.....	32
5.2.2. PROGRAMME .....	34
BIN .....	34
MODELS .....	35
MODULES .....	36
NODE_MODULE.....	37
PUBLIC.....	38
ROUTES .....	39
VIEW.....	40
5.2.3. HÉBERGEMENT .....	41
HEROKU.....	41
MÉTHODE DE DÉPLOIEMENT.....	42

5.2.4.	CONNEXION .....	46
5.2.5.	GESTION DES ERREURS .....	46
5.3.	GUIDE D'UTILISATEUR .....	47
5.3.1.	INTERFACE DE CONNEXION .....	47
5.3.2.	INTERFACE PROFESSEUR .....	48
	ACCUEIL .....	48
	ANNONCES .....	48
	RECHERCHES AVANCÉES .....	49
	MES CLASSES .....	49
	MES BRANCHES .....	50
	FICHE ÉLÈVE ET ÉVALUATION .....	51
5.3.3.	INTERFACE PARENT/ÉLÈVE .....	51
	ACCUEIL .....	51
5.3.4.	INTERFACE DIRECTEUR .....	52
	ACCUEIL .....	52
	ADMINISTRATION .....	53
5.3.5.	INTERFACE SUPERADMINISTRATEUR .....	54
	ACCUEIL .....	54
	ADMINISTRATION .....	55
5.3.6.	COMMUN .....	56
	ANNONCES .....	56
	RECHERCHES AVANCÉES .....	56
	FICHE ÉLÈVE .....	57
<b>6.</b>	<b>AMÉLIORATIONS FUTURES .....</b>	<b>58</b>
6.1.	INTÉGRATION DE JUSTIFICATIFS AUX ÉVALUATIONS .....	58
6.2.	GRAPHIQUES DE VISUALISATION .....	58
6.3.	EXPORT .....	58
6.4.	SÉCURITÉ DES DONNÉES .....	59
6.4.1.	CRYPTAGE DES DONNÉES .....	59
6.4.2.	DÉDOUBLEMENT D'INFORMATION DE CONNEXION .....	59
6.5.	GAMIFICATION POUR L'INTERFACE ÉLÈVE .....	59
6.6.	ÉVOLUTION DES OBJECTIFS .....	59
	<b>CONCLUSION .....</b>	<b>61</b>
	<b>RÉFÉRENCES .....</b>	<b>62</b>
	<b>RÉFÉRENCE DES FIGURES .....</b>	<b>66</b>
	<b>ANNEXE I : FEUILLE D'ÉVALUATION ACTUELLE DE L'ÉTAT DU VALAIS .....</b>	<b>69</b>

---

<b>ANNEXE II : NEUVIÈME VERSION DU SCHÉMA DE BASE DE DONNÉES SKOOL .....</b>	<b>70</b>
<b>ANNEXE III : PRODUCT BACKLOG .....</b>	<b>71</b>
<b>ANNEXE IV : RÉSUMÉ ITÉRATIONS .....</b>	<b>73</b>
<b>ANNEXE V : RÉSUMÉ SÉANCE CLIENT / RÉFÉRENT .....</b>	<b>83</b>
<b>7. DÉCLARATION DE L'AUTEUR .....</b>	<b>88</b>



## LISTE DES FIGURES

Figure 1 : Méthodologie en cascade .....	2
Figure 2 : Waterfall vs Agile .....	3
Figure 3: Méthodologie Scrum .....	3
Figure 4: User story .....	4
Figure 5 : Machine virtuelle Windows XP avec Evalog .....	6
Figure 6 : Critère d'évaluation Evalog .....	7
Figure 7 : Personnalisation des évaluations – Edumoov .....	8
Figure 8 : Évaluation – Edumoov.....	8
Figure 9 : Évaluation d'une classe de 3 h.....	9
Figure 10 : Évaluation d'une classe de 8 h .....	10
Figure 11: Requêtes en JavaScript au serveur avec Node.js .....	13
Figure 12: AngularJS .....	15
Figure 13: ReactJS.....	15
Figure 14: Bootstrap.....	16
Figure 15: Foundation.....	17
Figure 16: Materialized .....	17
Figure 17: MySQL.....	18
Figure 18: MongoDB .....	19
Figure 19: Exemple de diagramme de cas d'utilisation .....	20
Figure 20: Use Case du rôle "Professeur" .....	21
Figure 21: Use Case du rôle "Parent/Enfant" .....	22
Figure 22: Use Case du rôle "Directeur".....	23
Figure 23: Use Case du rôle "Superadministrateur" .....	24
Figure 24: Version 9 du schéma de base de données Skool .....	25
Figure 25: Tableau comparatif des outils .....	26
Figure 26: Webstorm .....	27
Figure 27: GitLab .....	27
Figure 28: Microsoft Visio .....	28
Figure 29: Mockflow .....	28
Figure 30 : Interface Login .....	29
Figure 31 : Interface d'accueil – Professeur .....	30
Figure 32 : Interface de recherche - Directeur.....	30
Figure 33 : Interface informations personnelle – Parent/Enfant .....	31
Figure 34: config.json – données de connexion à la base de données.....	32
Figure 35: Aperçu de la base de données sur MySQL Workbench .....	33
Figure 36: Structure des dossiers de Skool .....	34
Figure 37: Création de rôles .....	35

Figure 38: Table branche .....	36
Figure 39: Méthode insertAnnonce .....	36
Figure 40: Méthode getAllProf .....	37
Figure 41: Méthode addEtablissementSuperadmin .....	38
Figure 42: Modale de modification d'un élève .....	39
Figure 43: JQuery pour la modale d'édition d'un élève .....	39
Figure 44: Route pour les recherches avancées – interface professeur .....	40
Figure 45: PUG de l'interface prof – mes classes .....	41
Figure 46: Résultat du PUG prof_classe .....	41
Figure 47: Connexion au compte Heroku .....	42
Figure 48: Déplacement dans le dossier Deploy .....	43
Figure 49: Cloner git dans dossier local .....	43
Figure 50: Configuration du git de l'application Heroku .....	44
Figure 51: Erreur de connexion .....	44
Figure 52: Token Heroku .....	44
Figure 53: Connexion au git Heroku .....	45
Figure 54: Page de connexion de Skool .....	45
Figure 55: Connexion .....	47
Figure 56: Page d'accueil du professeur .....	48
Figure 57: Interface d'annonces .....	48
Figure 58: Recherches avancées .....	49
Figure 59: Recherche par classe .....	49
Figure 60: Recherche d'objectif par branches .....	50
Figure 61: Évaluation d'une classe .....	50
Figure 62: Évaluation d'un élève .....	51
Figure 63: Accueil élève .....	51
Figure 64: Accueil interface directeur .....	52
Figure 65: Interface d'administration d'un directeur .....	53
Figure 66: Accueil superadmin .....	54
Figure 67: Interface d'administration du superadministrateur .....	55
Figure 68: Interface d'annonce .....	56
Figure 69: Fiche élève directeur/superadministrateur .....	57

## GLOSSAIRE

- Scrum :** Scrum est un Framework faisant partie de la méthodologie Agile. Il s'agit d'un schéma d'organisation pour le développement de logiciel informatique.
- Product Backlog :** Le Product Backlog est un artéfact utilisé dans la méthodologie Scrum. Il recense toutes les fonctionnalités qui seront développées lors du projet.
- Sprint Backlog :** Le Sprint Backlog est un second artéfact utilisé dans la méthodologie Scrum. Celui-ci contient une partie des fonctionnalités, séparée en plus petites tâches qui seront développées lors d'une seule itération.
- User story (-ies) :** Les User Stories, également appelées récit d'utilisateur, sont de courtes phrases qui décrivent une fonctionnalité. Les User Stories sont répertoriées dans le Product Backlog.
- Backend :** Le Backend, ou également écrit « Back-end », est un terme utilisé pour définir une partie du programme qui ne se trouve pas en premier plan et qui n'est pas directement visible par un utilisateur. Cette partie du programme est généralement utilisée pour accéder aux données.
- Frontend :** À l'inverse du Backend, le Frontend, également écrit « Front-end », est le terme qui, lui, désigne la partie visible du programme. Cette partie du programme sert de couche de présentation.
- Use case :** Les Use cases, aussi appelés cas d'utilisation en français, sont utilisés en Unified Modeling Language (UML) afin de décrire une manière d'utiliser le système développé.
- Framework :** En programmation, un Framework est un ensemble de composants servant de base d'une partie de logiciel. Il peut contenir, par exemple, différents modules afin de faciliter la construction de l'architecture d'un logiciel.
- Mockup :** En terme informatique, un mockup désigne une maquette d'une interface utilisateur.

---

## LISTE DES ABRÉVIATIONS

UML	Unified Modeling Language
PHP	Hypertext Preprocessor
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
JS	JavaScript
BPMN	Business Process Model and Notation
SQL	Structured Query Language
OS	Operating System - Système d'exploitation
PER	Plan d'étude romand
RAD	Rapid Application Development
XP	eXtreme Programming
JS	JavaScript
URL	Uniform Resource Locator
CLI	Command-line interface

## INTRODUCTION

Lors de la rentrée scolaire 2014, le Plan d'étude romand (PER) est introduit dans tous les cycles d'enseignement de tous les cantons romands (CIIP, 2012).

L'évaluation est un problème récurrent en éducation. Avec l'introduction du PER, beaucoup d'objectifs sont devenus transdisciplinaires tels que les objectifs de formation générale et les capacités transversales. Aussi, le traditionnel examen écrit n'est plus suffisant pour justifier la progression des élèves et l'atteinte de tous les objectifs.

C'est dans le but de supprimer, sur le long terme, la notation numérique que le projet Skool est créé. Grâce à cette interface utilisateur, les élèves seront évalués sur les objectifs du PER et non plus sur un examen à un instant T. Le profil du membre est créé au commencement de sa scolarité et servira de référence pendant les onze années qui suivront.

Tous les objectifs peuvent être réalisés au rythme de l'élève et les évaluations sont basées sur les travaux que celui-ci réalisera et non pas sur sa connaissance à un instant précis.

Le but du projet est de proposer une solution centralisée qui permettrait aux professeurs d'évaluer les élèves sur les objectifs du PER. Il est nécessaire d'analyser tous les outils existants qui pourraient correspondre aux attentes du client. Si les outils trouvés ne conviennent pas ou ne remplissent pas toutes les attentes, un programme créé depuis le début doit être développé.

# 1. MÉTHODOLOGIE DE TRAVAIL

## 1.1. WATERFALL VS AGILE

La méthodologie dite « waterfall » (ou cascade) est une méthode de gestion de projet traditionnelle utilisée pour le développement software. Celle-ci suit la logique d'une chute d'eau, ce qui signifie qu'une fois qu'une étape du projet est terminée, l'équipe de développeurs passe à la suivante et ne peut pas, ou presque pas, revenir en arrière (Galiana, 2017).

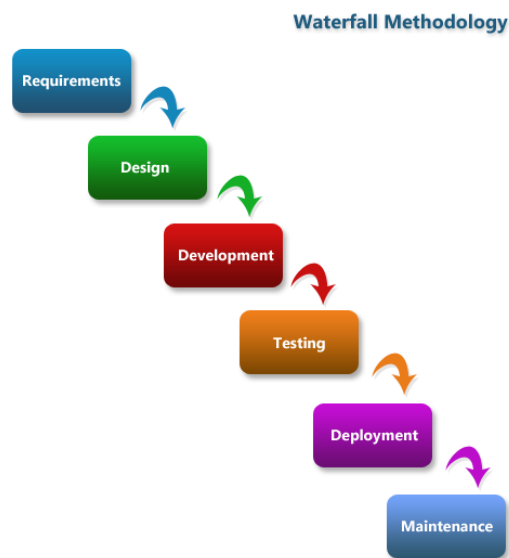


Figure 1 : Méthodologie en cascade

Cette démarche de travail était une révolution, car elle amenait une ligne directrice au développement software. Cependant, « l'inconvénient majeur de cette approche est son manque de flexibilité à cause de son déroulement séquentiel » (Galiana, 2017).

C'est dans le but de pallier ce genre d'inconvénients que la méthodologie Agile a été mise en place. Agile est une méthodologie qui va fonctionner par itérations de courte durée qui sont, en soi, de petits projets indépendants. Le client et l'utilisateur sont incorporés dans le projet et participeront de manière active à celui-ci. Comme l'explique Sébastien Boyer dans son article sur la méthodologie Scrum, « Le centre de gravité n'est donc plus le projet en lui-même, mais bel et bien le produit. » (Boyer, 2016) Le but étant de pouvoir améliorer en continu le projet en cours.

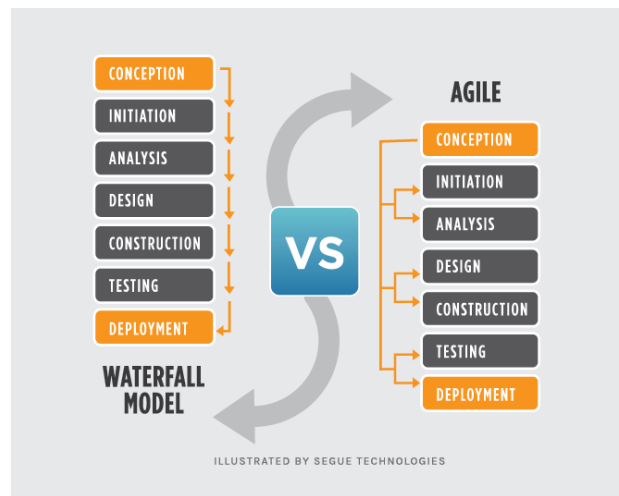


Figure 2 : Waterfall vs Agile

Un manifeste fut créé en 2001 par plusieurs experts de développement logiciel. « Le Manifeste Agile décrit la philosophie de cette nouvelle approche, et amorce un véritable changement culturel. L'humain est remis au centre des préoccupations en lieu et place des outils. » (Boyer, 2016)

Trois méthodes Agiles font partie des plus connues actuellement. Il s'agit de la Rapid Application Development (RAD), eXtreme Programming (XP) et Scrum. (Boyer, 2016) Dans le cadre de ce travail de Bachelor, la méthodologie Scrum est celle qui a été choisie, car utilisée lors des trois années de cours à l'HES. De plus, cette méthodologie a été adaptée aux besoins du projet Skool et elle est « de très loin, la méthodologie la plus utilisée parmi les méthodes agiles existantes » (Florent, s.d.).

### Scrum Process - Overview

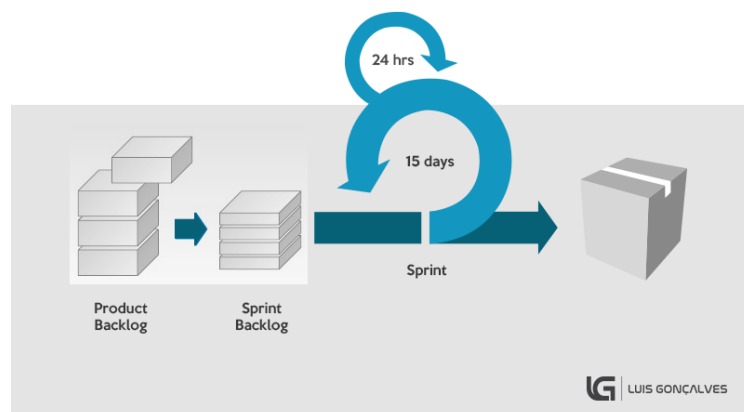


Figure 3: Méthodologie Scrum

## 1.2. TRAVAIL PAR ITÉRATION

Comme expliqué ci-dessus, une méthodologie de travail, par itérations de deux semaines, inspirée de la méthodologie Scrum, a donc été choisie pour la conception de ce travail.

C'est une version « Scrum light » qui sera donc appliquée. En effet, la méthodologie initiale est normalement utilisée pour des projets en équipe. Pour ce travail individuel, la méthode Scrum n'est donc pas adaptée. Cependant, une méthodologie de travail par itération sera exécutée dans la conception du projet et les artéfacts utilisés lors du projet sont les mêmes que ceux présents dans Scrum.

## 1.3. ARTÉFACTS

### 1.3.1. PRODUCT BACKLOG

Le Product Backlog est un document qui recense toutes les fonctionnalités voulues par le client. Il est construit avec des définitions aussi appelées « user stories ». Le modèle de ces « stories » est « En tant que... [rôle], je souhaite (que)... [fonctionnalité], afin de... [but] ».

L'objectif de ce document est de montrer les valeurs ajoutées par les fonctionnalités et, ainsi, évaluer leur pertinence dans le projet.



Figure 4: User story

Dans le même document que le Product Backlog, se trouvent également les résumés des différentes séances faites avec le client ou le référent. Ces fichiers permettent de garder un historique de tous les points et sujets abordés lors des rendez-vous, tels que les suggestions, les conseils, les interrogations et les futures tâches.



### 1.3.2. RÉSUMÉ D'ITÉRATION

Le résumé des itérations comporte des tableaux avec six catégories. Le but de cet artéfact est de représenter sur les six itérations le nombre d'heures de travail passé par thème.

Les catégories sont :

- Recherche – Activité consistant à rechercher des documentations utiles pour la rédaction du travail de Bachelor.
- Analyse – Temps consacré à la lecture et l'étude des documents trouvés.
- Séance – Temps utilisé pour les séances avec le client ou responsable.
- Administration – Période de rédaction et de création de tous les artéfacts.
- Développement – Durée de conception pratique du projet
- Rédaction – Temps utilisé pour la rédaction du document Word

En plus des différentes grilles horaires recensant les temps de travail, un journal de bord est annexé à toutes les feuilles d'itérations. Il contient toutes les activités faites lors desdites itérations. Chaque jour de travail est décrit brièvement avec les tâches exécutées.

Un Sprint Backlog est également intégré au document de résumé d'itérations. Cet artéfact est généralement utilisé pour extraire une partie des User Stories d'un Product Backlog et les détailler en tâches. Ce Sprint Backlog liste donc toutes les tâches prévues pour l'itération en cours ainsi que leur état.

### 1.4. MÉTHODOLOGIE ETAT DE L'ART

L'état de l'art de ce document va se concentrer sur les outils existants dans le domaine de l'évaluation de compétence dans le domaine éducatif ainsi que sur les langages de programmation qui seront utilisés pour la partie développement de ce travail.

L'analyse des outils existant se baseront sur les discussions eues avec le client du projet. Différents outils d'évaluation de compétences ont été transmis et seront donc analysés. En ce qui concerne les langages de programmation qui seront comparés, un premier tri a été fait en amont. Il n'était pas possible d'analyser tous les langages existants. Les plus pertinents ont été sélectionnés afin d'être utilisés dans l'état de l'art dû à leurs accessibilités (open source) et à la maîtrise ou connaissance de ceux-ci.

Le temps de travail ainsi que les tâches passées à la rédaction de l'état de l'art sont recensés dans les documents officiels de la gestion de projet par itération.

## 2. ÉTAT DE L'ART

### 2.1. OUTILS EXISTANT

Lors de la première séance avec le client, Mathieu Moser, différents outils encore utilisés de nos jours ont été mentionnés et feront l'objet d'une analyse.

#### 2.1.1. EVALOG

Evalog est « un gestionnaire d'évaluation par objectif. Il a été élaboré dans le cadre d'un projet pédagogique de décloisonnement qui a démarré en 1991 à Sierre en Suisse. Le but principal de ce projet est de différencier les rythmes et les parcours d'apprentissage, afin de permettre à chaque élève d'atteindre son meilleur niveau de compétences. » (Evalog, s.d.)

Cet outil est actuellement disponible uniquement sur les OS de Windows 98, Windows 2000, Windows XP et Windows Vista. Une machine virtuelle a donc dû être installée pour l'analyse de cet outil.

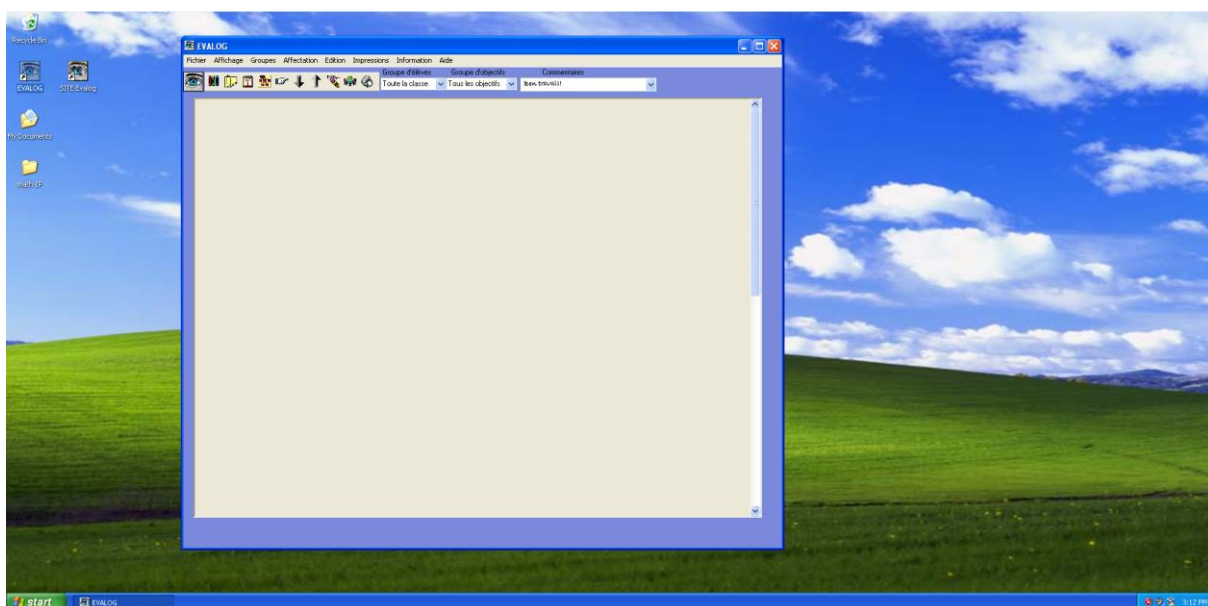


Figure 5 : Machine virtuelle Windows XP avec Evalog

Evalog est toujours utilisé aujourd'hui par quelques personnes ayant accès à des ordinateurs contenant les OS compatibles. L'outil a été mis à jour sur la base du PER. Les objectifs actuels peuvent donc être évalués via Evalog.

Le programme est à télécharger gratuitement sur le site d'Evalog et différentes listes d'objectifs sont fournies avec le fichier exécutable.

Malheureusement, Evalog est passablement difficile à prendre en main. Peu d'informations sur son utilisation sont présentes sur le site officiel.

La fonctionnalité principale de ce programme est de pouvoir évaluer les objectifs par branche d'une classe. Il est également possible de rajouter un commentaire prédéfini à chaque évaluation. Quatre symboles sont alors disponibles sur la feuille d'évaluation pour chaque objectif.

- Le code "carré plein" indique qu'un objectif fondamental est atteint.
- Le code "carré vide" indique que l'élève doit encore travailler cet objectif fondamental.
- Le code "rond plein" objectif non fondamental maîtrisé.
- Le code "rond vide" travailler encore cet objectif.

Figure 6 : Critère d'évaluation Evalog

De plus, Evalog donne la possibilité de créer des bilans individuels. « Au terme d'une période de 4 à 5 semaines, chaque enfant reçoit son bilan individuel. Il s'agit d'un tableau qui récapitule les objectifs atteints ou non atteints. Dans une perspective formatrice, cette évaluation vise à donner à l'élève la possibilité de faire des choix. » (Evalog, s.d.)

Malheureusement, comme mentionné plus haut, Evalog est un logiciel qui n'est plus compatible avec les nouvelles versions de Windows ou bien même Mac et Linux. D'autre part, ce programme reste un client lourd qu'il faut installer sur chaque poste utilisant ce système d'évaluation et la mise en place de tous les objectifs PER pourrait prendre du temps.

À l'inverse d'une plateforme web, la base de données n'est pas centralisée et les données utilisées dans Evalog devront être mises à jour de manière manuelle.

### 2.1.2. EDUMOOV

EdumooV est une plateforme web française qui propose différents outils pour aider les enseignants dans leurs tâches de préparation et de gestion de classe. Le site propose différents outils tels qu'une fiche de préparation par discipline, qui permettent aux enseignants de partager des exercices, un cahier-journal numérique afin d'y préparer les activités d'une classe et celui qui est intéressant pour ce travail de Bachelor, un livret scolaire numérique.

« L'application en ligne Ma classe propose un livret scolaire numérique permettant de gérer facilement les évaluations de vos élèves et d'éditer en quelques clics des bilans de compétences. » (EdumooV, s.d.)

L'outil « Ma classe » permet à l'utilisateur de rajouter une classe dans le système via un système de recherche avec un code postal. L'utilisateur pourra alors sélectionner son établissement et les

niveaux de classe qu'il souhaite évaluer. Il peut également ajouter des élèves dans sa classe, soit manuellement, soit grâce à un import de fichier csv.

Les objectifs sont entrés manuellement par l'enseignant et celui-ci peut personnaliser son codage des compétences afin que celles-ci correspondent à sa pratique pédagogique.

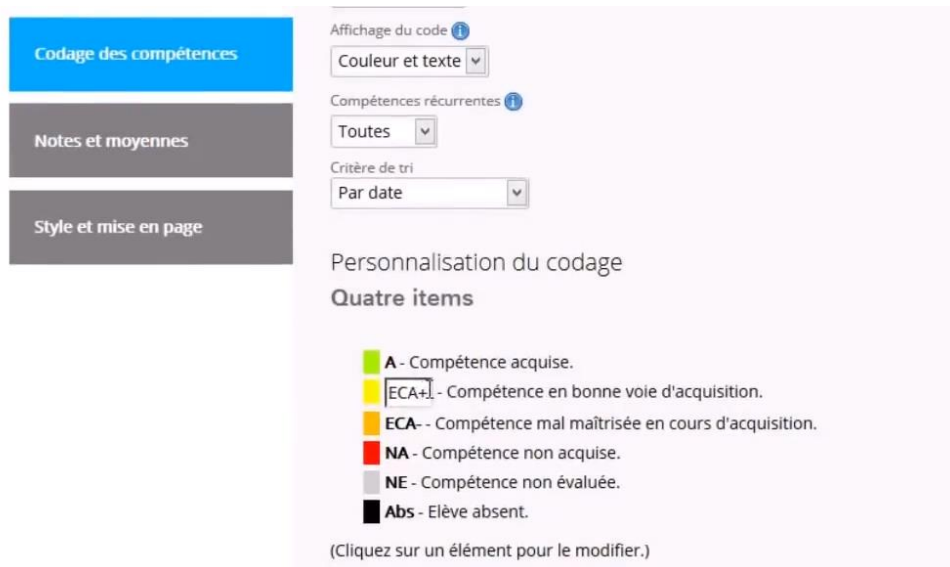


Figure 7 : Personnalisation des évaluations – EdumooV

Une fois le paramétrage fait, le professeur peut donc procéder à l'évaluation des compétences dans un tableau recensant tous ses élèves et tous les objectifs d'une évaluation. Un simple clic est suffisant pour indiquer si l'objectif est atteint ou non.

	Transformer une phrase simple affirmative en phrase négative ou interrogative, ou inversement.	Différencier une phrase affirmative d'une phrase interrogative.	Identifier le verbe conjugué dans une phrase simple.
Sélection globale	A B C D NE Abs ✖	A B C D NE Abs ✖	A B C D NE Abs ✖
<b>CE2</b>			
ABBÉ Oscar →	A B C D NE Abs ✖	A	A B C D NE Abs ✖
COCHET Henri →	A B C D NE Abs ✖	A	A B C D NE Abs ✖
COPTERE Elie →	A B C D NE Abs ✖	A B C D NE Abs ✖	A B C D NE Abs ✖
DORMI Djamal →	A B C D NE Abs ✖	A B C D NE Abs ✖	A B C D NE Abs ✖
DORSA Elsa →	A B C D NE Abs ✖	A B C D NE Abs ✖	A B C D NE Abs ✖
JAVEL Aude →	A B C D NE Abs ✖	A B C D NE Abs ✖	A B C D NE Abs ✖
MACHIN Bidule →	A B C D NE Abs ✖	A B C D NE Abs ✖	A B C D NE Abs ✖
MALE Annie →	A B C D NE Abs ✖	A B C D NE Abs ✖	A B C D NE Abs ✖
PORÉE Eva →	A B C D NE Abs ✖	A B C D NE Abs ✖	A B C D NE Abs ✖
ROÏD Paula →	A B C D NE Abs ✖	A B C D NE Abs ✖	A B C D NE Abs ✖
<b>CM1</b>			

Figure 8 : Évaluation – EdumooV

Cette application donne la possibilité de générer des livrets personnalisés par élèves au format PDF. L'enseignant peut s'occuper d'élèves de différents niveaux scolaires. L'application « Ma classe » répartit, automatiquement, les évaluations aux bons étudiants. Par exemple, un élève du premier cycle aura sur son livret uniquement les objectifs de ce cycle même si dans une évaluation (groupement d'objectifs) certains objectifs étaient destinés à plusieurs cycles.

Edumooov est une bonne représentation du but du projet Skool. Cependant, cette application manque de fonctionnalité. Le professeur ne peut pas laisser de commentaires personnalisés à chaque étudiant pour ses évaluations, et l'élève ne peut pas accéder à une plateforme qui résumerait ses objectifs.

De plus, l'application « Ma classe » d'Edumooov est basée sur le système français. Il n'est donc pas possible de travailler avec un système scolaire suisse avec cet outil. Lors de la création d'une classe, le programme propose le niveau scolaire des élèves et ce sont les années françaises (CE1, CE2, CM1, CM2, etc.).

### 2.1.3. ISM.VS.CH

Au jour d'aujourd'hui, les enseignants du canton du Valais utilisent la plateforme en ligne du Département de la Formation et de l'Economie, intitulée « Internet School Management » (ISM). En ce qui concerne les évaluations, ils remplissent un document interactif tout au long de l'année et par élèves. La plateforme calcule alors automatiquement la moyenne en fonction des pondérations de chaque note. Les enseignants peuvent, par la suite, exporter les résultats sous Excel ou PDF.

The screenshot shows the 'Statut des notes' interface for the subject 'Mathématiques'. It includes instructions on how to enter scores (1-4) and a table with columns for 'Elevé', 'Eval.', and '04.06'. The table lists 20 students with their respective scores and status icons.

Elevé	Eval.	04.06
Ran	1	☹
Isane	1	☹
Zharfate	1	☹
Siecs	4	☺
Itzine	1	☹
Jessia	1	☹
Mila	1	☹
Inzo	2	☹
Zharie	2	☹
Andreas	1	☹
Joline	2	☹
Don	1	☹
Anja	3	☹
Ihan	1	☹
Jodrey	2	☹
Mathias	1	☹
Jamini	3	☹
Lucome	1	☹
Lena	1	☹

At the bottom, the form fields for 'Nom de la série' (Evaluation Février) and 'Date' (06.06.2018) are visible.

Figure 9 : Évaluation d'une classe de 3 h

Au jour d'aujourd'hui, seules les classes de 3H sont évaluées avec un système d'appréciation. Les classes suivantes sont, quant à elles, évaluées selon une méthode de notation numérique (1 à 6).

Navigation: ELEVES | ABSENCES | NOTES | DOCUMENTS | STATISTIQUES | REMPLACEMENTS | MES INFOS | AIDE | QUITTER

Statut: Dernière connexion : 06.06.18 11:31

Titre: Saisie des notes

Branches: Musique | Barèmes: Notes de 0 - 1 à 6 | Type de saisie des notes: Saisie avec point décimal

Instructions:

- Vous modifiez une série existante en cliquant sur [icône]
- Vous supprimez une série existante en cliquant sur [icône]
- Vous visualisez l'histogramme de la répartition des notes en cliquant sur [icône]
- Vous pouvez coller une série de notes copiées d'un tableur (Excel-Open Office) en collant la série dans le champ du haut, puis en cliquant sur [icône]

Elève	Notes	Moy.
Amanda	5.4	
Keran	4.5	
Eloise	4.5	
Avril	6	
Alissia	6	
Charline	2.3	
Corentin	4.6	
Paul	3.4	
Coline	4.4	
Délyane	4.4	
Antoine	5.5	
Emma		
Samuel		
Jade		
Lorenzo		
Louisa		
Loane		
Anny		
Adrien		
Théo		
Léo		
Moyenne		

Figure 10 : Évaluation d'une classe de 8 h

Cette méthode est très restrictive et ne permet pas de garder une trace des travaux responsables de l'évaluation, comme demandé dans les fonctionnalités de Skool. Cette plateforme n'est accessible que par les enseignants et non par les élèves.

En annexe se trouve le document PDF résultant d'une évaluation mi semestrielle pour une élève. Ces documents peuvent être également exportés en format Excel.

## 2.2. ANALYSE DU LANGAGE DE PROGRAMMATION

Pour le développement d'une application web, il faut bien choisir les langages utilisés que ce soit pour le côté serveur – aussi appelé backend – ou pour le côté client, appelé frontend.

Les langages client vont montrer comment le site web doit s'afficher. Alors que les langages serveur vont montrer comment le site web doit se comporter. Par exemple, avec le langage utilisé en frontend, un programmeur peut demander qu'un article soit centré sur la page alors qu'avec un langage de backend, il peut demander que ce même article ne soit disponible uniquement pour les membres du site (Nebra, Les langages serveur, 2018).

Les langages choisis lors d'une recherche en amont seront analysés dans ce document. Ils ont été sélectionnés, comme décrit plus tôt dans le document, grâce à leur popularité, leur pertinence dans le développement web, mais également grâce aux connaissances déjà acquises avec ces langages et leur accessibilité.

En plus de la recherche concernant le langage de développement, une analyse sur le système de base de données est nécessaire lors de l'organisation d'un projet tel que Skool.

### 2.2.1. BACKEND

#### PHP

PHP est le langage de développement web le plus populaire sur le marché. Il est actuellement utilisé dans plus de 80 % des serveurs web (Thinkmobile, 2017). Entre les sites créés avec Wordpress, Drupal ou bien même Facebook, il est très difficile de se déplacer sur internet sans croiser du PHP (Wayner, 2017). Un certain nombre d'avantages peuvent se trouver sur les sites de comparatifs. En voici quelques un :

- L'environnement est facile d'installation (Thinkmobile, 2017).
- Il supporte beaucoup de Frameworks et CMS (Thinkmobile, 2017).
- Tous les services d'hébergement supportent PHP (Thinkmobile, 2017).
- PHP supporte beaucoup de système de management de base de données (Thinkmobile, 2017).
- C'est un langage simple à prendre en main (Wayner, 2017).
- Il est possible de mélanger le code avec le contenu sans utiliser de modèle (Wayner, 2017).
- PHP a été construit pour coexister avec MySQL et toutes ses variantes. Ce qui le rend parfaitement adapter pour faire des requêtes aux bases de données (Wayner, 2017).

En plus de tous ses arguments, PHP est une solution de développement d'application portable. Cela signifie qu'il peut être utilisé sur n'importe quel serveur et/ou plateforme (Chrzanowska, Node.js vs. PHP: Which Environment To Choose For Your Next Project?, 2017).

Néanmoins, PHP ne possède pas que des avantages. Si le développeur aime travailler avec le concept de séparation des préoccupations (Separation of Concerns - SoC), qui consiste à séparer le programme en plusieurs parties distinctes (données - comportement - présentation), alors PHP n'est pas un langage approprié, car il est, généralement, mélangé à du code HTML. Ce qui le rend difficile à maintenir et à étendre avec de nouvelles fonctionnalités (Chrzanowska, Node.js vs. PHP: Which Environment To Choose For Your Next Project?, 2017).

De plus, PHP utilise un modèle classique de client-serveur. En effet, chaque page doit s'instancier, effectuer une connexion à la base de données, paramétrer leurs différentes configurations et effectuer le code HTML pour la présentation. Ces actions à répétition rendent PHP plus lent en Backend (Chrzanowska, Node.js vs. PHP: Which Environment To Choose For Your Next Project?, 2017).

D'autre part, PHP gère les événements de manière synchrone : il va effectuer une requête et attendre la réponse avant d'exécuter une autre opération. Il n'est donc pas capable d'accomplir plusieurs tâches au même moment et perd, de surcroît, du temps (Laurent, 2016).

Pour finir, du point de vue du projet Skool, l'utilisation de PHP nuirait à la productivité du développement, car ce langage n'est pas utilisé au cours des trois années d'études. Un temps d'apprentissage serait alors nécessaire et ces périodes de travail ne seront pas consacrées au développement.

## **NODE.JS**

Node.js est « un environnement open source permettant de développer des applications multi-plates-formes [sic] » (Laurent, 2016). Cet environnement traite les données en JavaScript (JS). Le JS n'est plus utilisé que du côté client, mais également en backend (Laurent, 2016).

Node.js a été créé en 2009 et il utilise le moteur d'exécution V8 de Google, le même que Chrome (Thinkmobile, 2017). Encore très jeune, cet environnement a pourtant déjà fait ses preuves en étant reconnu par de très grandes compagnies telles que LinkedIn, PayPal ou bien même Netflix (Chrzanowska, 12 Top Applications Written in Node.js - Examples from Big Companies, 2017).

Il y a peu de temps, le langage JavaScript était utilisé uniquement en frontend. Le navigateur web exécutait du JS et effectuait des changements en conséquence sur la page web. Lorsqu'une requête était envoyée au serveur, un langage de backend exécutait la demande et envoyait la page demandée au client. Ensuite, toutes les modifications effectuées sur celle-ci étaient faites grâce à du JavaScript (Nebra, Node.js : mais à quoi ça sert ?, 2018).



Avec Node.js, les requêtes envoyées au serveur peuvent être exécutées avec le même langage que celui utilisé en frontend (Nebra, Node.js : mais à quoi ça sert ?, 2018). Cet environnement permet simplement de supprimer un langage en trop dans le projet. Cela apporte « un gain de temps pour le développeur et une économie d'argent pour l'entreprise. » (Sfez, 2017)

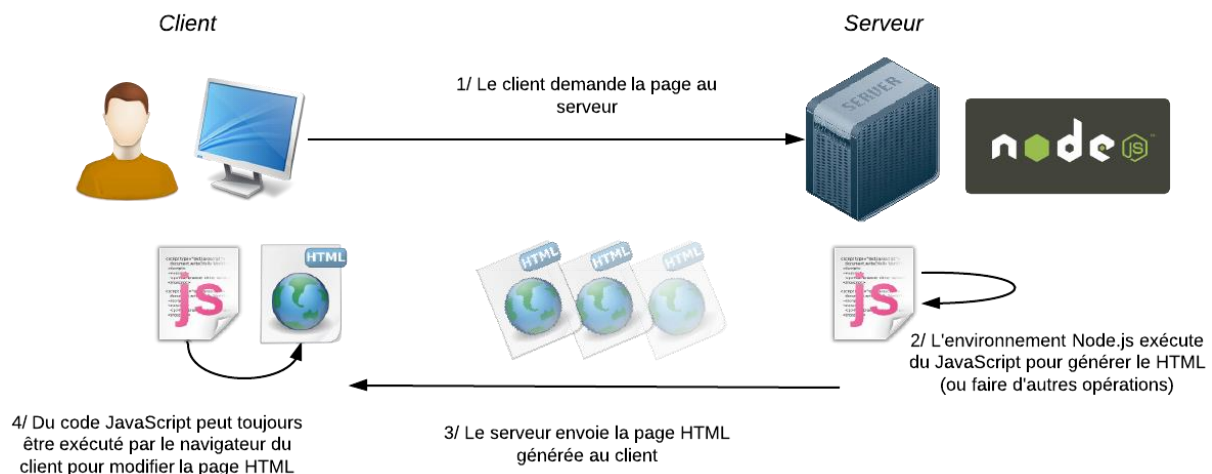


Figure 11: Requêtes en JavaScript au serveur avec Node.js

Les différents avantages de Node.js sont les suivants :

- Un projet entièrement avec du JavaScript permet un gain de temps et d'argent (Sfez, 2017).
- Des performances optimales grâce à sa gestion d'évènement en asynchrone (Sfez, 2017).
- Le gestionnaire de package npm avec une librairie de modules frontend et backend grandissante (Laurent, 2016).
- Node.js n'impose pas de convention stricte et laisse les développeurs choisir la meilleure architecture, modèle, des meilleurs modules et fonctionnalités dont ils ont besoin (Chrzanowska, Node.js vs. PHP: Which Environment To Choose For Your Next Project?, 2017).
- Configuration facile (Laurent, 2016).
- Node.js utilise du JSON qui permet d'avoir une seule syntaxe pour naviguer entre le serveur web et la base de données (Wayner, 2017).
- Une fois que le code JavaScript se trouve dans le cache du navigateur (instanciation), les seules choses qui changent ce sont les données. Pas besoin de faire de nouvelles instanciations (Wayner, 2017).

Évidemment, Node.js possède, lui aussi, des désavantages. Premièrement, c'est un environnement basé sur les évènements. Il n'est pas recommandé pour des opérations lourdes qui utiliseraient

beaucoup de processeurs (CPU) tels que l'édition graphique, audio ou vidéo (Chrzanowska, Node.js vs. PHP: Which Environment To Choose For Your Next Project?, 2017).

Deuxièmement, Node.js est un environnement utilisant du JavaScript. Ce langage n'est pas le plus simple à prendre en main et peut donc nécessiter un temps d'apprentissage conséquent pour le développeur (Lucas, 2014).

De plus, la gestion d'évènement asynchrone peut devenir un désavantage pour un projet développé en Node.js. Cette gestion rend la partie de débogage plus complexe, car il demandera au programmeur d'exécuter beaucoup d'actions afin de trouver la faille (Lucas, 2014).

Cependant, dans le cadre de Skool, Node.js pourrait présenter un avantage concernant la productivité. Ce langage a été utilisé plusieurs fois lors des trois ans d'apprentissage à l'HES.

### 2.2.2. FRONTEND

#### HTML

HTML est « l'une des trois inventions à l'origine du World Wide Web. » (Audrey, 2007) Ce langage est utilisé pour créer des pages web. Il est la fondation du web depuis le premier jour et sa structure, malgré de nombreuses mises à jour, est resté la même (Vincent, 2014).

La version actuelle du HTML est HTML5. Cette nouvelle version a permis d'amener des éléments tels que les éléments sémantiques, graphiques ou même audio (de la Mark & Paradanaud, 2012).

Il est le langage lu par tous les navigateurs. Qu'importe le Framework utilisé, celui-ci sera dynamiquement transformé en HTML afin d'être compris par les navigateurs.

#### PUG

Le Framework Pug est connu, à la base, sous le nom de Jade. Ce Framework est utilisé afin de développer des interfaces dynamiques avec Node.js. Pug n'est qu'un modèle contenu dans la librairie d'un projet Node.js (Iqra, 2017).

Ce Framework ressemble beaucoup au HTML en plus simple. Les balises, si représentatives du langage hypertexte, disparaissent dans un document Pug. Le grand avantage d'utiliser du Pug est qu'il est possible d'incorporer des variables dans le modèle. Cette action permet, par exemple, de décider si l'on veut rajouter des conditions d'affichage sur une page (Iqra, 2017).

### 2.2.3. JAVASCRIPT

#### ANGULAR.JS

Le Framework open source Javascript AngularJS a été créé en 2009 par Google (Andanson, 2016). Ce Framework est compatible avec HTML et Pug. Angular est « un Framework JavaScript qui permet donc de déplacer la logistique de présentation pour le client de la logistique de l'application qui reste sur le serveur. » (Andanson, 2016)

AngularJS peut être utilisé pour des projets destinés à n'importe quelle plateforme. Que ce soit du web, du mobile ou autre, ce Framework s'adaptera à tout (Angular, s.d.).



Figure 12: AngularJS

Ce Framework est facile d'utilisation, seulement, il fut très peu utilisé lors des trois années d'études à l'HES et un temps d'apprentissage sera nécessaire, ce qui baisserait la productivité du projet Skool.

#### REACT.JS

ReactJS est actuellement le Framework utilisé en frontend le plus populaire, même si celui-ci préfère être vu comme étant une bibliothèque plutôt qu'un Framework (Porteneuve, 2018).

Ce Framework est open source, il est piloté par Facebook et plus de 20 000 modules npm ont un lien avec l'environnement React (Porteneuve, 2018). Il est très souvent utilisé afin de créer des interfaces utilisateurs réutilisables (Billey, 2016).

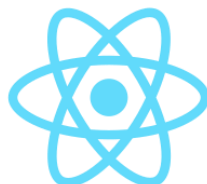


Figure 13: ReactJS

De base, React est utilisé avec des pages HTML, mais il existe des modules npm qui rendent ReactJS compatible avec des pages Pug.

Ce Framework, aussi considéré comme bibliothèque, est très intéressant pour le projet Skool, car il a été créé pour de la création de profils utilisateurs et c'est ce à quoi va ressembler l'interface d'évaluation développée dans ce projet.

Néanmoins, ReactJS n'a pas été travaillé durant la scolarité à l'HES. Un temps d'apprentissage sera dès lors nécessaire afin de l'utiliser pour le projet Skool.

### **EXPRESS.JS**

ExpressJS est un micro-Framework utilisé avec Node.js. Si l'on parle de « micro-Framework », c'est parce qu'il va fournir seulement des outils basiques afin d'accélérer la programmation d'une application Node.js (Nebra, Des applications ultra-rapides avec Node.js, 2014).

ExpressJS est très apprécié pour sa flexibilité, laissant une liberté de développement très grande. Il est simplement une très grande boîte à outils pour le développement d'application Node.js. Le Framework Express va permettre d'exécuter la même chose que Node.js, mais avec, au final, moins de code (Alioze, s.d.).

Ce Framework est très simple d'utilisation et fut employé à maintes reprises dans différents projets Node.js au cours des années d'HES.

#### 2.2.4. CSS

### **BOOTSTRAP**

Bootstrap est un Framework CSS utilisé afin de créer des pages web responsives (Bitschené, 2014). Au départ, il a été créé chez Twitter afin d'unifier les interfaces des applications. Depuis qu'il est devenu open source, il est rapidement devenu un incontournable des Frameworks CSS. Le Framework Bootstrap est très complet et apporte beaucoup de fonctionnalité tel qu'un système de grille pour la mise en page, des formes de texte, des tableaux et autres formulaires (Heuveline, 2017).

Bootstrap est compatible avec la majorité des navigateurs tels que Chrome, Firefox, Internet Explorer ainsi qu'Edge (Bootstrap, s.d.).



Figure 14: Bootstrap

L'évolution de Bootstrap est très rapide et cela peut faire apparaître différents désavantages tels que l'obsolescence du site due aux nouvelles versions du Framework. Étant un Framework très complet, il faut posséder des connaissances au préalable avant de pouvoir utiliser Bootstrap. Par exemple, avoir une bonne connaissance du HTML et du CSS ainsi que du JQuery est nécessaire afin d'utiliser au mieux ce Framework (Chavelli, 2018).

## FOUNDATION

Foundation fait partie des Frameworks les plus prisés du web. Il offre un environnement de conception front simple, mais complet. Il peut être utilisé afin de créer des sites web, des e-mails ainsi que des applications responsives (OpenTuto, 2016).

Foundation est sémantique, flexible, personnalisable et facilite la lecture du code HTML (Foundation, s.d.). Il possède beaucoup de modèles mis à disposition afin de bien le prendre en main, mais reste, cependant, légèrement plus complexe à utiliser (Lazzaroni, 2017).



Figure 15: Foundation

## MATERIALIZE

Ce Framework, basé sur Material Design, permet de créer des interfaces propres très rapidement. Il possède certains systèmes similaires à Bootstrap et quelques modules JavaScript en plus afin de donner des effets aux éléments affichés à l'écran (Berroukeche, 2016).

Materialize fut créé et conçu par Google et a pour particularité d'avoir un design très « flat » (MaterializeCSS, s.d.).

Ce Framework fait partie de cette analyse, car il a été utilisé à plusieurs reprises lors de projet Node.js à l'HES et n'a jamais fait défaut. Sa simplicité d'installation et d'utilisation permet d'obtenir des pages web élégantes très rapidement.



Figure 16: Materialized

## 2.2.5. BASE DE DONNÉES

### MYSQL

MySQL est « un serveur de bases de données relationnelles Open Source. » (Futura Tech, s.d.) Grâce à ce serveur, il est possible de classer les données d'une application dans des tables. Le langage utilisé dans MySQL est du SQL (Structured Query Language). Celui-ci est « le langage standard pour les traitements de bases de données. » (Futura Tech, s.d.)



Figure 17: MySQL

Les avantages d'utiliser MySQL sont les suivants :

- Étant utilisé par beaucoup de personnes, MySQL possède de nombreuses bibliothèques, de tutoriels d'installation et d'exemple afin d'aider à son utilisation (Slant, 2018).
- Ce serveur de bases de données relationnelles est fait pour gérer beaucoup de données (Slant, 2018).

Néanmoins, MySQL possède certains défauts tel que la mauvaise lisibilité des messages d'erreur qui rendent le débogage compliqué et fastidieux.

### MONGODB

MongoDB est un système de gestion de base de données (SGBD) utilisé en NoSQL. À la différence des bases de données relationnelles, les bases en « Not only SQL » n'ont pas besoin de jointure, de schéma ou même de transaction (Ben Debba & Bonneau, Introduction à MongoDB, 2015).

Il existe quatre familles de bases de données NoSQL : les clés/valeurs, orientées colonne, orientées graphe et orientées document. C'est dans cette dernière catégorie que se trouve MongoDB. Toutes les données dans une SGBD sont dans un même document (Ben Debba & Bonneau, Introduction à MongoDB, 2015).

Les avantages de l'utilisation de MongoDB sont les suivants :

- Les requêtes sont très rapides, car les données sont placées dans un seul document (Slant, 2018).
- Ce type de serveur peut contenir et stocker une grande quantité de donnée (Slant, 2018).

Pourtant, il existe des désavantages au serveur de base de données NoSQL tel que MongoDB. En premier lieu, MongoDB n'est pas une bonne solution pour des données relationnelles. De plus, il y existe passablement de rapports concernant la perte de donnée au fur et à mesure que le temps passe sur les bases de données MongoDB (Slant, 2018).



Figure 18: MongoDB

### 3. MODÉLISATION

Avant de commencer la conception du programme Skool, une analyse des processus et des besoins de l'utilisateur est nécessaire. Pour ce faire, la notation UML sera utilisée dans ce travail.

UML est « un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc. » (Roels, 2018)

La modélisation est importante dans le processus de développement logiciel. Elle fait office de « plan » au projet. Sans cette étape, le programme développé ne répondrait peut-être pas aux besoins et aux exigences de l'utilisateur final. « UML nous aide à faire cette description de façon graphique et devient alors un excellent moyen pour « visualiser » le(s) futur(s) logiciel(s). » (Roels, 2018)

#### 3.1. USE CASE

Sur sa page web, Laurent Audibert explique que « Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre. » (Audibert, s.d.)

Le diagramme de cas d'utilisation, appelé aussi Use Case, est construit avec différents éléments. Premièrement, l'utilisateur est généralement représenté par un bonhomme et son rôle est précisé en dessous. Deuxièmement, les frontières du système sont définies par un cadre avec le nom du système au sommet de celui-ci. Les besoins et fonctionnalités demandées sont, finalement, représentés dans des bulles reliées aux acteurs grâce à des associations. (Audibert, s.d.)

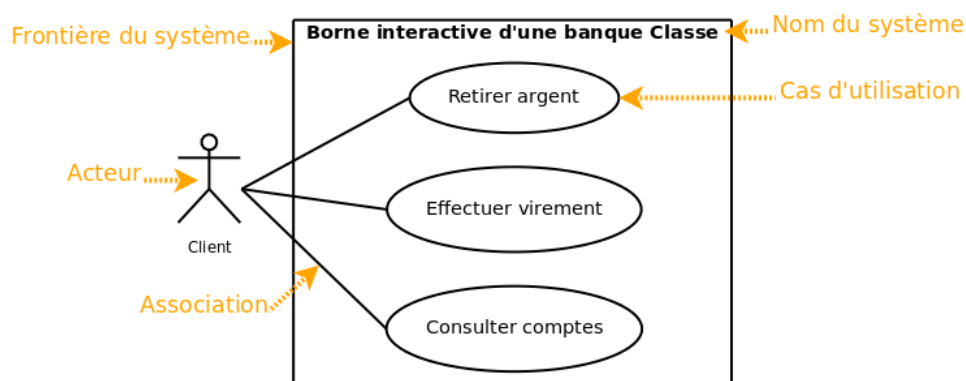


Figure 19: Exemple de diagramme de cas d'utilisation



La modélisation des diagrammes de cas d'utilisation de Skool s'est basée sur le Product Backlog défini avec le client et validée par le référent. Il y a quatre acteurs qui agiront sur la plateforme Skool. Premièrement, les enseignants des écoles concernées pourront venir sur Skool afin de procéder aux évaluations de leurs élèves. Ensuite, les parents et élèves auront un accès, sans droit d'écriture, sur l'évaluation des objectifs. Les deux derniers acteurs pourront, quant à eux, atteindre la plateforme uniquement dans un but de gestion des données et non d'évaluation. Il s'agit du rôle de directeur et celui de superadministrateur.

### 3.1.1. PROFESSEUR

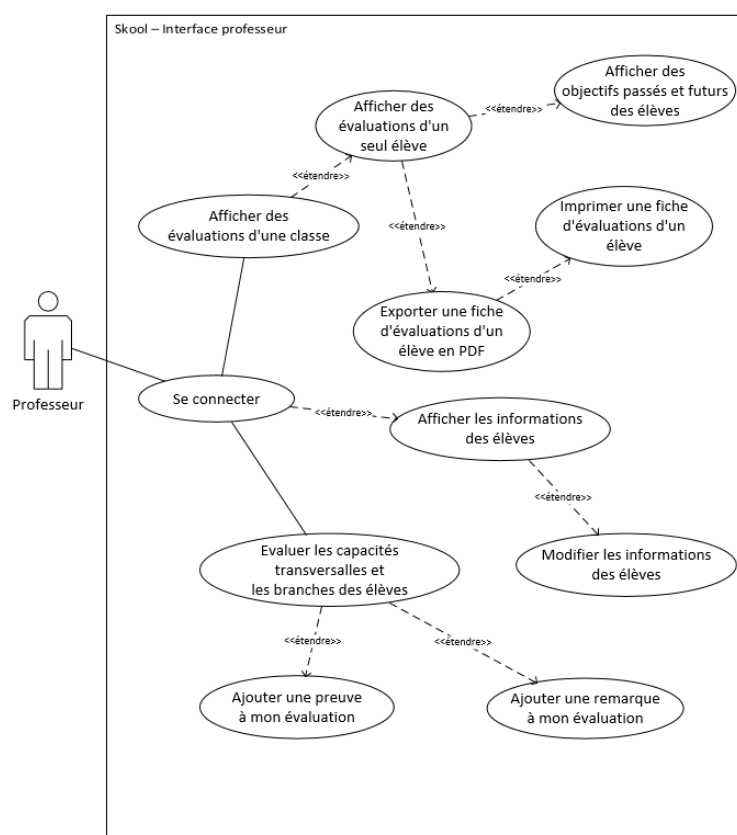


Figure 20: Use Case du rôle "Professeur"

Le diagramme de cas d'utilisation du professeur est le plus grand du projet Skool. En tout premier lieu, le professeur doit se connecter sur la plateforme afin d'avoir accès à toutes les fonctionnalités qui lui sont octroyées. Deux actions principales peuvent être faites par l'enseignant. Premièrement, il pourra évaluer les objectifs de ses élèves. Une fois l'évaluation faite, il peut exporter celle-ci en format PDF afin de l'enregistrer, mais également l'imprimer si voulu. La deuxième fonctionnalité principale de l'interface professeur est de pouvoir afficher toutes les évaluations de sa classe afin d'avoir un aperçu général de l'avancée des élèves dont il s'occupe. Lorsqu'un enseignant arrive sur

le profil d'un élève, composé de ses objectifs, il peut accéder aux objectifs des cycles passés ou futurs de celui-ci.

Lors de son évaluation, le professeur a la possibilité de rajouter une justification à celle-ci sous format audio, vidéo ou simplement un document. De plus, il pourra rajouter des remarques personnelles et totalement personnalisables à ses évaluations d'objectifs.

Les enseignants auront la possibilité de voir les informations personnelles d'un élève (nom, prénom, date de naissance, etc.) et pourront les modifier si nécessaire.

### 3.1.2. PARENTS / ENFANTS

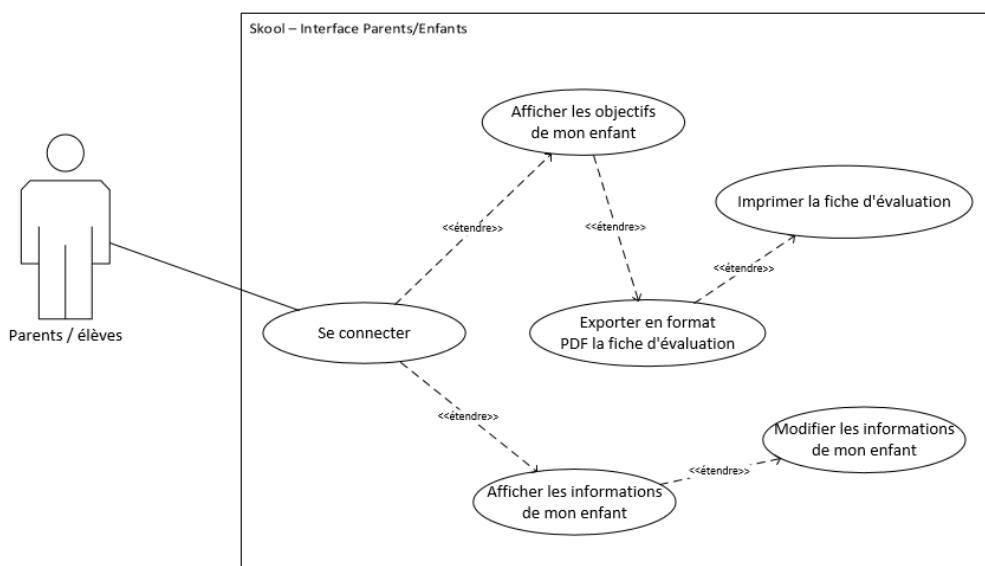


Figure 21: Use Case du rôle "Parent/Enfant"

La nouveauté de Skool, qui n'était pas présente dans les outils analysés précédemment, se trouve dans l'interface parent/enfant.

Ceux-ci pourront se connecter au site web et voir les objectifs de l'élève. Ils ne pourront pas modifier les informations, mais auront un droit de regard sur toutes les évaluations personnelles de l'enfant. Évidemment, comme le professeur, ils pourront exporter la fiche de profil et l'imprimer si tel est leur souhait.

De plus, les parents auront accès aux informations personnelles de leurs enfants (nom, prénom, date de naissance, etc.) et pourront les modifier si nécessaire.

### 3.1.3. DIRECTEUR

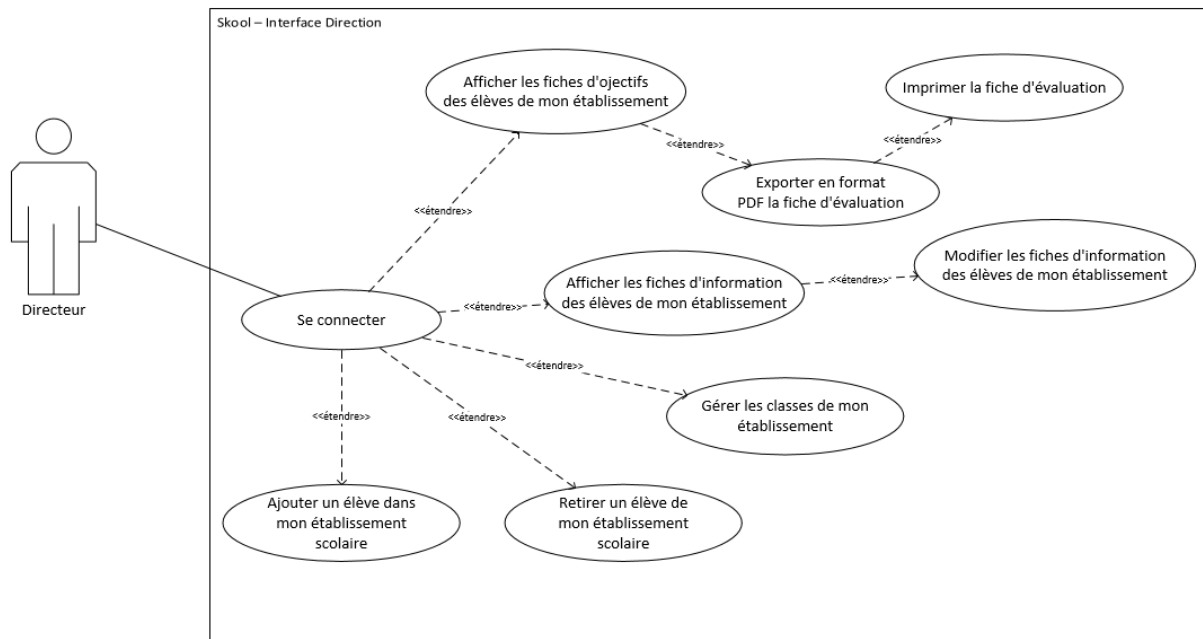


Figure 22: Use Case du rôle "Directeur"

Le directeur possède un droit de gestion sur le logiciel Skool. Il pourra se connecter et voir les fiches d'objectifs de tous les élèves présents dans son établissement.

Il aura également accès aux fiches d'informations personnelles de ceux-ci, comprenant les données telles que le nom, prénom, l'adresse ou bien même la classe de l'enfant, et pourra les modifier.

Le directeur aura l'option de retirer un élève de son établissement et d'en rajouter si celui-ci n'a pas d'établissement attribué. Cependant, il n'aura pas l'autorisation de déplacer directement un élève d'un établissement à un autre.

Une dernière action que le directeur pourra exécuter sur le programme Skool sera de gérer les classes de son établissement. Il pourra en créer de nouvelles et leur attribuer un professeur, supprimer d'anciennes classes n'étant plus utilisées et en modifier le nom si nécessaire.

### 3.1.4. SUPERADMINISTRATEUR

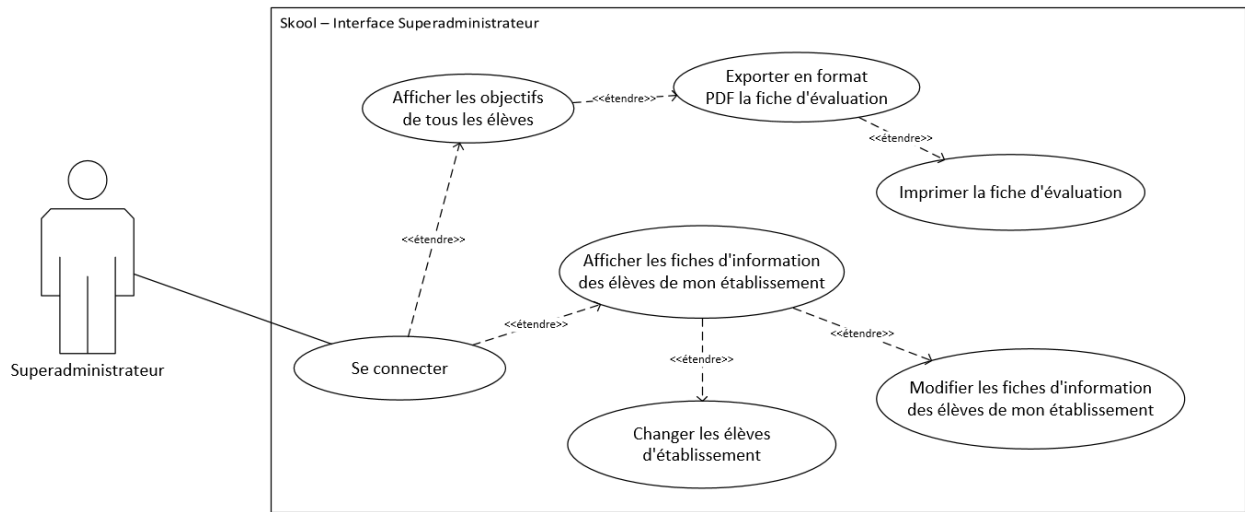


Figure 23: Use Case du rôle "Superadministrateur"

Le diagramme de cas d'utilisation du superadministrateur possède presque les mêmes fonctionnalités que celui du directeur.

Toutefois, le superadministrateur aura l'autorisation de voir les fiches d'évaluation de tous les élèves répertoriés dans le logiciel Skool et il pourra changer un élève d'établissement.

### 3.2. SCHÉMA BASE DE DONNÉES

« Un schéma de base de données représente la configuration logique de tout ou partie d'une base de données relationnelle. » (LucidChart, s.d.)

La création d'un schéma de base de données avant la phase de développement permet au programmeur de réfléchir en amont des connexions entre toutes les tables présentes dans le projet. Cela donne également la possibilité d'anticiper des problèmes qui pourraient ressortir.

Cette phase est importante, car un schéma de base de données donnera un fil rouge au développeur qui évoluera au cours de la création du logiciel. Évidemment, le schéma sur lequel se base le développeur peut être modifié et amélioré au fur et à mesure du projet.

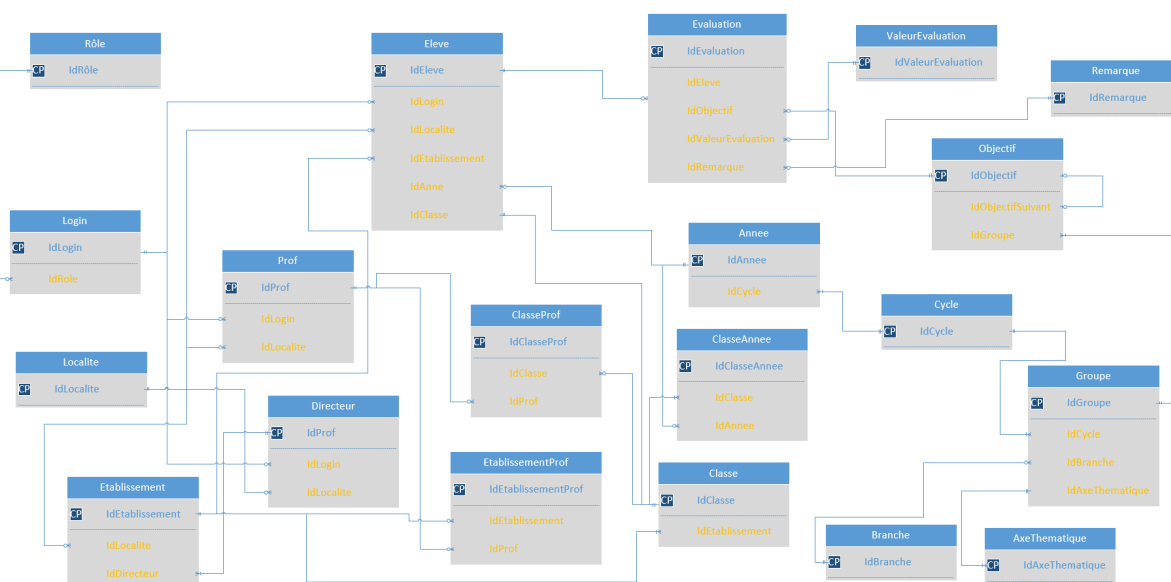


Figure 24: Version 9 du schéma de base de données Skool

Un premier schéma est créé pour le projet Skool. Celui-ci permet de visualiser les connexions entre les différentes tables et, d'un point de vue du développeur, d'imaginer les accès aux données dans l'interface web. Ce schéma se trouve également en annexe de ce document (cf. Annexe II : Neuvième version du schéma de base de données Skool).

Au fur et à mesure des séances avec le client et le référent, le schéma a évolué. Au début de la phase de développement, c'est la quatrième version qui a été choisie.

## 4. CHOIX

### 4.1. TECHNOLOGIE

À la suite des différentes analyses concernant les outils existants, un tableau a été généré afin de comparer les différentes fonctionnalités de chacun d'entre eux.

	Evalog	Edumoov	ism.vs.ch	Skool
Evaluer les objectifs PER	☑	☒	☒	☑
Ajouter un preuve	☒	☒	☒	☑
Aperçu général de tous les élèves	☑	☑	☑	☑
Aperçu d'un seul élève	☑	☒	☑	☑
Exporter en PDF	☒	☑	☑	☑
Login à l'interface (prof)	☑	☑	☑	☑
Client lourd à installer	☑	☒	☒	☒
Interface Web	☒	☑	☑	☑
Accès web pour les parents	☒	☒	☒	☑
Accès de gestion pour admin	☒	☒	☑	☑
Base de donnée en ligne	☒	☒	☑	☑
Evaluation avec niveau (0-4)	☑	☑	☑	☑
Remarque par évaluation	☑	☒	☒	☑

Figure 25: Tableau comparatif des outils

Grâce à ce tableau, il est évident que Skool doit être développé « From scratch », c'est-à-dire depuis le début. Avec les informations présentes dans l'état de l'art, il est maintenant nécessaire de décider des différents langages qui composeront le logiciel.

Premièrement, Node.js sera utilisé en BACKEND, car c'est un langage qui a beaucoup de potentiel et qui est destiné à s'agrandir de plus en plus à l'avenir. Avoir une application entièrement en JavaScript, que ce soit en Frontend ou en backend, va permettre de gagner beaucoup de temps lors de la phase de développement.

En ce qui concerne le FRONTEND, le choix s'est porté sur du Pug plutôt que du HTML. Évidemment, cela revient sensiblement au même, mais le Pug a été développé pour Node.js, il est donc plus judicieux de le choisir.

Concernant le Framework JavaScript, Express.js paraît être un bon point de départ. Il est facile d'utilisation et d'installation. Il facilitera la mise en place des routes entre les différentes interfaces web.

Ensuite, pour ce qui est du Framework CSS, Materialize sera utilisé dans le projet Skool car il est facile à prendre en main et déjà connu.

Pour finir, le serveur de base de données qui sera utilisé sera du MySQL, car les données de Skool sont rangées dans différentes tables. Ayant créé un schéma de base de données relationnelle, il n'était pas possible d'utiliser un serveur NoSQL.

Naturellement, ces choix ne sont pas immuables. Lors de l'avancée de la partie de développement, d'autres Frameworks se joindront, certainement, au premier lot.

## 4.2. OUTILS

### 4.2.1. OUTILS DE DEVELOPPEMENT

Tout le développement de l'application Skool est fait grâce à l'environnement de développement intégré (IDE) Webstorm. Cet environnement supporte les Frameworks cités ci-dessus et aide à la création de projets en Node.js.



Figure 26: Webstorm

Un logiciel de gestion de versions décentralisé est également requis lors de la phase de développement de logiciel. Dans le cadre du projet Skool, c'est l'interface GitLab qui est utilisée.

Ce logiciel de gestion permet également de mettre en place un environnement de test, plus connu sous le nom d'intégration continue (CI - Continuous integration). Grâce à cette gestion automatique, il est simple d'exécuter des tests de validation sur le code qui est sauvegardé.



Figure 27: GitLab

#### 4.2.2. OUTILS DE MODÉLISATION

Durant la phase de modélisation, les logiciels utilisés étaient Microsoft Visio et le site internet mockflow.com.

Visio est un programme développé par Microsoft et fait partie de la suite Office. Il permet la création de diagrammes professionnels (Microsoft, s.d.).

Dans le cadre du travail de Bachelor, Visio est utilisé afin de créer le schéma de base de données (cf. Annexe II : Neuvième version du schéma de base de données Skool) ainsi que les différents uses case (cf. chapitre 3.1 : use case).



Figure 28: Microsoft Visio

En plus de Microsoft Visio, le deuxième outil de modélisation utilisé est le site internet mockflow.com. Ce site permet de créer des mockups (cf. chapitre 5.1 : mockups). Grâce à cet outil, il est possible de développer le visuel d'un site internet et de paramétrer les navigations entre les différentes pages.

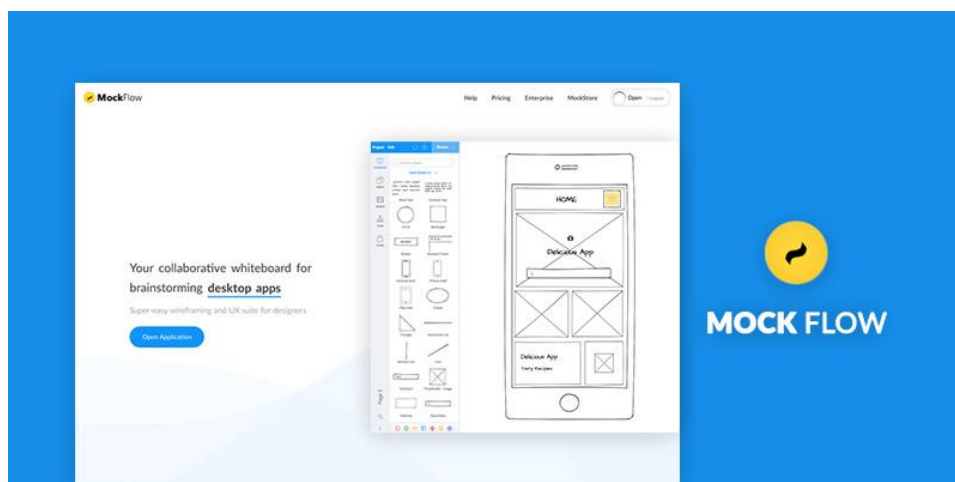


Figure 29: Mockflow



## 5. DEVELOPPEMENT

### 5.1. MOCKUPS

Au commencement de la phase de développement, il est important que les besoins du client soient compris par le programmeur. Pour cela, les deux acteurs vont mettre au point des mockups qui représentent les différentes interfaces utilisateur du logiciel.

En ce qui concerne Skool, une vingtaine de maquettes ont été élaborées. Voici les plus importantes :

#### 5.1.1. LOGIN

La page de login sera disponible pour tous les acteurs de l'application Skool. Suivant l'identifiant entré, l'utilisateur sera redirigé vers sa page d'accueil avec des droits qui lui sont propres.

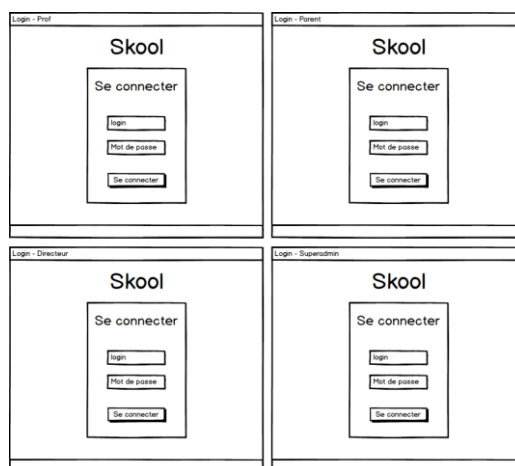


Figure 30 : Interface Login

#### 5.1.2. ACCUEIL

Les professeurs, directeurs et superadministrateurs arriveront, après la page de connexion, directement sur une interface d'accueil.

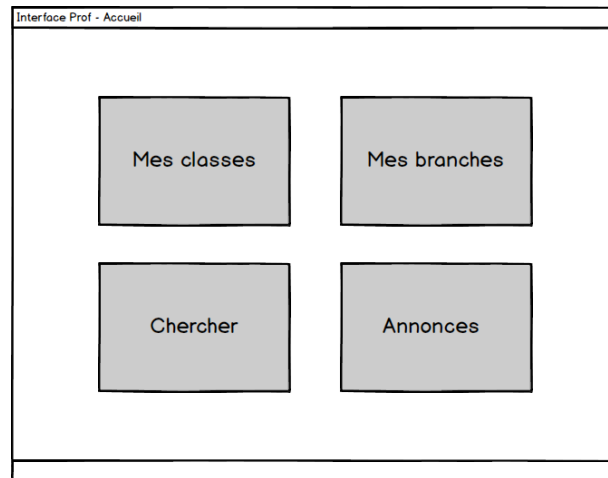


Figure 31 : Interface d'accueil – Professeur

Ces trois rôles auront un accès commun à une page d'annonces. Celle-ci permettra aux directeurs et superadministrateurs de poster des messages d'informations importantes en tout temps. Les professeurs, quant à eux, ne pourront qu'afficher les messages, mais n'auront pas de droit d'écritures sur ceux-ci.

Les parents et élèves n'ont pas de page d'accueil similaire aux autres acteurs de l'application. Ils seront directement redirigés, après leur connexion, sur la fiche d'information de l'enfant voulu.

### 5.1.3. RECHERCHE

La volonté de pouvoir recherche facilement un élève vient directement du client. Il est important que les professeurs, directeurs et autres administrateurs puissent accéder rapidement aux informations recherchées. C'est pourquoi une interface de recherche est intégrée au logiciel Skool.

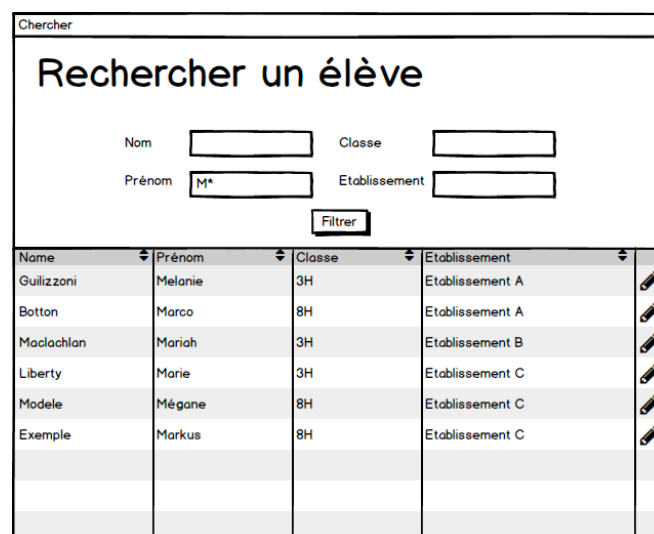


Figure 32 : Interface de recherche - Directeur

Évidemment, il sera possible de trouver les fiches personnelles des élèves via d'autres chemins. Par exemple, le professeur pourra les rechercher au travers des classes qui lui sont assignées et les directeurs via des établissements.

#### 5.1.4. FICHE D'INFORMATIONS

Une des interfaces les plus importantes n'est autre que la fiche d'information personnelle de l'élève. C'est sur cette page que se trouveront toutes les données personnelles de l'enfant ainsi que ses objectifs.

The screenshot shows a web interface titled "Interface parent - Accueil". On the left is a silhouette of a person. To the right are labels for personal information: "Nom", "Prénom", "Date de naissance", "Adresse", and "Localité". A printer icon is in the top right. Below is a list of "Branche A", "Branche B", "Branche C", and "Branche D". Under "Branche B", there are two entries for "Sous-branche B".

Figure 33 : Interface informations personnelle – Parent/Enfant

Cette page sera accessible par tous les acteurs de l'application Skool. Cependant, les différentes personnes ne pourront pas toutes changer les informations.

## 5.2. GUIDE TECHNIQUE

Ce guide est réalisé afin de permettre, dans un futur proche, de reprendre le projet de Skool et d'en connaître toutes les spécificités techniques telles que les accès à la base de données, la construction du projet Node.js, etc.

### 5.2.1. BASE DE DONNÉES

La base de données est construite en MySQL. Lorsque le programme est exécuté en local, il utilise la base de données « development ». Il est possible d'accéder aux tables grâce à l'interface phpMyAdmin à l'adresse suivante :

<http://localhost/phpmyadmin/>.

Les informations de connexion sont écrites dans le document « config.json » du projet.



```
1 {
2   "development": {
3     "username": "root",
4     "password": null,
5     "database": "skooldb",
6     "host": "localhost",
7     "dialect": "mysql"
8   },
9   "test": {
10    "username": "root",
11    "password": null,
12    "database": "database_test",
13    "host": "127.0.0.1",
14    "dialect": "mysql"
15  },
16  "production": {
17    "username": "root",
18    "password": null,
19    "database": "database_prod",
20    "host": "localhost",
21    "dialect": "mysql"
22  }
23 }
24 }
```

Figure 34: config.json – données de connexion à la base de données

La version déployée de l'application possède, quant à elle, une base de données en ligne. L'extension « ClearDB MySQL » est installée sur l'application de Skool.

Cette base de données est accessible via un programme de lecture de base de données tel que MySQL Workbench. L'URL de la base de données contient toutes les informations de connexion :

mysql://b75629952d9f7d:5ce97830@eu-cdbr-west-02.cleardb.net/heroku\_b04cfc9b108fab5?reconnect=true

L'hôte de la base de données est ce qui est écrit après le @ et avant le/: eu-cdbr-west-02.cleardb.net.

Le login d'utilisateur se trouve au début de l'URL jusqu'au double point : b75629952d9f7d.

Le mot de passe est ce qui reste entre le double point et le @ : 5ce97830.

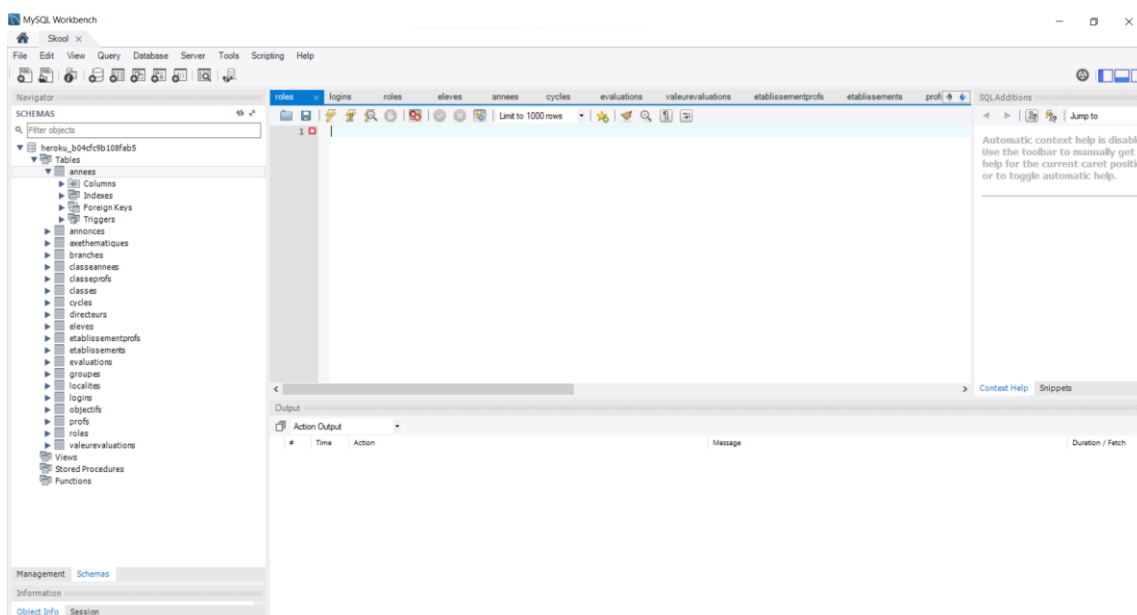


Figure 35: Aperçu de la base de données sur MySQL Workbench

## 5.2.2. PROGRAMME

Le programme est construit comme suit :

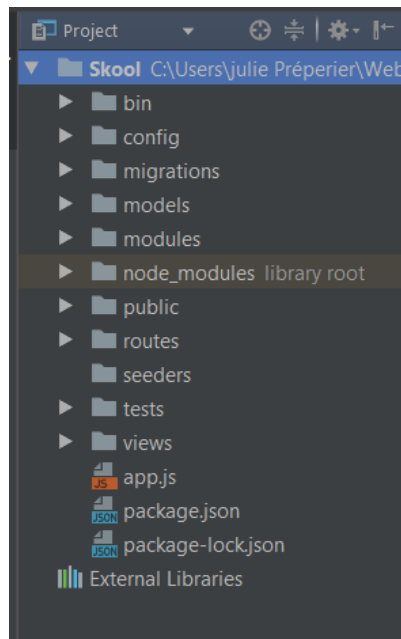


Figure 36: Structure des dossiers de Skool

### **BIN**

Le premier dossier contient le document qui permet le lancement de l'application, c'est-à-dire, `www.js`. C'est dans ce fichier que l'on peut préciser le port à écouter. On peut lancer la synchronisation de Sequelize (module gérant la base de données du programme) et c'est également à cet endroit que les données qui ne sont pas modifiables dans le programme sont créées.

Dans le cadre du projet Skool, il existe plusieurs informations qui sont immuables et qui sont donc créées au premier lancement de l'application. Par exemple, il n'est pas possible de créer un rôle supplémentaire dans le projet. Il y a uniquement quatre types de personnes qui existent dans Skool : les élèves, les professeurs, les directeurs et le superadministrateur.

Il n'est donc pas nécessaire de créer une interface de création de ces rôles qui est sujette à l'erreur telle que des fautes d'orthographe. C'est pour cela que les données de la table Rôle sont créées au premier lancement de Skool.

```
models.Role.findAll({
}).then((roles) => {
  if (roles.length == 0) {
    models.Role.create({
      id_role: 1,
      roleName: 'Super administrateur'
    }).then(() => {
      models.Role.create({
        id_role: 2,
        roleName: 'Professeur'
      })
    }).then(() => {
      models.Role.create({
        id_role: 3,
        roleName: 'Eleve'
      })
    }).then(() => {
      models.Role.create({
        id_role: 4,
        roleName: 'Directeur'
      })
    })
  }
});
```

Figure 37: Création de rôles

Il existe quatre autres tables de la base de données qui sont également remplies dans le fichier `www.js`. Il s'agit de la table `Cycle` – les trois cycles existant dans l'éducation suisse –, `Année` – les onze ans d'école obligatoire –, `ValeurEvaluation` - le nouveau système de notation choisi pas le client - et la table `Login` afin de créer le login du superadministrateur même en cas de réinitialisation de la base de données.

## MODELS

Le dossier « `models` » contient tous les fichiers pour la création d'une table dans la base de données avec le module `Sequelize`. Il faut un fichier par table existante et dans le cas de `Skool`, cela représente 19 fichiers.

Grâce à ces documents, il est possible de définir les champs existant dans une table et les jointures entre les tables d'une base de données.

Par exemple, la table qui désigne les branches scolaires est construite comme tel :

```

1 module.exports = (sequelize, DataTypes) => {
2
3   /* -- BRANCHE TABLE IN DB --*/
4
5   var Branche = sequelize.define('Branche', {
6     id_branche: {
7       type: DataTypes.INTEGER,
8       primaryKey: true,
9       autoIncrement: true
10    },
11    brancheName: {
12      type: DataTypes.STRING,
13      allowNull: false
14    }
15  });
16  Branche.associate = (models) => {
17    Branche.hasMany(models.Groupe, { foreignKey: { name: 'id_branche', allowNull: false}, as: 'groupeBranche'}); // Groupe has a FK id_branche
18  };
19
20  return Branche;
21 };

```

Figure 38: Table branche

Grâce à ce code, il est possible de voir que la table branche contient une colonne « id\_branche » qui est un identifiant unique dans la base de données et une colonne « brancheName » qui désigne le nom de la branche scolaire.

De plus, il est montré que la table Branche est liée à la table Groupe grâce à une méthode appelée « hasMany ». Cela signifie qu'une branche peut avoir plusieurs groupes/sous-catégories. La table groupe sera composée d'une clé étrangère « id\_branche ».

## MODULES

Le dossier module est utilisé pour stocker les méthodes JavaScript spécifiques à chaque classe. Ce sont dans ces fichiers que les actions sur la base de données sont faites telles que les insertions, les suppressions et les mises à jour.

Par exemple, la création d'une annonce par le directeur ou le superadministrateur utilisera la méthode *insertAnnonce(titre, text, auteur)* présente dans le fichier annonce.js.

```

insertAnnonce(titre, text, auteur){
  return new Promise(function (resolve, reject) {
    models.Annonce.create({
      annonceTitre: titre,
      annonceTexte: text,
      annonceAuteur: auteur
    }).then(function (annonce) {
      resolve(annonce)
    })
  })
},

```

Figure 39: Méthode insertAnnonce



Depuis ces méthodes, il est également possible d'appeler les données stockées dans la base et d'inclure les tables qui leur sont liées.

Par exemple, si l'utilisateur souhaite récupérer tous les professeurs stockés ainsi que leur information de connexion et leur localité, il exécutera la méthode `getAllProf()`.

```
getAllProf(){
  return new Promise(function(resolve, reject){
    models.Prof.findAll({
      include: [{
        model: models.EtablissementProf,
        as: 'profEtablissementProf'
      },{
        model: models.Login,
        as: 'loginProf'
      },{
        model: models.Localite,
        as: 'localiteProf'
      }]
    }).then(function(profs){
      resolve(profs)
    })
  })
},
```

Figure 40: Méthode `getAllProf`

Ces méthodes sont construites grâce au module Sequelize qui est ajouté au projet Skool.

## NODE\_MODULE

Ce dossier contient tous les modules supplémentaires ajoutés au projet Node.js tel que Sequelize (pour les appels à la base de données), Express (le Framework JavaScript choisi pour Skool) ou même JQuery (pour utiliser des méthodes en JQuery).

Ce dossier est obligatoire pour réussir à faire tourner le programme en local. Cependant, il ne doit pas être présent sur le serveur, car les modules externes sont téléchargés directement en ligne lors du déploiement.

## PUBLIC

Tous les documents reliés à la partie publique du programme sont sauvegardés dans ce dossier. En plus des fichiers utilisés pour le frontend (CSS), le dossier public contient les fichiers JavaScript qui permettent la manipulation de données.

Premièrement, le document « adminquery.js » est utilisé pour transmettre des informations de l'interface utilisateur vers le serveur grâce à des méthodes Ajax.

Par exemple, lorsque le superadministrateur entre des informations pour créer un nouvel établissement dans le formulaire, ces données sont récupérées et envoyées dans une méthode du serveur afin d'enregistrer une nouvelle entrée dans la base de données grâce à la méthode *addEtablissementSuperadmin ()*.

```
// Ajouter un nouvel établissement
function addEtablissementSuperadmin() {
    var nom = $('#nom_inline').val();
    var adresse = $('#adresse_inline').val();
    var npa = $('#npa_inline').val();
    var localite = $('#localite_inline').val();
    var idDirecteur = $('#directeurPick').val();

    $.ajax({
        url: "/superadmin/addEtablissement",
        type: "POST",
        data: {nom: nom, adresse: adresse, npa: npa, localite: localite, idDirecteur: idDirecteur},
        success: function(data) {
            console.log(data);
            setTimeout(function() {
                window.location.reload(true);
            }, 500);
        }
    })
}
```

Figure 41: Méthode addEtablissementSuperadmin

Deuxièmement, le document « selectquery.js » est utilisé afin d'enregistrer toutes les méthodes écrites en JQuery. Le JQuery permet de simplifier l'écriture de script dans les documents côté public/utilisateur. Le fichier « selectquery.js » est plus précisément utilisé dans le cadre du projet Skool afin de remplir les modales<sup>1</sup> avec des données.

<sup>1</sup> Les modales sont un type de composant du Framework CSS Materialize utilisé dans le projet Skool. Les modales sont des pop-up intégrées au programme.

Figure 42: Modale de modification d'un élève

Lorsqu'un utilisateur souhaite modifier les informations personnelles d'un élève, il va accéder à la modale de celui-ci. La méthode JQuery reliée à cette modale grâce à son identifiant unique, va retrouver les données qui sont enregistrées dans le bouton d'ouverture et va les insérer dans les champs de ladite modale.

```
// Création de la modal pour l'édition d'un élève
$('.modal#modalEditEleve').modal({
  dismissible: true,
  ready: function (modal, trigger) {
    var id = trigger.data("id");
    var lastname = trigger.data("lastname");
    var firstname = trigger.data("firstname");
    var tel = trigger.data("tel");
    var email = trigger.data("email");
    var adresse = trigger.data("adresse");
    var npa = trigger.data("npa");
    var localite = trigger.data("localite");
    var username = trigger.data("username");
    var password = trigger.data("password");
    var date = trigger.data("date");

    modal.find('input[id="id"]').val(id);
    modal.find('input[id="lastname"]').val(lastname);
    modal.find('input[id="firstname"]').val(firstname);
    modal.find('input[id="tel"]').val(tel);
    modal.find('input[id="email"]').val(email);
    modal.find('input[id="adresse"]').val(adresse);
    modal.find('input[id="npa"]').val(npa);
    modal.find('input[id="localite"]').val(localite);
    modal.find('input[id="date"]').val(date);
    modal.find('input[id="username"]').val(username);
    modal.find('input[id="password"]').val(password);
  }
});
```

Figure 43: JQuery pour la modale d'édition d'un élève

## ROUTES

Les fichiers du dossier routes permettent de configurer la redirection URL du site. Les différentes interfaces utilisateurs possèdent des chemins d'accès qui leur sont propres.

Il existe quatre types de méthodes dans les routes : GET, POST, DELETE et PUT. Les méthodes GET servent à appeler une page précise. Les méthodes POST vont, en premier lieu, exécuter une insertion

dans la base de données avant d'appeler une page. Les méthodes PUT et DELETE vont, quant à elles, exécuter des méthodes de suppression et de modification dans la base de données.

Avant d'appeler la page voulue, il est parfois nécessaire de récupérer des informations de la base de données afin de les afficher dans le côté client. Pour cela, les méthodes stockées dans le dossier modules sont utilisées.

Par exemple, afin de parvenir à la page de recherche avancée de l'interface professeur, il est nécessaire que le programme récupère toutes les classes appartenant à l'enseignant actuellement connecté. Il faut donc aller chercher l'objet professeur qui correspond au login de la session et, par la suite, récupérer toutes les classes de la base de données qui sont en lien avec cet utilisateur.

```
router.get('/recherches',function(req, res, next) {
  profModule.findOneProfWithIdLogin(req.session.user.id_login).then((prof)=>{
    classeModule.getAllClasseWithEleve(prof.id_prof).then((classes)=>{
      if(req.session.user.id_role === 2){ // Seul les prof accède à cette page
        res.render('prof_recherche', {backurl: "/prof", titre: 'Recherches avancées', classes:classes});
      }
      else{
        req.session.authenticated = false;
        res.redirect('/login');
      }
    })
  })
});
```

Figure 44: Route pour les recherches avancées – interface professeur

## VIEW

Le dernier dossier important du projet Skool est celui appelé view. Celui-ci contient tous les documents PUG de l'application qui sont appelés dans le programme.

Les documents PUG permettent la mise en page de nos différentes interfaces. Toutes les pages du programme sont basées sur des modèles appelés « layout ». La construction des interfaces est basée sur le Framework Materialize. Grâce à ce modèle, il est possible d'intégrer des composants, comme des barres de navigation avec onglet, plus simplement.

```

prof_classe.pug x
Enable File Watcher to compile Pug to HTML?
1 extends layout
2
3 block content
4   div.container(style="margin-top:50px")
5     div.row
6       br
7       font(size="3", color='red') #{messageErreur}
8     .row
9       ul.tabs(style="border-bottom: 1px groove")
10        each classe in classes
11          .classe
12            li.tab.col.s3(style="width: 100%")
13              a(href="#classe"+classe.id_classe)
14                value=classe.classeName
15

```

Figure 45: PUG de l'interface prof – mes classes

Dans cet exemple, il est possible de voir la construction d'un menu avec onglet. Il est construit dynamiquement suivant le nombre de classes dans lesquelles le professeur enseigne. Ci-dessous, voici le résultat de l'interface professeur avec deux onglets.

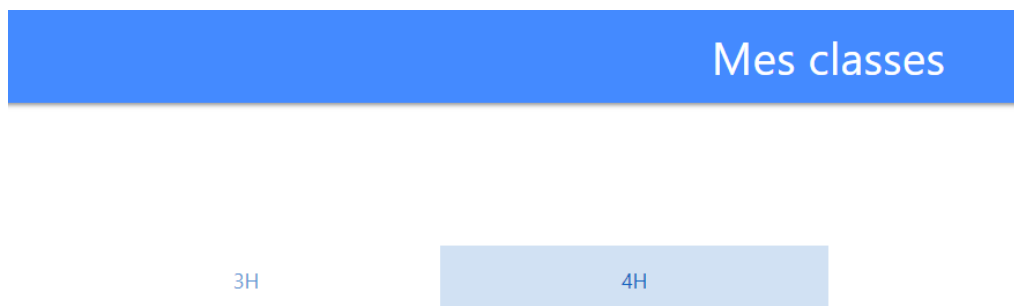


Figure 46: Résultat du PUG prof\_classe

### 5.2.3. HÉBERGEMENT

#### HEROKU

Skool est actuellement déployé grâce à la plateforme Heroku. « Fondé en 2007, Heroku a pour objectif de rendre le déploiement des applications de cloud computing de nouvelle génération aussi simples [sic] que leur développement. » (Montbeillard, 2010)

Le terme de « cloud computing » désigne le fait de stocker et d'accéder à des données d'une application via internet et non pas par le biais d'un disque dur local. Il est donc possible d'accéder à des données directement sur internet lorsque celles-ci sont enregistrées sur une plateforme de cloud applications (L, 2017).

Heroku est donc choisi pour stocker les données de Skool et permet de rendre l'application disponible sur le Net à l'adresse suivante : <https://skoolapp.herokuapp.com> .

Cette plateforme a été choisie pour le projet, car elle est gratuite et elle supporte le Node.js utilisé dans Skool. Il y a beaucoup de documentation disponible sur le site afin de faciliter le déploiement d'application et la communauté autour de cette plateforme est très présente.

## MÉTHODE DE DÉPLOIEMENT

Afin de déployer l'application sur Heroku, il faut exécuter plusieurs commandes dans le terminal de Webstorm. Grâce au module « git ». Si celui-ci n'est pas installé dans le programme, il faut taper la commande :

```
$ npm install --save git
```

Tout d'abord, il faut installer le CLI d'Heroku grâce à la commande disponible sur le site officiel à l'adresse suivante :

<https://devcenter.heroku.com/articles/getting-started-with-nodejs#set-up>

Une fois installé, il faut se connecter à son compte avec la commande :

```
$ heroku login
```

Le terminal demande de se connecter avec l'adresse email du compte ainsi que le mot de passe.

```
C:\Users\julie Préperier\WebstormProjects\Skool>heroku login
heroku: Enter your login credentials
Email [julie.preperier@gmail.com]: julie.preperier@gmail.com
Password: *****
Logged in as julie.preperier@gmail.com

C:\Users\julie Préperier\WebstormProjects\Skool>
```

Figure 47: Connexion au compte Heroku

Les crédits utilisés pour accéder à l'application de Skool sont l'adresse email de l'étudiant : [julie.preperier@gmail.com](mailto:julie.preperier@gmail.com) ainsi que du mot de passe : « Bachel0r2018. ».

Une fois la connexion réussite, il faut cloner le contenu du git de stockage de l'application dans un dossier en local. Le projet Skool est actuellement stocké sur GitLab à l'adresse : <https://gitlab.com/JuliePreperier/Skool.git> .

Afin de cloner ce git, il faut se déplacer grâce au terminal de commande pour d'atteindre l'emplacement voulu. De base, l'utilisateur se trouve dans le dossier contenant le projet ouvert dans Webstorm.

Dans le cadre du projet, le clone sera stocké dans un dossier nommé « Deploy ».

```
C:\Users\julie Préperier\WebstormProjects\Skool>cd ..  
  
C:\Users\julie Préperier\WebstormProjects>cd Deploy  
  
C:\Users\julie Préperier\WebstormProjects\Deploy>
```

Figure 48: Déplacement dans le dossier Deploy

Une fois dans le bon dossier local, il faut cloner le dossier grâce à la commande :

```
$ git clone https://gitlab.com/JuliePreperier/Skool.git
```

Cette commande va prendre quelques minutes afin de copier toutes les informations du projet cloné.

```
C:\Users\julie Préperier\WebstormProjects\Deploy>git clone https://gitlab.com/JuliePreperier/Skool.git  
Cloning into 'Skool'..  
remote: Counting objects: 10164, done.  
remote: Compressing objects: 100% (6943/6943), done.  
remote: Total 10164 (delta 2566), reused 10098 (delta 2521)  
Receiving objects: 100% (10164/10164), 14.85 MiB | 139.00 KiB/s, done.  
Resolving deltas: 100% (2566/2566), done.  
Checking connectivity... done.
```

Figure 49: Cloner git dans dossier local

Dès que le git est cloné dans un dossier local, la prochaine étape est de se déplacer dans le dossier cloner grâce au terminal de commande et de déployer l'application sur Heroku.

Afin d'atteindre le git d'Heroku appartenant à l'application « skoolapp », il est nécessaire de se connecter à la branche distante (remote branche) de Skool grâce à la commande suivante :

```
$ heroku git:remote -a skoolapp
```

Cette commande va informer Webstorm que le git que l'utilisateur veut atteindre se trouve dans l'application nommée skoolapp.

```
C:\Users\julie Préperier\WebstormProjects\Deloy\Skool>heroku git:remote -a skoolapp  
set git remote heroku to https://git.heroku.com/skoolapp.git
```

Figure 50: Configuration du git de l'application Heroku

Il est maintenant d'envoyer tout le contenu du dossier cloné dans le stockage de l'application Heroku. Tout ce qui est dans ce dossier sera utilisé pour l'application en ligne.

Pour l'envoi des données, il faut taper la commande :

```
$ git push heroku master
```

Cette action va pousser les données dans le git et, en plus, déployer l'application en ligne.

Il est possible qu'un message d'erreur apparaisse lors de la commande précédente.

```
C:\Users\julie Préperier\WebstormProjects\Deloy\Skool>git push heroku master  
remote: ! WARNING:  
remote: ! Do not authenticate with username and password using git.  
remote: ! Run `heroku login` to update your credentials, then retry the git command.  
remote: ! See documentation for details: https://devcenter.heroku.com/articles/git#http-git-authentication  
fatal: Authentication failed for 'https://git.heroku.com/skoolapp.git/'
```

Figure 51: Erreur de connexion

Si cela arrive il est possible de se connecter au git grâce à un jeton d'identification, aussi appelé « token ». Pour obtenir un token la commande est :

```
$ heroku auth:token
```

Le terminal affichera un code alphanumérique d'une trentaine de caractères.

```
C:\Users\julie Préperier\WebstormProjects\Deloy\Skool>heroku auth:token  
» Warning: token will expire 07/20/2019  
» Use heroku authorizations:create to generate a long-term token  
2751e4bc-b2af-48ac-bd24-47d04ecb8a51
```

Figure 52: Token Heroku



Il est important de copier le token et de réexécuter la commande « push » écrite ci-dessus. Un pop-up Windows apparaîtra alors sur l'écran et le token doit être collé dans le champ destiné au mot de passe. (Aucun nom d'utilisateur n'est requis)

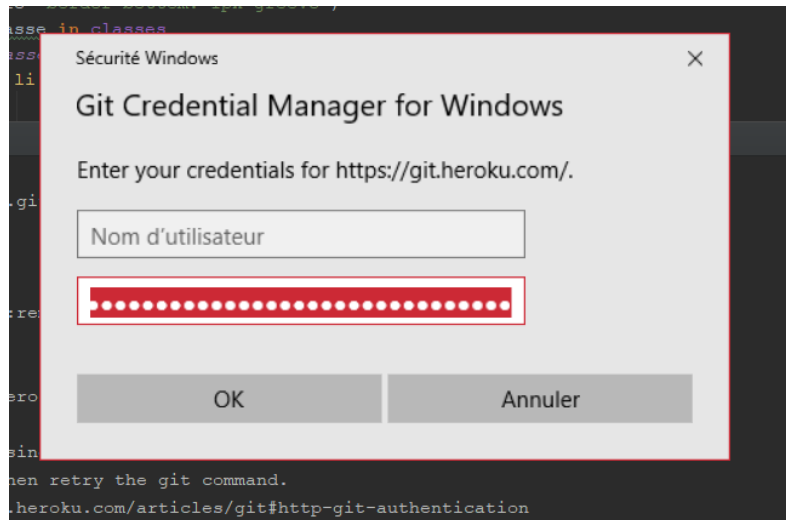


Figure 53: Connexion au git Heroku

Dès que la connexion est établie, le déploiement commence. Une fois fini, il est possible d'accéder à l'application en tapant la commande :

```
$ heroku open
```

Et voilà, l'application est déployée sur l'URL <https://skoolapp.herokuapp.com/>.

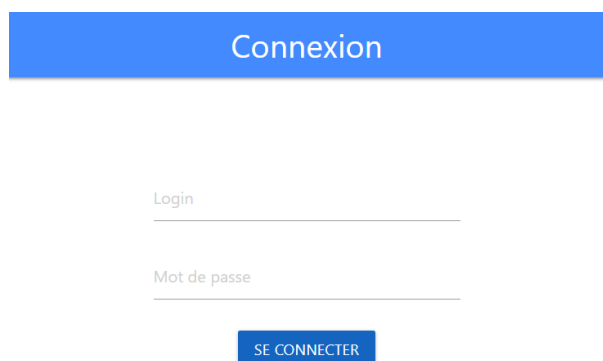


Figure 54: Page de connexion de Skool

#### 5.2.4. CONNEXION

Lors du premier déploiement de Skool, la base de données n'est pas remplie. L'unique moyen de connexion possible est l'identifiant du superadministrateur.

Nom d'utilisateur : sysadmin

Mot de passe : password

Cet identifiant est créé dans le document `www.js` du projet `Node.js`. Il donne accès à l'interface superadministrateur de Skool. Il n'est pas possible de renseigner ici les autres noms d'utilisateurs du programme, car ceux-ci sont créés en même temps que l'utilisateur est créé.

Les comptes des utilisateurs suivent tous le même modèle, à l'exception du superadministrateur. La première partie du nom d'utilisateur représente son rôle (« ens » pour enseignant, « dir » pour directeur et « el » pour élève). Il est suivi d'un point et d'une concaténation des deux premières lettres de son prénom et de la première lettre de son nom de famille.

Par exemple, lors de la création d'un professeur du nom de John Doe, un identifiant est généré automatiquement : `ens.jod`.

De base, les mots de passe générés sont « 1234 ». Il est tout à fait possible de modifier ces informations de connexion à tout moment.

#### 5.2.5. GESTION DES ERREURS

Étant donné que l'application Skool est un prototype, la gestion des erreurs sur l'interface graphique n'est pas complète. Il est possible que l'application réfléchisse dans le vide lors d'une action ou d'un renvoi sur une page d'erreur.

Afin de trouver l'erreur, il est possible d'exécuter une commande dans le terminal de Webstorm :

```
$ heroku logs --tail
```

Cette commande affiche tous les logs<sup>2</sup> de l'application. Grâce à cette vision, il est plus simple de découvrir le problème en cours.

<sup>2</sup> Un log est une entrée dans un journal d'évènement.

### 5.3. GUIDE D'UTILISATEUR

Le guide d'utilisation permet de faciliter la compréhension des différentes interfaces et mécaniques du site de l'application Skool. Il existe actuellement quatre grands axes dans l'application.

Premièrement, l'interface du professeur qui permet à celui-ci d'accéder aux élèves dont il a la responsabilité et de faire des évaluations des objectifs soit individuellement soit pour toute une classe. Le professeur a également accès à une page d'annonces postées par les directeurs des écoles ou par le superadministrateur.

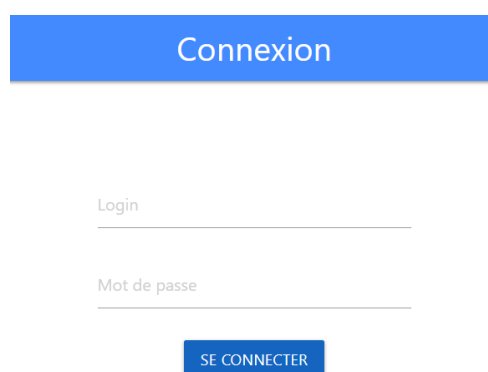
Deuxièmement, l'interface de l'élève qui est aussi destinée aux parents. Ceux-ci pourront changer les informations personnelles de l'élève et visualiser les résultats des évaluations.

Troisièmement, l'interface des directeurs qui a plus une fonction d'administration et de gestion des établissements dont ils sont rattachés. Par exemple, la création de classes, l'attribution de celles-ci aux profs travaillant dans l'établissement, etc.

Dernièrement, l'interface du superadministrateur qui a également pour but l'administration et la gestion des données. Cependant, à la différence de la gestion d'un directeur, un superadministrateur configure des données plus générales telles que la création d'établissements ou d'utilisateurs.

#### 5.3.1. INTERFACE DE CONNEXION

Lorsque l'utilisateur souhaite atteindre l'application de Skool, il arrive sur la page de connexion. Selon l'identifiant qu'il entre, le programme le redirige vers la bonne interface..



Connexion

Login

Mot de passe

SE CONNECTER

Figure 55: Connexion

### 5.3.2. INTERFACE PROFESSEUR

#### ACCUEIL

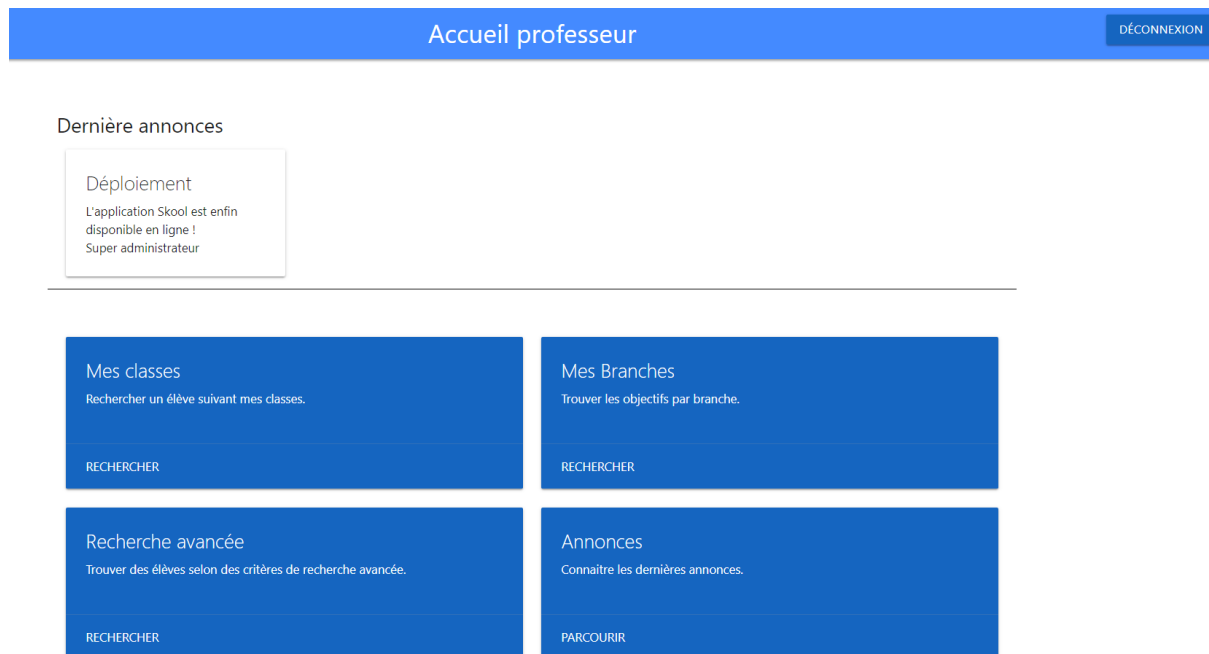


Figure 56: Page d'accueil du professeur

Sur la page d'accueil de l'interface prof, l'utilisateur peut voir les quatre dernières annonces postées sur l'application.

De plus, il a accès à quatre menus différents. En premier, un accès à la liste de ses classes. Ensuite, un accès vers une liste des branches existantes. Le troisième accès est celui vers une interface de recherches avancées dans la liste des élèves. Et, finalement, le professeur a accès à la page contenant toutes les annonces postées par les directeurs et le superadministrateur.

#### ANNONCES

La page dédiée aux annonces permet au professeur de se tenir au courant des dernières nouvelles des différents établissements scolaires. Il ne possède aucun droit d'écriture sur celle-ci.



Figure 57: Interface d'annonces

## RECHERCHES AVANCÉES

La page de recherches avancées permet au professeur de trouver rapidement un élève selon son nom de famille, son prénom, sa classe ou même son établissement scolaire. Le tableau en dessous du champ de recherche est trié de manière dynamique selon les filtres entrés.

Nom	Prénom	Classe	Etablissement	Action
Test	Charlotte	3H	Ecole primaire	<a href="#">EDITER</a>
Modèle	Charlie	4H	Ecole primaire	<a href="#">EDITER</a>
Modèle	Andréa	4H	Ecole primaire	<a href="#">EDITER</a>

Figure 58: Recherches avancées

## MES CLASSES

L'enseignant connecté peut aller rechercher un élève selon une classe précise. Pour cela, l'interface des classes affiche une liste d'élèves selon leur classe. Il est possible pour le professeur de se déplacer au travers des différentes classes grâce aux onglets présents au sommet de la page.

Nom	Prénom	Action
Test	Célie	<a href="#">EDITER</a>
Test	Enzo	<a href="#">EDITER</a>
Modèle	Charlie	<a href="#">EDITER</a>
Modèle	Andréa	<a href="#">EDITER</a>

Figure 59: Recherche par classe

## MES BRANCHES

Le professeur peut, s'il le souhaite, rechercher un objectif précis qu'il souhaite évaluer pour toute une classe. L'interface des branches présente toutes les branches, groupes et objectifs existants dans la base de données.

The screenshot shows the 'Mes branches' interface. At the top, there is a blue header with 'Mes branches' and a 'DECONNEXION' button. Below the header, there is a navigation bar with '← ACCUEIL'. The main content area is titled 'Capacités transversales' and contains a table with the following data:

Id	Nom	Action
30	Prise en compte de l'autre	<a href="#">EVALUER</a>
31	Connaissance de soi	<a href="#">EVALUER</a>
32	Action dans le groupe	<a href="#">EVALUER</a>

Below the table, there are sections for 'Communication' (Stratégie d'apprentissage, Pensée créatrice, Démarche réflexive) and 'Mathématique'.

Figure 60: Recherche d'objectif par branches

Une fois l'objectif sélectionné, le prof est redirigé vers une nouvelle page pour l'évaluation de toute une classe.

The screenshot shows the 'Evaluation d'une classe' interface. At the top, there is a blue header with 'Evaluation d'une classe' and a 'DECONNEXION' button. Below the header, there is a navigation bar with '← ACCUEIL'. The main content area has two tabs: '3H' and '4H'. Below the tabs, there is a table with the following data:

Nom	Prénom	Action
Test	Yann	<input type="radio"/> 0 - Pas évalué <input type="radio"/> 1 - Non acquis <input type="radio"/> 2 - Partiellement acquis <input type="radio"/> 3 - Acquis <input type="radio"/> 4 - Acquis avec aisance
Test	Loanne	<input type="radio"/> 0 - Pas évalué <input type="radio"/> 1 - Non acquis <input type="radio"/> 2 - Partiellement acquis <input type="radio"/> 3 - Acquis <input type="radio"/> 4 - Acquis avec aisance
Test	Charlotte	<input type="radio"/> 0 - Pas évalué <input type="radio"/> 1 - Non acquis <input type="radio"/> 2 - Partiellement acquis <input type="radio"/> 3 - Acquis <input type="radio"/> 4 - Acquis avec aisance

Figure 61: Évaluation d'une classe

## FICHE ÉLÈVE ET ÉVALUATION

Qu'il passe via l'interface de recherches avancées ou de classe, une fois qu'il a choisi un élève, le professeur est redirigé vers la fiche personnelle de l'enfant. Sur celle-ci, il peut modifier les informations personnelles telles que le nom, prénom ou adresse.

De plus, toutes les branches sont affichées avec leurs objectifs. L'enseignant connecté peut sélectionner un objectif afin de faire une évaluation. Il sera alors redirigé vers une page prévue à cet effet.

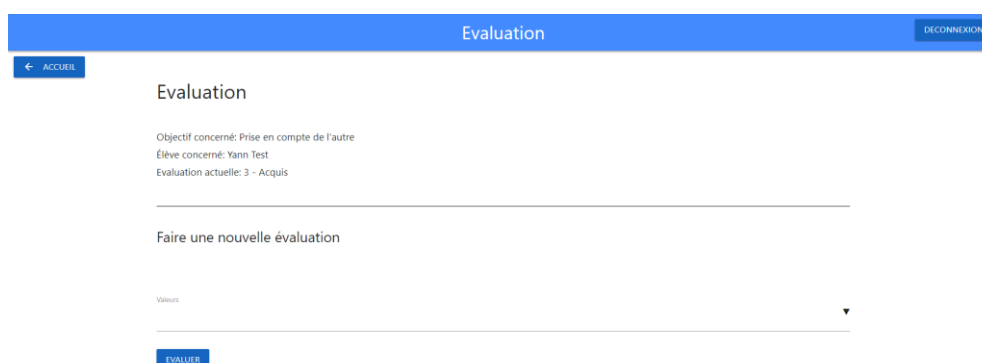


Figure 62: Évaluation d'un élève

### 5.3.3. INTERFACE PARENT/ÉLÈVE

#### ACCUEIL

L'interface d'accueil ressemble exactement à la fiche d'élève atteint par le professeur. La seule différence est que l'utilisateur connecté – élève/parent – ne peut pas modifier les évaluations. Cette partie de la page ne montre que la progression de l'élève à titre informatif.

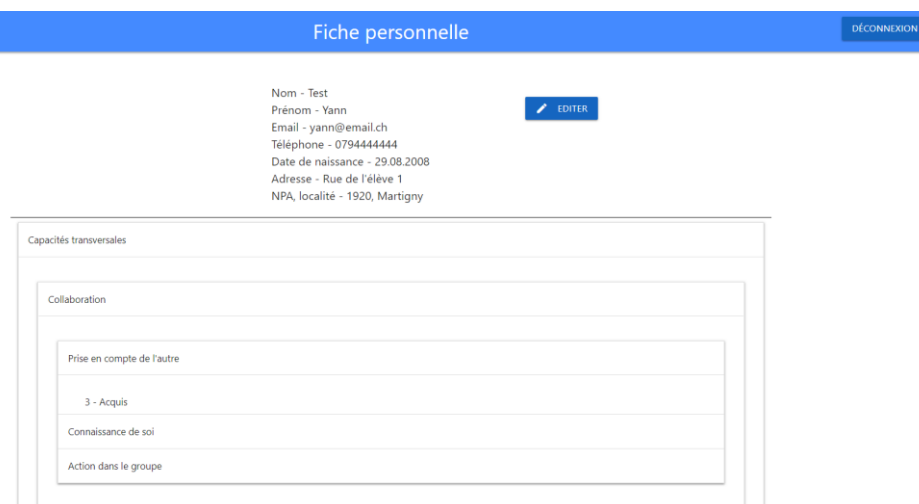


Figure 63: Accueil élève

### 5.3.4. INTERFACE DIRECTEUR

#### ACCUEIL

L'accueil d'un directeur est construit sur le même modèle que celui d'un professeur. Les quatre dernières annonces postées sont affichées au sommet de la page et quatre choix s'offrent à l'utilisateur.

À la différence d'une interface d'un enseignant, le directeur peut rechercher un élève grâce à une recherche avancée ou selon une liste de classe présente dans tous les établissements qu'il dirige.

Il a aussi accès à la liste complète des annonces présentes sur l'application. Cependant, le directeur peut rédiger une nouvelle annonce quand il le souhaite.

Pour finir, la quatrième option est une interface administrative permettant la gestion des établissements et des classes.

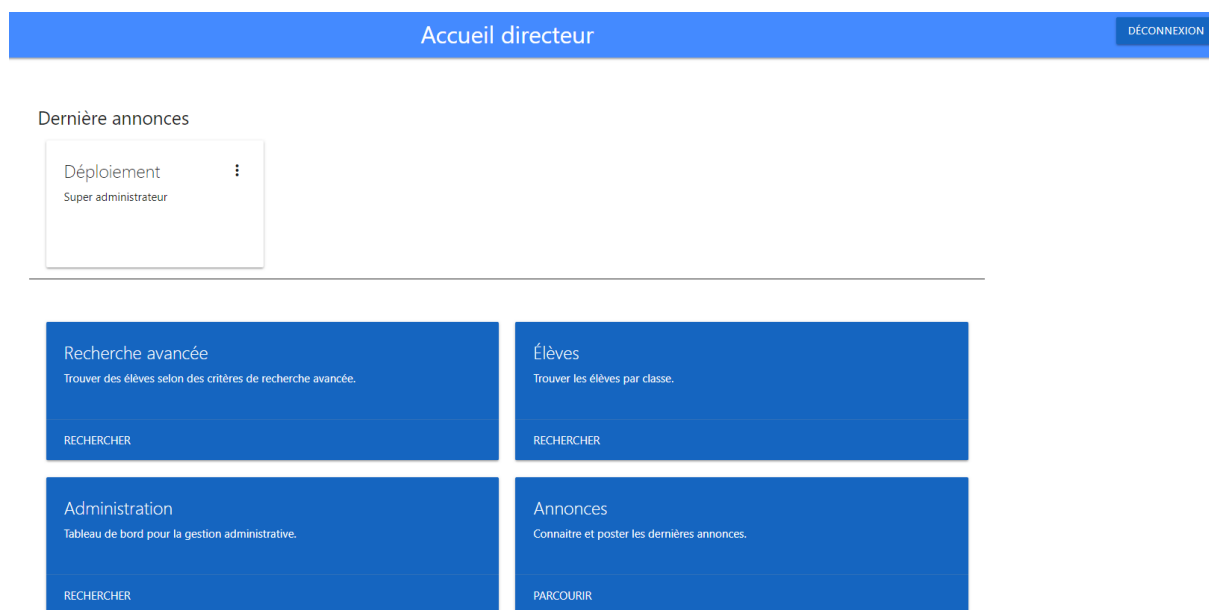


Figure 64: Accueil interface directeur



## ADMINISTRATION

Le directeur ne peut changer des informations concernant uniquement les établissements qu'il dirige. La plateforme administrative offre sept options de gestion.

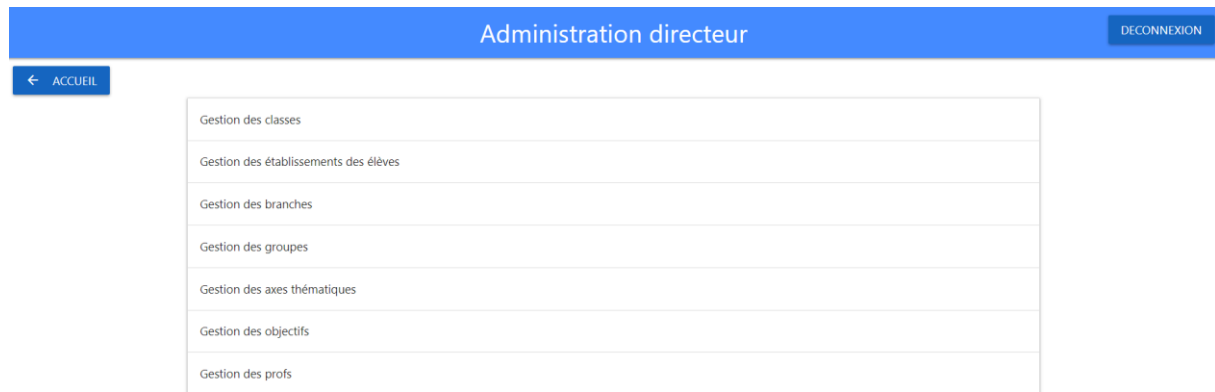


Figure 65: Interface d'administration d'un directeur

Les actions concernant les classes sont :

- La création de classe
- L'édition des classes
- L'ajout ou la suppression d'une année scolaire à une classe
- L'ajout ou la suppression d'un élève dans une classe

Lors de la création d'une classe, il est demandé d'ajouter une année. Seulement, il est possible qu'une école n'ait pas beaucoup d'élèves et, de ce fait, décide de mélanger des élèves de plusieurs degrés dans la même classe. C'est dans ces situations qu'une option de rajout d'année est proposée dans l'administration d'un directeur.

Les actions possibles concernant les établissements sont :

- Le retrait ou l'ajout d'un élève dans un établissement

Les directeurs n'ont pas accès aux établissements qu'ils ne dirigent pas. C'est pour cela qu'ils ne peuvent que retirer un élève de leur établissement ou aller chercher dans une liste un élève qui ne possède pas d'établissement et le rajouter directement dans une classe.

Le fait de l'ajouter dans une classe va enregistrer automatiquement l'établissement concerné et le degré de scolarité en lien avec la classe dans le profil de l'élève.

Le directeur est également responsable des branches, groupe et objectif. Grâce à l'interface d'administration, il peut créer, éditer et supprimer les branches qu'il souhaite. Les axes thématiques

sont créés automatiquement lors de la création d'un groupe. Il est toutefois possible pour le directeur d'éditer ou de supprimer un axe.

Les actions concernant la gestion des professeurs sont :

- L'ajout ou le retrait d'un professeur dans un établissement
- L'ajout ou le retrait d'un professeur dans une classe

### 5.3.5. INTERFACE SUPERADMINISTRATEUR

#### ACCUEIL

L'accueil du superadministrateur ressemble aussi à l'interface professeur et directeur à la seule différence qu'il ne possède que trois possibilités d'action.

La première, tout comme le professeur et le directeur, il peut rechercher un élève via un formulaire de recherches avancées. Cependant, le superadministrateur aura accès à tous les élèves du système et ne sera pas limité. La deuxième option est l'accès aux annonces et à la possibilité d'en créer de nouvelles. La dernière option est l'interface d'administration.

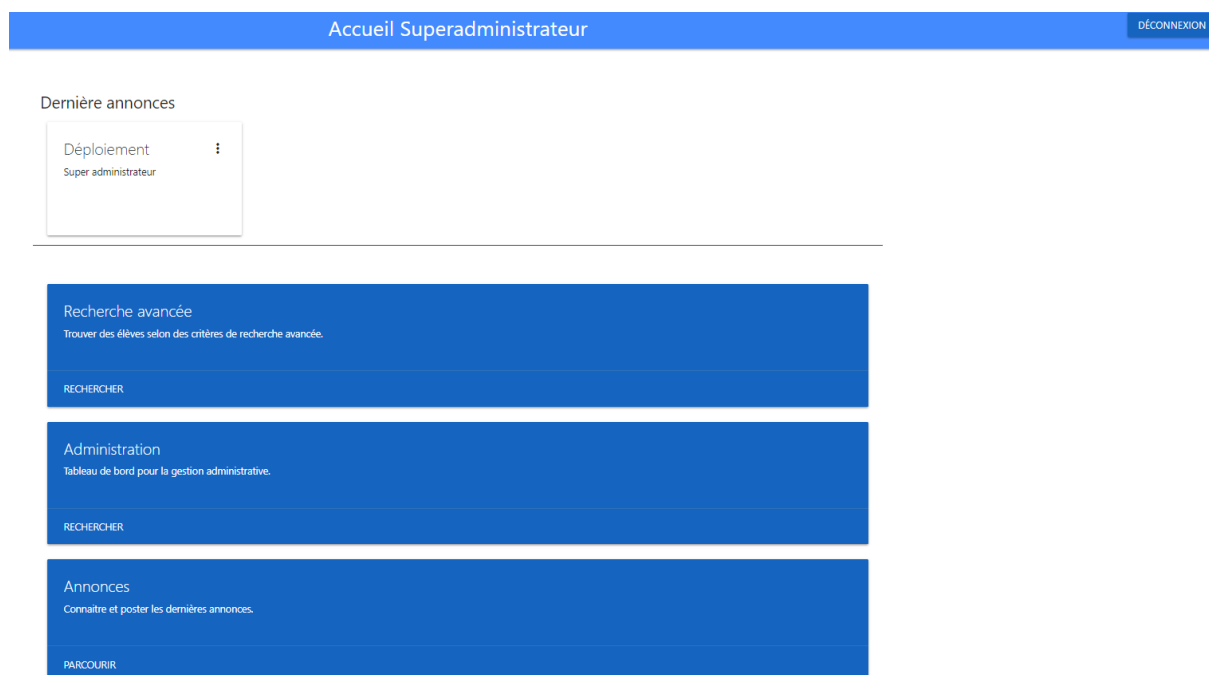


Figure 66: Accueil superadmin

## ADMINISTRATION

Le superadministrateur a la possibilité de gérer toutes les informations générales du programme. Il effectue la création de nouveau compte élève, professeur ou directeur, la création de nouveaux établissements ou de nouvelles localités.

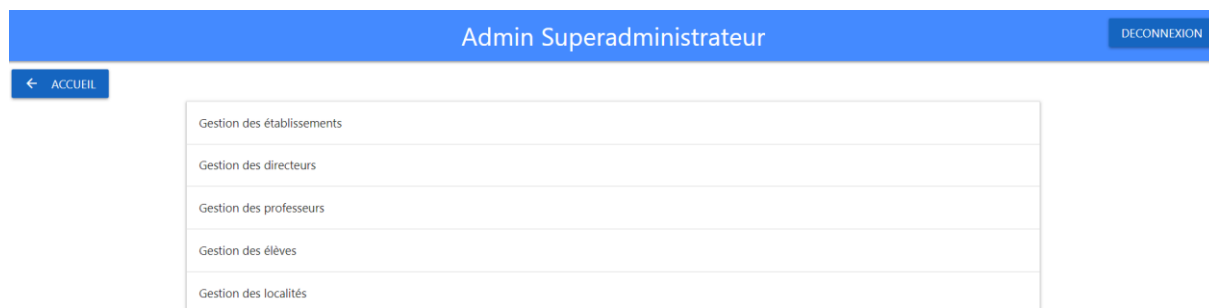


Figure 67: Interface d'administration du superadministrateur

Les actions concernant les établissements sont :

- Création, édition ou suppression d'un établissement
- Ajout d'un professeur dans un établissement

Une fois qu'un enseignant appartient à un établissement, il est pris en charge par le directeur dudit établissement.

Les uniques actions concernant les professeurs et les directeurs sont la création, l'édition ou la suppression d'un directeur/professeurs.

En ce qui concerne les élèves, les actions sont :

- Création, édition ou suppression d'un élève
- Gestion de l'établissement d'un élève
- Gestion d'une année scolaire d'un élève

À la différence d'un directeur, le superadministrateur peut changer l'établissement d'un élève, car il a accès à tous les établissements créés dans le système.

Le superadministrateur peut créer directement une localité via le formulaire et a la possibilité de la modifier. Cependant, il ne peut pas supprimer de localités, car, même si elle n'est pas utilisée dans un profil d'utilisateur, une localité n'a pas besoin d'être supprimée et peut être réutilisée plus tard.

### 5.3.6. COMMUN

Il y a trois interfaces qui sont communes au superadministrateur et au directeur.

#### ANNONCES

La page affichant toutes les annonces du système est la même chez le superadministrateur et chez les directeurs. Tous ont la possibilité de lire les annonces existantes et d'ajouter de nouvelles annonces et même de supprimer les plus anciennes.

Annonces superadministrateur

← ACCUEIL

Ajouter une nouvelle annonce

Titre de l'annonce

Texte de l'annonce

AJOUTER ANNONCE

Déploiement

L'application Skool est enfin disponible en ligne !

Super administrateur Fri Jul 20 2018 14:01:29 GMT+0200 (Paris, Madrid (heure d'été))

DECONNEXION

Figure 68: Interface d'annonce

#### RECHERCHES AVANCÉES

Comme mentionné ci-dessus, le directeur, le superadministrateur et même le professeur peuvent accéder aux fiches des élèves grâce à une interface de recherches avancées. Le seul détail qui différencie les trois rôles est l'accès aux données des élèves. Un professeur ne verra que les élèves des classes dans lesquelles il enseigne, le directeur uniquement les élèves faisant partie des établissements qu'il dirige et le superadministrateur aura accès à tous les élèves.

## FICHE ÉLÈVE

La fiche élève à laquelle accède le directeur et le superadministrateur est basée sur le même modèle que celle accédée par les enseignants et l'accueil d'un élève.

Cependant, les directeurs et le superadministrateur ne possèdent pas de droit d'écriture sur ces fiches à l'exception des informations personnelles de l'élève.

Fiche personnelle: Yann Test

← ACCUEIL

DECONNEXION

Nom - Test  
Prénom - Yann  
Email - yann@email.ch  
Téléphone - 0794444444  
Date de naissance - 29.08.2008  
Adresse - Rue de l'élève 1  
NPA, localité - 1920, Martigny

EDITER

Capacités transversales

Collaboration

Prise en compte de l'autre

3 - Acquis

Connaissance de soi

Action dans le groupe

Figure 69: Fiche élève directeur/superadministrateur

## 6. AMÉLIORATIONS FUTURES

Au début du projet Skool, un Product Backlog a été rédigé avec le client. Malheureusement, due au temps donné pour le travail de Bachelor, toutes les fonctionnalités n'ont pas pu être réalisées et intégrées dans le logiciel Skool.

### 6.1. INTÉGRATION DE JUSTIFICATIFS AUX ÉVALUATIONS

Lors d'une évaluation, le professeur pourrait avoir la possibilité de télécharger un justificatif de son évaluation. Par exemple, un travail écrit de l'élève, une vidéo ou une image.

Afin d'intégrer cette fonctionnalité dans le programme, il faudrait rajouter une table dans la base de données. Cette table gèrerait les objets de type BLOB (Binary Large Object) et la clé primaire de cette entrée sera liée à une évaluation.

De plus, il faudrait ajouter un bouton de téléchargement qui permettrait à l'enseignant de sélectionner un document sur son ordinateur et de l'envoyer dans la base de données.

### 6.2. GRAPHIQUES DE VISUALISATION

Au début de projet Skool, l'idéateur a imaginé la possibilité d'avoir un aperçu global de l'avancée des élèves. Ces graphiques se trouveraient dans le profil de chaque enfant et montreraient, d'une manière plus visuelle, où en est la progression de l'élève. En plus d'une vision individuelle, le client pensait à afficher une vision globale de l'avancée d'une classe complète.

Pour l'intégration de cette fonctionnalité, il faudrait installer un module qui aiderait à la création de graphiques tel que Vega.

### 6.3. EXPORT

L'export des fiches personnelles des élèves est une fonctionnalité facultative ajoutée dans le Product Backlog de Skool.

L'objectif de cette fonctionnalité est que les parents, professeurs et directeurs puissent exporter les informations des élèves ainsi que leurs résultats aux évaluations afin de les imprimer ou de les enregistrer en local.

La difficulté de cette fonctionnalité est de créer un document avec les informations présentes sur la page web. Le but étant d'exporter un document officiel, celui-ci ne pourrait pas simplement ressembler à la page web. Un travail de mise en page est nécessaire à cette fonctionnalité.

## 6.4. SÉCURITÉ DES DONNÉES

### 6.4.1. CRYPTAGE DES DONNÉES

En l'état, la base de données de Skool n'est pas sécurisée pour les informations sensibles. Les mots de passe sont lisibles et ne requièrent aucune sécurité minimum.

Une amélioration pour l'application serait de crypter les entrées de la base de données afin que celles-ci ne soient pas accessibles trop facilement. De plus, il est possible de rajouter une sécurité quant à la force du mot de passe de l'application afin que les chaînes de caractère tel que « 1234 » ne soient plus possibles. Par exemple, un nombre de caractères minimum, obliger l'utilisation de chiffre et de lettre et d'obligation d'utiliser des caractères spéciaux.

### 6.4.2. DÉDOUBLEMENT D'INFORMATION DE CONNEXION

Actuellement, dans l'application, si deux personnes possèdent le même nom d'utilisateur (deux premières lettres du prénom et première lettre du nom de famille), le programme ne fera pas de différence entre les deux et créera les deux entrées.

Une solution possible serait qu'un message d'avertissement apparaisse lors de la création d'un nouveau compte. Ce message informerait que le nom d'utilisateur n'est pas possible. Une création automatique d'un nom de secours peut être intégrée au programme. Par exemple, le logiciel prendrait les deux premières lettres du nom de famille et du prénom.

## 6.5. GAMIFICATION POUR L'INTERFACE ÉLÈVE

Sur le document de données du travail de Bachelor, l'idée de créer une option de gamification pour les élèves était inscrite.

Cette idée était d'améliorer l'interface des élèves avec un système de petit jeu. Un personnage à faire évoluer au fur et à mesure qu'ils maîtrisent leurs objectifs les accompagnerait tout au long de leur scolarité et les encouragerait à acquérir les différentes compétences.

## 6.6. ÉVOLUTION DES OBJECTIFS

Les objectifs du PER ne sont pas des entrées individuelles. Généralement, un objectif est progressif et suit un schéma évolutif.

Par exemple, l'objectif « Identifier, classer et nommer des formes planes simples » devient « Identifier, classer, nommer et décrire des formes planes simples selon le nombre d'angles droits et les côtés de même mesure, en utilisant un vocabulaire personnel ». Ce deuxième objectif se trouve

dans la base de données comme étant un but à atteindre par l'élève. Cependant, tant que l'élève n'aura pas acquis la première compétence, il ne lui est pas possible d'acquérir la deuxième.

Pour le moment, l'application Skool ne fait pas de distinction entre les objectifs. L'amélioration future serait d'afficher les objectifs qui sont premier de leur schéma de progression et d'afficher les objectifs suivant uniquement si l'élève a atteint la première étape.

La base de données de Skool est déjà construite dans ce sens. Dans la table « Objectif », il existe une colonne « objectifSuivant » qui désigne quel est la prochaine étape de l'objectif. Une deuxième colonne, appelée « premierListe » fait partie de cette table et celle-ci permet de préciser si l'entrée est la première d'une série ou non.



## CONCLUSION

Ce projet s'est terminé avec le développement d'un prototype en ligne qui répond à la majorité des demandes établies en début de travail.

À la suite de l'analyse des outils existant, il est démontré clairement qu'un outil permettant l'évaluation des objectifs du PER n'existe pas. Ou, du moins, aucun outil ne répond à tous les besoins apportés par l'idéateur. Actuellement, certains objectifs sont toujours évalués dans des carnets de correspondance papier que l'élève doit garder avec lui.

Ce nouvel outil peut apporter une interface centralisée pour toute la Suisse romande. Toutes les écoles pourraient évaluer leurs élèves de la même manière et le transfert d'information serait plus facile.

Par exemple, si un enfant habite dans le canton de Vaud et déménage en Valais, son compte sera simplement d'un établissement à un autre. Toutes ses évaluations sont liées à son profil. Cela réduira l'administration de son transfère concernant sa progression.

Ce travail m'a permis de renforcer mes connaissances en programmation Node.js, à apprendre une logique concernant les parties d'un programme web (frontend, backend) et le transfert d'information entre les deux. Le travail par itération est une méthode de travail adapté à mon organisation. Cette méthodologie m'a permis de structurer ma progression et de ne pas m'égarer dans la construction du projet.

Skool n'est, pour le moment, qu'au stade de prototype mais, qui sait, peut-être qu'un jour il fera partie intégrante du quotidien de tous les enseignants romands.

## RÉFÉRENCES

- Alioze. (s.d.). *Agence Express JS*. Récupéré sur Alioze: <https://www.alioze.com/express-js>
- Andanson, M. (2016, 10 31). *Supinfo*. Récupéré sur AngularJS - Présentation: <https://www.supinfo.com/articles/single/3528-angularjs-presentation>
- Angular. (s.d.). *Angular.io*. Récupéré sur Angular: <https://angular.io/>
- Audibert, L. (s.d.). *UML 2 - De l'apprentissage à la pratique*. Récupéré sur Developpez.com: <https://laurent-audibert.developpez.com/Cours-UML/?page=introduction-modelisation-objet#L1-4-3-a>
- Audrey. (2007, 04 02). *Le langage Html*. Récupéré sur Gralon: <https://www.gralon.net/articles/internet-et-webmaster/creation-site-internet/article-le-langage-html-260.htm>
- Ben Debba, N., & Bonneau, J. (2015). *Introduction à MongoDB*. Récupéré sur GitBook: <https://legacy.gitbook.com/book/mongoteam/introduction-a-mongodb/details>
- Berroukeche, H. (2016, 10 20). *Materialize: Un Framework CSS*. Récupéré sur SupInfo: <https://www.supinfo.com/articles/single/2862-materialize-framework-css>
- Billey, V. (2016, 04 21). *Introduction à React*. Récupéré sur Synbioz: [https://www.synbioz.com/blog/introduction\\_a\\_react](https://www.synbioz.com/blog/introduction_a_react)
- Bitschené, Q. (2014, 03 11). *Atelier IHM*. Récupéré sur Atelier IHM: [http://atelierihm.unice.fr/wp-content/uploads/sites/3/2014/11/Comparaison\\_bitschene.pdf](http://atelierihm.unice.fr/wp-content/uploads/sites/3/2014/11/Comparaison_bitschene.pdf)
- Bootstrap. (s.d.). *Browsers and devices*. Récupéré sur Bootstrap: <https://getbootstrap.com/docs/4.0/getting-started/browsers-devices/>
- Boyer, S. (2016, 05 18). *La méthodologie Scrum expliquée simplement - Le Guide Ultime de la Méthode Agile Scrum*. Récupéré sur nutcache: <https://www.nutcache.com/fr/blog/methodologie-scrum/>
- Chavelli, M. (2018, 05 04). *Prenez en main Bootstrap*. Récupéré sur OpenClassroom: <https://openclassrooms.com/courses/prenez-en-main-bootstrap?status=published>
- Chrzanowska, N. (2017, 03 17). *12 Top Applications Written in Node.js - Examples from Big Companies*. Récupéré sur Netguru: <https://www.netguru.co/blog/top-companies-used-nodejs-production>

- Chrzanowska, N. (2017, 09 07). *Node.js vs. PHP: Which Environment To Choose For Your Next Project?* Récupéré sur netguru: <https://www.netguru.co/blog/nodejs-vs-php>
- CIIP. (2012, Juin). *Plan d'études romand (PER) - Aperçu des contenus - Cycle 1*. Récupéré sur Plan d'études romand: [https://www.plandetudes.ch/documents/10136/19192/cycle\\_1\\_webCIIP.pdf](https://www.plandetudes.ch/documents/10136/19192/cycle_1_webCIIP.pdf)
- Confédération Suisse. (s.d.). *Durée et organisation de la scolarité obligatoire*. Récupéré sur ch.ch: <https://www.ch.ch/fr/duree-ecole-obligatoire/>
- de la Mark, S., & Paradanaud, J. (2012). *Dynamisez vos sites web avec JavaScript*. OpenClassrooms - ex-Site du Zéro.
- Edumoov. (s.d.). *Livret Scolaire Numérique*. Récupéré sur Edumoov: [https://www.edumoov.com/pages/livret\\_scolaire\\_numerique](https://www.edumoov.com/pages/livret_scolaire_numerique)
- Evalog. (s.d.). *Evalog, c'est quoi?* Récupéré sur Evalog: <http://evalog.ch/zwook/textes/evalog-c-est-quoi>
- Florent, L. (s.d.). *Introduction aux méthodes agiles et Scrum*. Récupéré sur L'agiliste: <https://www.agiliste.fr/introduction-methodes-agiles/>
- Foundation. (s.d.). *Responsive design gets a whole lot faster*. Récupéré sur Zurb Foundation: <https://foundation.zurb.com/>
- Futura Tech. (s.d.). *MySQL*. Récupéré sur Futura Tech: <https://www.futura-sciences.com/tech/definitions/internet-mysql-4640/>
- Galiana, D. (2017, 07 04). *Qu'est-ce que la méthodologie Waterfall ?* Récupéré sur Planzone: <https://www.planzone.fr/blog/quest-ce-que-la-methodologie-waterfall>
- Heuveline, M. (2017, 07 24). *Brève présentation de Bootstrap*. Récupéré sur Supinfo: <https://www.supinfo.com/articles/single/4823-breve-presentation-bootstrap>
- Iqra, A. (2017, 11 26). *Pug vs HTML*. Récupéré sur C# Corner: <https://www.c-sharpcorner.com/article/pug-vs-html/>
- Kazankov, V. (2017, 10 12). *PHP VS NODE.JS*. Récupéré sur Belitsoft: <https://belitsoft.com/php-development-services/php7-vs-nodejs>
- L, B. (2017, 02 10). *Cloud Computing - Définition, avantages et exemples d'utilisation*. Récupéré sur Le Big Data: <https://www.lebigdata.fr/definition-cloud-computing>

- Laurent, S. (2016, 11 4). *Développement: faut-il se tourner vers Node.JS à l'avenir ?* Récupéré sur Alioze: <https://www.alioze.com/developpement-nodejs-2017>
- Lazzaroni, T. (2017). *Les meilleurs frameworks css pour vos sites.* Récupéré sur Graphiste: <https://graphiste.com/blog/meilleurs-frameworks-css>
- Lucas, S. (2014, 01 14). *Node.js : la tendance JavaScript côté serveur.* Récupéré sur Journal du Net: <https://www.journaldunet.com/developpeur/expert/56253/node-js---la-tendance-javascript-cote-serveur.shtml>
- LucidChart. (s.d.). *Qu'est-ce qu'un schéma de base de données ?* Récupéré sur LucidChart: <https://www.lucidchart.com/pages/fr/quest-ce-quun-sch%C3%A9ma-de-base-de-donn%C3%A9es>
- Martsoukos, G. (2015, 01 21). *Getting Started with Skeleton, the Simple CSS Boilerplate.* Récupéré sur Sitepoint: <https://www.sitepoint.com/getting-started-with-skeleton-simple-css-boilerplate/>
- MaterializeCSS. (s.d.). *About.* Récupéré sur MaterializeCSS: <https://materializecss.com/about.html>
- Microsoft. (s.d.). *Visio.* Récupéré sur Microsoft: <https://products.office.com/fr-ch/visio/flowchart-software>
- MongoTeam. (2015). *NoSQL.* Récupéré sur Introduction à MongoDB: <https://legacy.gitbook.com/book/mongoteam/introduction-a-mongodb/details>
- Montbeillard, A. D. (2010, 12 15). *Salesforce.com rachète Heroku.* Récupéré sur RelationClientmag.fr: <http://www.relationclientmag.fr/Thematique/techno-ux-1256/Breves/Salesforce-com-rachete-Heroku-181326.htm#gYhEvZ003xCTpcEK.97>
- Nebra, M. (2014). *Des applications ultra-rapides avec Node.js.* OpenClassrooms - ex-Site du Zéro.
- Nebra, M. (2018, mai 07). *Les langages serveur.* Récupéré sur Openclassroom: <https://openclassrooms.com/courses/comprendre-le-web/les-langages-serveur-1>
- Nebra, M. (2018, 03 08). *Node.js : mais à quoi ça sert ?* Récupéré sur OpenClassroom: <https://openclassrooms.com/courses/des-applications-ultra-rapides-avec-node-js/node-js-mais-a-quoi-ca-sert>
- OpenTuto. (2016, 09 14). *Introduction au Framework Foundation.* Récupéré sur OpenTuto: <http://www.opentuto.com/introduction-au-framework-foundation/>

- Porteneuve, C. (2018, 05 14). *Réalisez une application web avec React.js*. Récupéré sur OpenClassroom: <https://openclassrooms.com/courses/realisez-une-application-web-avec-react-js>
- Présentation générale du Plan d'étude romand*. (s.d.). Récupéré sur Plan d'études romand: <https://www.plandetudes.ch/pg2-sommaire>
- Roels, C. (2018, 05 23). *UML, c'est quoi?* Récupéré sur OpenClassrooms: <https://openclassrooms.com/courses/debutez-l-analyse-logicielle-avec-uml/uml-c-est-quoi>
- Sfez, J. (2017, 08 30). *6 raisons de choisir Node.js*. Récupéré sur Smooth code: <https://www.smooth-code.com/articles/6-raisons-de-choisir-node-js>
- Slant. (2018, 06 2). *What are the best databases to use for Node.js applications?* Récupéré sur Slant: <https://www.slant.co/topics/179/~best-databases-to-use-for-node-js-applications>
- Thinkmobile. (2017). *PHP vs NodeJS Comparison and Benchmark*. Récupéré sur thinkmobile: <https://thinkmobiles.com/blog/php-vs-nodejs/>
- Vincent. (2014, 10 18). *Jade vs HTML : Why it matters*. Récupéré sur Stark's blog: <http://vstark.net/2014/10/18/jade-vs-html/>
- Wayner, P. (2017, 02 09). *PHP vs. Node.js: An epic battle for developer mind share*. Récupéré sur InfoWorld: <https://www.infoworld.com/article/3166109/application-development/php-vs-nodejs-an-epic-battle-for-developer-mind-share.html>

## RÉFÉRENCE DES FIGURES

- Image de titre aperçue sur : <https://www.convittonapoli.it/wp-content/uploads/2016/09/Questionario.jpg>
- Figure 1 aperçue sur : <https://castellansystems.com/Waterfall.cshtml>
- Figure 2 aperçue sur : <https://www.seguetech.com/waterfall-vs-agile-methodology/>
- Figure 3 aperçue sur : <https://luis-goncalves.com/what-is-scrum-methodology/>
- Figure 4 aperçue sur : <http://blog.olivierdurand.net/wp-content/uploads/2016/08/userStory.png>
- Figure 5 : Faite par l'auteur sur une machine virtuelle Windows XP
- Figure 6 aperçue sur : <http://evalog.ch/zwook/la-pratique>
- Figure 7 aperçue sur : [https://www.edumoov.com/pages/livret\\_scolaire\\_numerique](https://www.edumoov.com/pages/livret_scolaire_numerique)
- Figure 8 aperçue sur : [https://www.edumoov.com/pages/livret\\_scolaire\\_numerique](https://www.edumoov.com/pages/livret_scolaire_numerique)
- Figure 9 aperçue sur : Faite par le client sur ISM et transmis à l'auteur
- Figure 10 aperçue sur : Faite par le client sur ISM et transmis à l'auteur
- Figure 11 aperçue sur : <https://openclassrooms.com/courses/des-applications-ultra-rapides-avec-node-js/node-js-mais-a-quoi-ca-sert>
- Figure 12 aperçue sur : <https://angular.io/assets/images/logos/angular/angular.svg>
- Figure 13 aperçue sur : <http://slides.com/alexanderfarennikov/react-js-fundamentals/fullscreen#/>
- Figure 14 aperçue sur : <https://getbootstrap.com/docs/4.1/assets/img/bootstrap-stack.png>
- Figure 15 aperçue sur : <http://www.blogduwebdesign.com/integration/9-frameworks-front-end-en-css-sass-et-less/892>
- Figure 16 aperçue sur : <https://nearsoft.com/blog/how-to-center-your-content-with-materialize/>
- Figure 17 aperçue sur : <https://openclassrooms.com/courses/administrez-vos-bases-de-donnees-avec-mysql>
- Figure 18 aperçue sur : <https://www.humancoders.com/formations/mongodb>
- Figure 19 aperçue sur : <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation#L2-2>

---

<b>Figure 20 :</b>	Faite par l'auteur sur Visio
<b>Figure 21 :</b>	Faite par l'auteur sur Visio
<b>Figure 22 :</b>	Faite par l'auteur sur Visio
<b>Figure 23 :</b>	Faite par l'auteur sur Visio
<b>Figure 24 :</b>	Faite par l'auteur sur Visio
<b>Figure 25 :</b>	Faite par l'auteur sur Visio
<b>Figure 26 aperçue sur :</b>	<a href="https://www.developpez.com/public/images/news/webstorm.PNG">https://www.developpez.com/public/images/news/webstorm.PNG</a>
<b>Figure 27 aperçue sur :</b>	<a href="https://assets.gitlab-static.net/uploads/-/system/project/avatar/904528/gitlab-logo-square.png">https://assets.gitlab-static.net/uploads/-/system/project/avatar/904528/gitlab-logo-square.png</a>
<b>Figure 28 aperçue sur :</b>	<a href="http://alfacomsv.com/wp-content/uploads/2017/04/visio.png">http://alfacomsv.com/wp-content/uploads/2017/04/visio.png</a>
<b>Figure 29 aperçue sur :</b>	<a href="https://mockflow.com/images/home-images2/mfMetalImage.jpg">https://mockflow.com/images/home-images2/mfMetalImage.jpg</a>
<b>Figure 30 :</b>	Faite par l'auteur sur mockflow.com
<b>Figure 31 :</b>	Faite par l'auteur sur mockflow.com
<b>Figure 32 :</b>	Faite par l'auteur sur mockflow.com
<b>Figure 33 :</b>	Faite par l'auteur sur mockflow.com
<b>Figure 34 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 35 :</b>	Faite par l'auteur sur MySQL Workbench
<b>Figure 36 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 37 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 38 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 39 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 40 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 41 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 42 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 43 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 44 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 45 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 46 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com

---

<b>Figure 47 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 48 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 49 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 50 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 51 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 52 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 53 :</b>	Faite par l'auteur sur Webstorm
<b>Figure 54 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 55 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 56 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 57 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 58 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 59 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 60 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 61 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 62 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 63 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 64 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 65 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 66 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 67 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 68 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com
<b>Figure 69 :</b>	Faite par l'auteur sur skoolapp.herokuapp.com



ANNEXE I : FEUILLE D'ÉVALUATION ACTUELLE DE L'ÉTAT DU VALAIS



Commune1, EP  
 6Ha, 6H  
 Ginette Modèle

Nom : **Exemple**  
 Prénom : **Michelle**

Année scolaire 2017-2018 **mi-semester 1**

<u>Attitude et comportement</u>	Bien	Assez bien	Suffisant	Insuffisant	<b>Observations</b>
<b>Comportement</b> respecte les règles de vie de l'école, respecte ses camarades et le matériel,...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Michelle respecte les règles de vie de l'école, est attentive et manifeste de l'intérêt. Elle participe aux travaux de groupe. Elle est capable de réinvestir ses connaissances dans un nouveau contexte. Elle accomplit son travail seul et se met au travail spontanément.
<b>Collaboration</b> pose des questions, demande de l'aide, s'implique,...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
<b>Stratégies d'apprentissage</b> intègre les consignes, questionne de manière pertinente, apprend de ses erreurs,...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
<b>Tâches personnelles</b> effectue régulièrement ses devoirs, apprend ses leçons et les réinvestit en classe,...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

<u>Bulletin intermédiaire</u>	<b>Remarques</b>
<b>Français</b>	5,5
<b>Mathématiques</b>	5,5
<b>Autres disciplines</b>	
<b>Suggestion(s) éventuelle(s)</b>	

<b>Entretien avec les parents - remarque(s) - évent. mesure(s)</b>	Date	Signature / Parents	Signature / Titulaire



### ANNEXE III : PRODUCT BACKLOG

Le Product Backlog ci-dessous représente l'état du projet le 22 juillet 2018. Il est sujet à des modifications d'ici à la date butoir du travail de Bachelor.<sup>3</sup>

Product backlog / cahier des charges					
N°	Priorités	User Stories			État
		En tant que...	Je souhaiterais...	Afin de...	
1	1000	Professeur	Evaluer les capacités transversales de mes élèves	voir leur progression au fil de leur scolarité	●
5	1000	Parents	Avoir un aperçu de mon enfant	l'aider au mieux dans ses révisions	●
15	1000	Directeur	Avoir un aperçu uniquement des élèves de mon établissement	connaître la progression des occupants de mon établissement scolaire	●
30	1000	SuperAdmin	Modifier les informations de mes élèves (personnelle + classe + établissement)	garder à jour les informations stockées dans la DB	●
2	999	Professeur	Evaluer les objectifs par branche de mes élèves	voir leur progression au fil de leur scolarité	●
13	980	Professeur	Avoir un aperçu uniquement de ma classe	d'évaluer au mieux les élèves dont je m'occupe	●
20	970	Professeur	Pouvoir évaluer mes élèves avec un graduation à 4 niveaux	Connaître facilement le niveau de mes élèves	●
17	960	Professeur	Voir les objectifs de mes élèves par branche	me concentrer uniquement sur un aspect de leur progression	●
18	950	Professeur	Voir les objectifs de branche séparés par MSN	faciliter mon évaluation	●
14	950	Parents	Avoir un aperçu uniquement de mon enfant	Connaître ses évolutions et sa progression	●
16	950	SuperAdmin	Avoir un aperçu de tous les élèves	d'accéder au paramétrage de chacun	●
27	950	Directeur	Modifier les informations de mes élèves (personnelle + classe)	garder à jour les informations stockées dans la DB	●
19	940	Professeur	Voir les objectifs par MSN filtrés en sous-catégorie	faciliter mon évaluation	●
9	930	Professeur	Pouvoir me connecter à l'interface web	d'accéder à la liste de mes élèves	●
23	920	Professeur	Pouvoir ajouter des remarques à côté des objectifs	de faire une évaluation personnalisée pour chacun de mes élèves	●
3	910	Professeur	Pouvoir ajouter une preuve à mon évaluation, tel qu'une vidéo, bande son ou même pdf	justifier mon évaluation	●

<sup>3</sup> Ce tableau est également disponible sur le CD accompagnant le travail écrit sous le nom de « Product Backlog & séances.xlsx »

4	900	Professeur	avoir un aperçu de l'avancée de mes élèves	mieux les orienter dans leurs études	●
10	900	Parents	Pouvoir me loger à l'interface web	s'accéder à la fiche de mon enfant	●
12	900	SuperAdmin	Pouvoir me loger à l'interface web	d'accéder à la liste de tous les élèves	●
26	900	Directeur	Ajouter des élève dans mon établissement	Gérer les élèves de mon établissement	●
25	890	Professeur	Pouvoir exporter l'état actuel de la progression des élèves par branche ou sous catégorie	sauvegarder leur progression à un instant T	●
6	880	Professeur	Pouvoir imprimer un rapport PDF résumant les objectifs d'un élève/mon enfant	sauvegarder sa progression à un instant T	●
22	870	Professeur	Que les améliorations des objectifs apparaissent une fois que l'élève à acquis le premier niveau	facilité mon évaluation	●
24	860	Professeur	Pouvoir connaître la date de la dernière évaluation	Savoir la ligne temporelle de la progression de l'élève	●
21	850	Professeur	Avoir un aperçu des objectifs passés et futurs de mes élèves	connaître la progression des élèves suivant leurs cycles	●
7	850	Parents	Pouvoir imprimer un rapport PDF résumant les objectifs d'un élève/mon enfant	sauvegarder sa progression à un instant T	●
28	850	Directeur	Sortir un élève de mon établissement	lui permettre d'être ajouter à un autre établissement scolaire	●
29	850	Directeur	Rajouter un élève dans mon établissement	lui attribuer une classe de travail	●
11	800	Directeur	Pouvoir me loger à l'interface web	d'accéder à la liste de tous les élèves de mon établissement	●
8	750	Directeur	Pouvoir imprimer un rapport PDF résumant les objectifs d'un élève/mon enfant	sauvegarder sa progression à un instant T	●

## ANNEXE IV : RÉSUMÉ ITÉRATIONS

Les résumés d'itération ci-dessous représente l'avancée du projet à la date du 22 juillet 2018. La dernière itération (N°9) et la page de résumé total sont sujettes à modifications avant la fin du travail de Bachelor.<sup>4</sup>



Résumé total itérations											
Titre	Skool										
Étudiant	Julie Préperier										
Période	7.05.2018 au 29.07.2018										
	Itération 1	Itération 2	Itération 3	Itération 4	Itération 5	Itération 6	Itération 7	Itération 8	Itération 9	TOTAL	
Recherche / Lecture	15.5	17	4	0	3	1	10	8	0	58.5	360
Analyse	6.5	13.5	0	0	0	0	0	0	0	20	
Séance	2	2	2.5	0	1	1	1	0	0	9.5	
Administration	9	6.5	11	3	3	4.5	2.5	3.5	0	43	
Développement	0	0	19.5	0	14	32.5	32.5	19	0	117.5	
Rédaction	4	25	7	4	6	0.5	1	16	0	63.5	48
<b>TOTAL</b>	<b>37</b>	<b>64</b>	<b>44</b>	<b>7</b>	<b>27</b>	<b>39.5</b>	<b>47</b>	<b>46.5</b>	<b>0</b>	<b>312</b>	

Recherche / Lecture	Recherche + lecture de document concernant le TB
Analyse	Analyse des informations trouvées
Séance	Séance avec le Client ou le répondant
Administration	Préparation de documents / email etc.
Développement	Développement Informatique
Rédaction	Rédaction du document personnel

<sup>4</sup> Ces tableaux sont également disponibles sur le CD accompagnant le travail écrit sous le nom de « Iterations.xlsx »

Résumé itération 1																				
Titre																				
Skool																				
Etudiant																				
Julie Prépeier																				
Objetif																				
Mise en place document admin, recherches																				
Période																				
7.05.2018 au 20.05.2018																				
	Lundi	Mardi	Mercredi	Jeu	Vendredi	Samedi	Dimanche	Lundi	Mardi	Mercredi	Jeu	Vendredi	Samedi	Dimanche	TOTAL					
	07.05.2018	08.05.2018	09.05.2018	10.05.2018	11.05.2018	12.05.2018	13.05.2018	14.05.2018	15.05.2018	16.05.2018	17.05.2018	18.05.2018	19.05.2018	20.05.2018						
Recherche	0	0	0	0	0	4	3	0	0	0	3	0	2	0	3.5					
Analyse	0	0	0	0	0	0	1	0	0	1	0	2	0	0	2.5					
Séance	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
Administration	2	0	0	0	0	4	0	0	0	0	0	1	0	0	9					
Développement	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
Rédaction	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
<b>TOTAL</b>	<b>2</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>8</b>	<b>4</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>8</b>	<b>3</b>	<b>4</b>	<b>0</b>	<b>37</b>					
<b>Tâches pour itération 1</b>																				
Date	Détails							Journal de bord							N°		Description		Statuts	
07.05.2018	Création Todo List + début docs administratifs - Planning - Document Agile - Table des matières - document Word APA - Product backlog														1		Signer feuille "donnée du travail de bachelor"		●	
09.05.2018	X							Travail pour Module 626-1							2		Faire planning des jours de travail (agenda + communiquer au PO)		●	
11.05.2018	X							Travail pour Module 616							3		Definir itération		●	
12.05.2018	Rédaction des documents obligatoires: - Planning par itérations - Table des matières + document officiel APA - Document pour rdv avec client							6 itérations de 2 semaines Planning (mercredi, vendredi pm, samedi, dimanche)							4		Feuille de sprint/itération		●	
13.05.2018	Recherche concernant méthodologie de travail Agile - PB, DSDM etc														5		Journal de bord		●	
16.05.2018	Recherche programme évaluation Recherche langage prog pour Web interface Rédaction avant-propos Recherche concernant le PER, mise en place														6		Préparation document Word Format. APA		●	
17.05.2018	RDV Mathieu Moser Définition des fonctionnalités pour PB Découverte des outils déjà existant														7		Rédaction table des matières brut		●	
18.05.2018	Recherche et analyse des outils déjà existant - Rédaction PB + résumé séance Client - Rédaction table des matières - Recherche outils existant - documents - Installation machine virtuelle Win XP pour Evalog - Recherche langage programmation - PHP, Node.js, Angular - Analyse outils Evalog sur machine virtuelle														8		Acceptation table des matières. <b>Déplacé à l'itération 2</b>		●	
20.05.2018								Voir document Résumé de séance							9		Définition Product backlog avec client (17.05.2018)		●	
															10		Préparation document Product backlog		●	
															11		Type de suivi de projet		●	
															12		Solution existante		●	
															13		Rédaction avant-propos		●	
															14		Mise au propre PB		●	
															15		Installation Machine virtuelle avec outils existant		●	
															16		Recherche langage programmation pour Web interface		●	
															17		Analyse outils existant		●	
														Temps prévu		60				
														Temps restant		23				

Résumé itération 2															
Titre															
Etudiant															
Objectifs															
Période															
	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche	TOTAL
	21.05.2018	22.05.2018	23.05.2018	24.05.2018	25.05.2018	26.05.2018	27.05.2018	28.05.2018	29.05.2018	30.05.2018	31.05.2018	01.06.2018	02.06.2018	03.06.2018	TOTAL
Recherche	2	1	1	0	2	0	0	0	1	1	2	1	2	2	17
Analyse	3.5	1	2	0	0	1.5	0	0	0	1	1	1	1	1.5	13.5
Séance	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2
Administration	0	0	0	1	0	0.5	0	0	1	1	0	0	1.5	0.5	6.5
Développement	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rédaction	0.5	1	2	0	0	3	0	0	3	4	4	1	3.5	3	25
<b>TOTAL</b>	<b>6</b>	<b>3</b>	<b>5</b>	<b>2</b>	<b>2</b>	<b>5</b>	<b>0</b>	<b>1</b>	<b>5</b>	<b>8</b>	<b>7</b>	<b>5</b>	<b>8</b>	<b>7</b>	<b>64</b>
<b>Journal de bord</b>															
<b>Détails</b>															
<b>Commentaires</b>															
<b>Statuts</b>															
Date	N°														
21.05.2018	Recherche documentations pour analyse langages prog. Analyse documentations Rédaction intro chapitre 2.2 Rédaction To-Do list pour itération 2 M3J table des matières Recherche documentation pour Agile Entrer les sources sur doc:Word														
22.05.2018	Rédaction chapitre Agile Analyse document Agile Recherche document Agile														
23.05.2018	Séance avec responsable Recherche sur le sujet Rédaction résumé séance														
24.05.2018	Rédaction du glossaire Rédaction Liste des abréviations Rédaction analyse Evalog Rédaction analyse Edumoov														
25.05.2018	Recherche pour analyse langages prog														
26.05.2018	Installation outils pour UML & BPMN (Visio) Rédaction méthodologie de travail + méthodologie état de l'art Rédaction outils existant (pm.vs.ch)														
28.05.2018	Recherche documentation pour BDD														
29.05.2018	Rédaction intro "Modélisation" Création diagramme de cas d'utilisation Rédaction use case Admin Iteration														
30.05.2018	Recherche diagramme de classe vs domaine model														
31.05.2018	Séance avec responsable Rédaction schéma base de donnée Recherche concernant UML Analyse du besoin des diagrammes UML														
01.06.2018	Admin pour résumé séance + préparation itération 3 Correction état de l'art suite au rlv avec responsable Création BDD														
02.06.2018	Analyse des langages backend Rédaction chapitre Choix Recherche sur les différents langages Analyse des différents langages														
03.06.2018	Rédaction chapitre Analyse langages frontend + BDD Recherche sur les différents langages Analyse des différents langages														
<b>Tâches pour itération 2</b>															
	Description														
	1 Acceptation table des matières (24.05.2018)   Déplacé de l'itération 1														
	2 Acceptation documents admin (24.05.2018)														
	3 Analyse langages programmation														
	4 Analyse outils existant (déplacé au 6.06)														
	5 Use case														
	6 Schéma BDD														
	7 Rédaction 1ère partie "méthodologie de travail"														
	8 Appuyer dans la méthodologie de travail → état de l'art (comment j'ai procédé)														
	9 Ecrire glossaire														
	10 Installation outils pour UML														
	11 Analyse de la BDD														
	12 Correction état de l'art suite au rlv avec responsable														
	13														

Résumé itération 3																
Titre																
Etudiant																
Objectifs																
Période																
	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche	TOTAL	Temps prévu
	04.06.2018	05.06.2018	06.06.2018	07.06.2018	08.06.2018	09.06.2018	10.06.2018	11.06.2018	12.06.2018	13.06.2018	14.06.2018	15.06.2018	16.06.2018	17.06.2018		79
Recherche	2	0	0	0	0	1	0	1	0	0	0	0	0	0	4	
Analyse	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Séance	0	0	1.5	0	0	0	0	0	0	0	0	0	1	0	2.5	
Administration	0	3	0	1	0	2	0	1	0	0	0	0	1	2	11	
Développement	0	1	4.5	0	0	0	0	2	0	0	4	2	1	3	19.5	
Rédaction	0	0	0	1	0	3	0	0	0	0	0	0	0	2	7	
<b>TOTAL</b>	<b>2</b>	<b>4</b>	<b>8</b>	<b>8</b>	<b>6</b>	<b>6</b>	<b>0</b>	<b>4</b>	<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>	<b>3</b>	<b>6</b>	<b>44</b>	<b>35</b>
Journal de bord																
Date	Détails													N°	Description	Statuts
05.06.2018	Début Mockup pour rdv client du 6.6.18 Admin organisation Iteration3 Mise en place environnement développement													1	Création Mockup pour analyse des besoins (voir le 06.06.2018)	●
	RDV Client Mockup													2	Validation Mockup avec Mathieu	●
	mise en place projet Dev mise en place Git													3	Mise en place de l'environnement de développement	●
	Début création DB													4	Création Base de donnée	●
	Admin séance client													5	Mise en place de l'environnement de test	●
	Correction suite à séance													6	Fin analyse outils existants (voir ism vs ch avec Mathieu) <b>Déplacé de l'itération 2</b>	●
	- DB													7	Installation Antidote sur Word	●
	- Use case													8	Création Git	●
	- Tableau comparatif													9	Mise en place environnement test sur Git	●
	- Analyse ISM													10	Modification schéma BD + ajout dans rapport	●
	Création Mockup															
10.06.2018	Admin itération Fin Mockup															
	Création DB (start)															
13.06.2018	Mise en place Git															
	Création DB															
14.06.2018	Mise en place environnement test															
	Mise à jour DB															
15.06.2018	Création DB															
	Séance référent															
	Admin séance référent															
	Installation Antidote															
16.06.2018	Changement schéma DB dans rapport															
	Création relation dans DB															
	Fin création de DB															
17.06.2018	Fin mise en place environnement test															
	Début correction avec Antidote - doc Word															





Résumé itération 4												
Titre												
Skool												
Etudiant												
Julie Préperier												
Objectifs												
Examen Module - Début Interface Prof												
Période												
18.06.2018 au 24.06.2018												
	Lundi	Mardi	Mercredi	Jendredi	Samedi	Dimanche	TOTAL		Temps prévu			
Recherche	0	0	0	0	0	0	0	0	0	0	65	
Analyse	0	0	0	0	0	0	0	0	0	0	0	
Séance	0	0	0	0	0	0	0	0	0	0	0	
Administration	0	0	0	0	0	0	3	3	0	0	0	
Développement	0	0	0	0	0	0	0	0	0	0	0	
Rédaction	0	0	0	0	0	0	4	4	4	4	58	
<b>TOTAL</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>7</b>	<b>7</b>	<b>7</b>	<b>7</b>	<b>0</b>	
Journal de bord												
Date	Détails						Commentaires			N°	Description	Status
18.06.2018	Semaine d'examen de module						Impossibilité de travailler le TB			1	Priorisation du PB	●
22.06.2018	Correction document "itérations" --> Passage à des sprints de 1 semaine --> Placement jours de travail prof. --> Remplissage dernier jour de travail itération 3 Antidote dans doc Word									2	Correction document itérations --> sprint de 1 semaine	●
24.06.2018	Chapitre sur Mockup selon discussion avec référent Mise en place des priorités dans PB - Rôle Prof									3	Correction document Word avec Antidote + envoi au client	●
										4	Rédaction chapitre Mockup	●



Résumé itération 5											
Titre	Skool										
Etudiant	Julie Préperier										
Objectifs	Interface Prof										
Période	25.06.2018 au 01.07.2018										
	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche	TOTAL		Temps prévu	
	25.06.2018	26.06.2018	27.06.2018	28.06.2018	29.06.2018	30.06.2018	01.07.2018	TOTAL		88	
Recherche	0	0	0	0	2	0	1	3			3
Analyse	0	0	0	0	0	0	0	0			0
Séance	0	1	0	0	0	0	0	1			1
Administration	0	1	1	0	0.5	0	0.5	3			3
Développement	0	0	5	0	5.5	0	3.5	14			14
Rédaction	0	6	0	0	0	0	0	6			6
<b>TOTAL</b>	<b>0</b>	<b>8</b>	<b>6</b>	<b>0</b>	<b>8</b>	<b>0</b>	<b>5</b>	<b>27</b>			<b>61</b>
Tâches pour itération 5											
Date	Détails	Commentaires	N°	Description	Statuts						
26.06.2018	Création Interface prof en données brutes Création Accueil superadmin + directeur Création fiche infos élève en read-only Séance avec référent + client	Le programme est, pour le moment construit avec des données brutes qui ne viennent pas de la DB --> dès que l'hébergement est fait, transmettre l'adresse à Mathieu	1 2 3 4	Création démo avec données brutes (interface prof) Création Interface accueil Prof (Déplacé à l'itération 6) Création Interface accueil Directeur (Déplacé à l'itération 6) Création Interface accueil Superadministrateur (Déplacé à l'itération 6)	● ● ● ●						
27.06.2018	Insertion données dans DB	doc .txt avec donnée de base créé	5 6 7	Création Interface Fiche d'info élève (modifiable) (Déplacé à l'itération 6) Création Interface Fiche d'info élève (read-only) (Déplacé à l'itération 6) Insertion données dans DB	● ● ●						
29.06.2018	Changement DB --> Ajout table axe Thematique Finir remplir DB avec valeur fictive Vérification orthographe dans doc Word (faute dans citations) Méthode d'appel de table (Classe, Prof) Visual view ClasseProf	Méthode d'appel pas complete --> prend pas les élèves par classe mais seulement tous	8 9 10	Création document insert data Création page ClasseProf Création page BrancheProf (Déplacé à l'itération 6)	● ● ●						
01.07.2018	Changement DB --> Ajout table classeEleve (jointure) Méthode d'appel des classes avec les élèves visual view classeProf				●						

Résumé itération 6															
Titre															
Skoold															
Étudiant															
Julie Préperier															
Objectifs															
Interface Prof -> branche, recherche avancée élève, fiche élève															
Période															
02.07.2018 au 08.07.2018															
	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche	TOTAL					Temps prévu		
02.07.2018	03.07.2018	04.07.2018	05.07.2018	06.07.2018	07.07.2018	08.07.2018	08.07.2018						91		
Recherche	0	0	0	0	0	0	1						1		
Analyse	0	0	0	0	0	0	0						0		
Séance	1	0	0	0	0	0	0						1		
Administration	1	0	1	0	1	1	0.5						4.5		
Développement	6.5	0	7	0	7	7	5						32.5		
Rédaction	0	0	0.5	0	0	0	0						0.5		
<b>TOTAL</b>	<b>8.5</b>	<b>0</b>	<b>8.5</b>	<b>0</b>	<b>8</b>	<b>8</b>	<b>6.5</b>						<b>39.5</b>		
Temps restant													51.5		
Date	Détails						Journal de bord			Commentaires			N°	Description	Statuts
02.07.2018	Admin itération (fin itération 5, début itération 6) Interface Prof : - view "mises branche" - view "recherche avancée"									Séance avec Roger Shaer pour problème technique Moodle.js			1	Création page BrancheProf (Déplacé à l'itération 5)	●
04.07.2018	Refaire view ClasseProf suite à séance avec Roger Méthodes d'appel pour BrancheProf view BrancheProf Ajout table de jointures classeAnnée pour view BrancheProf Modification DB pour table jointure Modification schéma DB + ajout dans le docx												2	Création Interface accueil Directeur (Déplacé à l'itération 5)	●
06.07.2018	Création view des différents rôles (méthode appel): - Accueil Elèves, Directeur, Superadministrateur - Recherche Directeur, superadministrateur - Classe Directeur, superadministrateur - Annonces Directeur, superadministrateur												3	Création Interface accueil Superadministrateur (Déplacé à l'itération 5)	●
	Interface Login Check des pages obligatoire -> titres -> méthodes appel -> sécurité de l'accès Administration feuille itération Changement schéma DB Correction page accueil superadmin Correction layout + ajout bouton déconnexion+ ajout bouton navigation Changement code couleur Accès des fiches personnelles des élèves depuis interface prof Correction DB pour id_établissement dans table Elève (p-e = a NULL)												4	Création Interface Fiche d'info élève (modifiable) (Déplacé à l'itération 5)	●
07.07.2018	Suite interfaces obligatoires avec méthode d'appel Correction affichage bouton retour Remplir DB pour évaluation élève -> 1ère classe Directeur ne voit que les élèves de son/ses établissements Méthode post des annonces (directeur + superadmin) Correction accès fiche élève-ro												5	Création Interface accueil Prof (Déplacé à l'itération 5)	●
08.07.2018													6	Création Interface view classeProf Fin de création view classeProf	●
													7	Méthodes d'appel pour toutes les pages (déplacé à l'itération 7)	●
													8	Méthodes modification fiche élève (readOnly) (déplacé à l'itération 7)	●
													9	Correction DB	●
													10		●

Résumé itération 7							
Titre	Skool						
Étudiant	Julie Préprier						
Objectifs	Fin création méthode d'appel, début création méthode modification						
Période	09.07.2018 au 15.07.2018						
	Lundi	Mardi	Mercredi	Jeu	Vendredi	Samedi	Dimanche
	09.07.2018	10.07.2018	11.07.2018	12.07.2018	13.07.2018	14.07.2018	15.07.2018
Recherche	2	0	2	2	0	2	10
Analyse	0	0	0	0	0	0	0
Séance	0	0	0	0	1	0	1
Administration	1	0	0	0	0.5	1	2.5
Développement	6	0	6	5	4.5	4	32.5
Rédaction	0	0	0	1	0	0	1
<b>TOTAL</b>	<b>9</b>	<b>0</b>	<b>8</b>	<b>8</b>	<b>6</b>	<b>7</b>	<b>47</b>
							1 Temps restant
							34.5
							81.5

Date	Détails	Journal de bord	Commentaires	N°	Description	Statuts
09.07.2018	Page recherche avancée pour Prof / Dir / superAdmin Afficher modal eleve_accueil Méthode update infos_eleve + localité + login Méthode delete etablissement Méthode ajout classe Affichage admin directeur --> gestion classe + gestion etablissement eleve Méthode delete classe Visual edit classe (modal)			1	Méthodes d'appel pour toutes les pages (déplacé de l'itération 6)	●
11.08.2018	Affichage des évaluations des élèves sur les fiches perso -> accès prof -> accès read-only Debug affichage evaluation eleve dans branche prof mise en ligne -> déploiement		Les évaluations des élèves ne s'affiche toujours pas sur la page branche_prof Déploiement pas fini -> erreur	2	Méthodes modification fiche élève (readOnly) (déplacé de l'itération 6)	●
12.07.2018	Séance avec le PO: -> possible solution avec l'interface Prof Création page admin du superadministrateur -> création d'objet -> modification / suppression objets Création de la page admin pour superadmin -> création d'objet (etablissement, localite, eleve, prof, directeur) -> affichage des modal pour la modification des objets Finition page admin superadmin + directeur		Document écrit corrigé par Mathieu	3	Méthode d'administration pour Directeur --> gestion classe	●
13.07.2018				4	Méthode d'administration pour Directeur --> gestion établissement pour élève	●
14.07.2018				5	Page de recherche avancée + méthode de recherche pour Prof	●
15.07.2018				6	Page de recherche avancée + méthode de recherche pour Directeur	●
				7	Page de recherche avancée + méthode de recherche pour Superadmin	●
				8	Page admin superadministrateur --> création/suppression/édition/établissement	●
				9	Page admin superadministrateur --> création/suppression/édition directeur	●
				10	Page admin superadministrateur --> création/suppression/édition prof	●
				11	Page admin superadministrateur --> création/suppression/édition élèves	●
				12	Page admin superadministrateur --> création/suppression/édition localité	●
				13	Evaluation d'élève depuis leur fiche élève	●
				14	Evaluation d'une classe depuis interface branche (prof) (déplacé vers l'itération 8)	●
				15	Déploiement de l'application	●
				16	Page admin directeur --> création/suppression/édition d'objectif	●
				17	Page admin directeur --> création/suppression/édition de groupe	●
				18	Page admin directeur --> création/suppression/édition classe	●
				19	Page admin directeur --> création/suppression/édition branche	●
				20	Page admin directeur --> création/suppression/édition groupe	●
				21	Page admin directeur --> création/suppression/édition objectif	●
				22	Page admin directeur --> création/suppression/édition classe	●
				23	Page admin directeur --> création/suppression/édition classe	●
				24	Page admin directeur --> création/suppression/édition classe	●
				25	Page admin directeur --> création/suppression/édition classe	●
				26	Ajout date de naissance pour élève	●
				27	Création des données de base au lancement de l'application	●

Résumé Itération 8										
Titre	Skool									
Étudiant	Julie Prérier									
Objectifs	Fin du développement									
Période	16.07.2018 au 22.07.2018									
	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche	TOTAL	Temps prévu	
Recherche	2	3	2	1	0	0	0	8	8	64.5
Analyse	0	0	0	0	0	0	0	0	0	
Séance	0	0	0	0	0	0	0	0	0	
Administration	1	0	0	0	1	0.5	1	3.5	3	
Développement	5	0	4	3	0	3	4	19	16	
Rédaction	0	0	2	0	7	4.5	2.5	16	18	
<b>TOTAL</b>	<b>8</b>	<b>3</b>	<b>8</b>	<b>4</b>	<b>8</b>	<b>8</b>	<b>7.5</b>	<b>46.5</b>	<b>18</b>	

Tâches pour l'itération 8			
Date	Détails	Commentaires	Statuts
16.07.2018	Ajout de la date de naissance d'un élève Check du user experience: - bouton save au même endroit pour toutes les modals - bouton annuler dans toutes les modals - orthographe Déploiement de la nouvelle version interface pour ajouter un élève dans une interface Analyse de la base de donnée afin de créer toutes les interfaces d'administration du superadmin et directeur Fin interfaces administrative superadmin + directeur		●
17.07.2018	Déploiement nouvelle version Rédaction intro, résumé, avant propos Corrections manque interface administrative -> création pour les tables de jointures Déploiement nouvelle version Rédaction fin technical guide Rédaction début user guide Ajouter remarque dans évaluation Ecrire chapitre sur les améliorations futures Ajouter les références des figures Ecrire introduction Afficher la date de la dernière evaluation Changer le schéma de DB		●
18.07.2018	Check du document écrit dans Word Ecrire conclusion Ajouter Annexe Check de l'orthographe avec Antidote		●
19.07.2018			●
20.07.2018			●
21.07.2018			●
22.07.2018			●



ANNEXE V : RÉSUMÉ SÉANCE CLIENT / RÉFÉRENT

<b>Date</b>	17.05.2018
<b>Lieu</b>	Bureau centre ICTVS, Sion

Points discutés	
N°	Details
1	Les outils utilisé maintenant ism.vs.ch => remplir pdf interactif et export sur Excel 1x par année scolaire et par élève
2	Décision de faire les capacités transversales + 1 branche spécifique = Maths, 5ème degré (2e cycle)
3	Volonté de séparer par MSN
4	Les capacités transversales sont sur le site du PER et les objectifs de math dans doc que Mathieu à envoyé
5	Voir ce qui se fait dans les autre cantons
6	MITIC fait par canton de Vaud et Fribourg sont sur papier (pour les 3 cycles)
7	Les objectifs évolus. Ex: 5ème pour les math = 1 objectif et pour la 6ème = meme objectif mais amélioré (!\ toujours le même cycle)
8	Décision qu'une réévaluation supprimera la 6ème
9	Definition des fonctionnalité pour le PB
10	Amélioration pour avenir : intégration système absence
11	Amélioration pour avenir : Integration à ism.vs.ch

<b>Date</b>	24.05.2018
<b>Lieu</b>	Technopôle, Sierre

Points discutés	
N°	Details
1	Décrire le contexte du projet dans l'introduction
2	Faire une analyse aussi d'un point de vue des fonctionnalité (pour les bibliothèques d'image etc.)
3	Expliquer les critère de productivité. (Pas autant productive avec du C# qu'avec du Node. Parce que je repars from scratch
4	Dans etat de l'art avec l'analyse du langage --> faire également analyse pour BDD
5	Objectif itération 2 : finir état de l'art + mise en place environnement de dev
6	la 5ème et 6ème itérations pourront etre coupée en sprint d'une seule semaine
7	Definition des priorités des rôles: Prof (1), Super admin pour la gestion (2), et parent avant meme directeur
8	Ajouter dans méthodologie comment j'ai choisi mes thèmes de l'état de l'art, comment j'ai travailler pour l'état de l'art
9	Les mots en anglais dans le doc on les met entre guillemets si ce n'est pas habituel
10	Ecrire un glossaire et dire dans l'introduction que j'ai utilisé des mots classique en

<b>Date</b>	01.06.2018
<b>Lieu</b>	Bellevue, Sierre

<b>Points discutés</b>	
<b>N°</b>	<b>Details</b>
1	Correction Etat de l'art + conseil d'amélioration
2	Pas besoin de trop écrire (pas se répéter) Si ils comprennent pourquoi j'ai fait les choix, c'est suffisant
3	Création d'un tableau comparatif des fonctionnalités entre les différent outils
4	Chapitre "choix" devrait aller après "Modélisation" car il y a encore un lien entre la modélisation et l'état de l'art
5	Possibilité de faire un accès rapide de la liste d'élève aux objectifs (par nom ou par résultat d'évaluations)
6	Aide pour le schéma de base de donnée (associations)
7	Pour l'état de l'art de la BDD, montrer schéma maintenant + schéma final
8	Conseil de création de mockup pour la compréhension et l'analyse de tous les besoins du client --> utilisés pour le rdv avec client (6.6.18)
9	Rajouter au chapitre choix mes outils de visualisation +environnement de test
10	Avoir des discussions pour le choix du framework



Date	06.06.2018
Lieu	Bureau centre ICTVS, Sion

Points discutés	
N°	Details
1	Demonstration + envoi de capture d'écran d'ism.vs.ch Il est possible de tout gérer sur cette plateforme tel que la liste d'élève, les absences, les évaluations, les remplacements etc.
2	A partir de la 4H, les notations se font de manière numérique (1 à 6) En 3H, les notations sont faite en 4 niveaux (pas atteint, part. atteint, atteint, atteint avec aisance)
3	Il est possible de mettre un nombre d'évaluation illimité --> fait la moyenne à la fin
4	<b>Pour Skool --&gt; pas de notation numérique du tout + une seule évaluation fait foi</b>
5	Correction Use Case: Parent + Prof --> afficher + modifier les infos personnelles d'un élève + mettre un champ "remarque" Directeur --> doit pouvoir gérer les classes (prof etc)
6	Correction tableau comparatif: - Ism != objectifs PER - Ism = ok pour aperçu tous les élève - ISM = ok pour aperçu 1 seul élève - ISM = ok pour admin - ISM --> pas de remarque sur les évaluations mais sur élève en général
7	Se baser sur cn.bibnumerique.com pour le visuel
8	Création de mockup avec le client
9	Correction de la DB: Suppression du lien entre la table Classe et la table Annee Création du lien entre la table Eleve et la table Annee 1 prof peut etre dans plusieurs établissements

<b>Date</b>	15.06.2018
<b>Lieu</b>	Bellevue, Sierre

<b>Points discutés</b>	
<b>N°</b>	<b>Details</b>
1	En ce qui concerne les mockups, expliquer dans le chapitre Développement que j'en ai fait, et montrer les principaux.
2	Penser à faire une option de réinitialisation de la DB depuis le visuel
3	<p>Discussion sur l'oral:            Rapport écrit sera lu par 2 personnes            Normalement pas d'auditeur externe sauf sur demande            3 personnes à l'oral            ~30-35 min de passage</p> <p>Il faut partir de l'hypothèse que l'expert a lu le rapport --&gt; reprendre des choses du rapport mais apporter des éléments supplémentaires            Ex: Schéma sur le processus avec les mockups et client            Reprendre les étapes peu développées dans le rapport mais les approfondir à l'oral            Suivre de toute manière le fil rouge du rapport écrit            Avoir une démo (peut-être commencer avec)            Next step, Qu'est ce qui aurait été fait différemment, ce qui a été fait en plus</p>
5	Validation correction Page 5
6	Installer Antidote sur le document Word
7	Ne pas montrer l'évolution de la construction de la DB mais montrer la dernière version et préciser qu'elle à été améliorée

<b>Date</b>	26.06.2018
<b>Lieu</b>	Bellevue, Sierre

<b>Points discutés</b>	
<b>N°</b>	<b>Details</b>
1	Check du changement de délai pour itération
2	Check de la table des matière pour la partie dev
3	Check du chapitre mockup
4	<p>Démo --&gt; amélioration proposées</p> <ul style="list-style-type: none"> <li>- ajouter d'un filtre pour la/les classe/s (checkbox) avant l'acordéon de branche</li> <li>- Dans la page branche, au niveau des objectifs, avoir une liste des étudiants des classes sélectionnées plus haut et pouvoir mettre des évaluations + bouton de sauvegarde par objectif</li> <li>- Fiche infos élèves --&gt; bouton de sauvegarde général au fond de la page + message avertissement si pas sauvegardé</li> <li>- page accueil prof --&gt; faire une preview des annonces (soit sommet, soit coté)               <ul style="list-style-type: none"> <li>- les pseudo des utilisateurs indique leur rôle (ens.xxx, ad.xxx etc)</li> </ul> </li> <li>- Fiche info élève (read only) --&gt; pas d'accordéon pour les objectifs mais mettre un bouton "information" avec modal qui montre plus de détails sur l'évaluation</li> <li>- Code couleur : style bleu Word mais essayer de respecter les couleurs officielle pour les branches (voir capture écran Mathieu)</li> </ul>

<b>Date</b>	13.07.2018
<b>Lieu</b>	Bellevue, Sierre

<b>Points discutés</b>	
<b>N°</b>	<b>Details</b>
1	<p>Ajouter un bouton annulé sur la modal de modification de l'élève</p> <ul style="list-style-type: none"> <li>-&gt; accès prof</li> <li>-&gt; accueil élève</li> <li>-&gt; accès read-only</li> </ul>
2	<p>Changement des priorités:</p> <p>-&gt; faire toutes les autres interfaces avant de reprendre l'interface prof</p>
3	Si problème -> prendre rdv avec Roger Schaer
4	Si vraiment pas possible de créer les pages, montrer ce qui a été fait meme si c'est supprimé
5	<p>Priorité -&gt; pour le dimanche 15 = finir toutes les autres interface à 100% + déploiement.</p> <p>Donner le lien du site en ligne au client et référent</p>
6	Nouvelle possibilité : pour fiche élève et branche prof, au lieu de faire des modals et des collapsibles en plus, ouvrir une nouvelle page.

## 7. DÉCLARATION DE L'AUTEUR

« Je déclare, par ce document, que j'ai effectué le travail de Bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de Bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après : Mathieu Moser ».