

GestAuto, un site de gestion d'autoécoles

Gestion d'autoécoles

TRAVAIL DE BACHELOR

ALEXANDRE MONTERROSO

Mai 2016

Supervisé par :

Prof. Dr. Jacques PASQUIER–ROCHA
Software Engineering Group

« *Il n'y a pas de projet urgent, il n'y a que des projets en retard.* »

Remerciements

Tous mes remerciements vont aux personnes ayant contribué à la réalisation de ce travail. Je pense tout particulièrement au Professeur Jacques Pasquier-Rocha, qui par sa patience, son temps et sa disponibilité m'ont permis de beaucoup apprendre de ce travail. Ses conseils m'ont apporté les clefs pour fournir le meilleur de moi-même dans ce travail, tout en me permettant d'apprendre la documentation d'un tel travail.

Je remercie également l'entreprise Wicki-autoécole, qui par sa disponibilité, ses explications et sa mise à disposition des informations dont j'avais besoin, a grandement contribué à la réalisation du présent travail.

Résumé

Dans une entreprise moderne, un système d'informations apparaît comme une nécessité incontournable. Ainsi, ce travail décrit comment une autoécole qui gère actuellement toute sa logistique, sa facturation et la gestion de l'entreprise manuellement et sur papier, peut améliorer son efficacité en utilisant un système d'information regroupant ses besoins actuels.

GestAuto se propose de résoudre les problèmes courants de l'entreprise afin de lui faire économiser un temps considérable et par la même occasion, gagner en efficacité. Le travail commence en décrivant l'entreprise, puis en étudiant les besoins d'une telle entreprise. Pour terminer, un prototype de programme a été réalisé, GestAuto, qui permet à l'entreprise de réduire sa charge opérationnelle.

Pour cela, une base de données cohérente et complète (réalisation d'un modèle entité-association, puis en le transformant en tables) est décrite. Les technologies PHP, CSS et jQuery sont étudiées, puis utilisées.

Sommaire

1	Introduction	11
1.1	Motivations et objectifs	11
1.2	Organisation du présent travail.....	11
2	Présentation de l'entreprise et sa problématique.....	12
2.1	Etat actuel	12
2.1.1	L'autoécole.....	12
2.1.2	Permis de conduire	13
2.1.3	Les cours	17
2.1.4	Etat actuel de la gestion de l'entreprise.....	19
2.2	Proposition de logiciel répondant à la problématique	20
2.2.1	Base de données	20
2.2.2	Description des fonctionnalités du site.....	20
3	Présentation du prototype.....	27
3.1	Authentification	27
3.2	Fonctions du moniteur	28
3.2.1	Inscription d'un élève par l'élève lui-même	28
3.2.2	Gestion des élèves par le moniteur.....	30
3.2.3	Types de cours.....	33
3.2.4	Calendrier	34
3.2.5	Valider un cours	36
3.2.6	Liste des cours	37
3.3	Fonctions de le secrétaire	37
3.3.1	Gestion des moniteurs	37
3.3.2	Gestion des secrétaires	38
3.4	Fonctions de l'élève.....	38

3.4.1	Calendrier	38
3.4.2	Liste des cours	38
4	Présentation de l'implémentation.....	40
4.1	Modèle entité-association	40
4.2	Transformation des entités en tables	43
4.2.1	Utilisateurs	44
4.2.2	Cours	45
4.3	Contraintes.....	48
4.4	Description du Framework Symfony 2	55
4.4.1	Framework	55
4.4.2	Modèle-vue-contrôleur.....	56
4.4.3	Configuration	57
4.4.4	Le routage.....	57



Liste des Figures

Figure 1 : Diagramme d'activités « permis B »	Erreur ! Signet non défini.
Figure 3 : Diagramme d'activités « permis A »	Erreur ! Signet non défini.
Figure 4 : Diagramme d'activités "permis A1"	Erreur ! Signet non défini.
Figure 5 : Diagramme d'activités « permis C »	Erreur ! Signet non défini.
Figure 6 : Diagramme d'activités « permis C1 »	Erreur ! Signet non défini.
Figure 7 : Diagramme d'activités « permis D »	Erreur ! Signet non défini.
Figure 8 : Diagramme d'activités « permis D1 »	Erreur ! Signet non défini.
Figure 9 : Diagramme d'activités « permis E »	Erreur ! Signet non défini.
Figure 10 : Diagramme d'activités « permis F »	Erreur ! Signet non défini.
Figure 11 : Diagramme d'activités « permis G »	Erreur ! Signet non défini.
Figure 12 : Diagramme d'activités « permis M »	Erreur ! Signet non défini.
Figure 13 : Diagramme d'activités « permis bateau cat. A »	Erreur ! Signet non défini.

Liste des Tableaux

Tableau 1 : cas d'utilisation « s'authentifier »	65
Tableau 2 : cas d'utilisation « inscrire un élève »	66
Tableau 3 : Cas d'utilisation « modifier un élève »	67
Tableau 4 : Cas d'utilisation « supprimer un élève »	68
Tableau 5 : Cas d'utilisation « désactiver un élève »	69
Tableau 6 : Cas d'utilisation « inscrire un de ses élèves à un cours collectif »	70
Tableau 7 : Cas d'utilisation « supprimer un de ses élèves à un cours collectif »	71
Tableau 8 : Cas d'utilisation « voir détail cours collectif »	72
Tableau 9 : Cas d'utilisation « voir liste cours individuels »	73
Tableau 10 : Cas d'utilisation « créer un type de cours »	74
Tableau 11 : Cas d'utilisation « modifier un type de cours »	75
Tableau 12 : Cas d'utilisation « supprimer un type de cours »	76
Tableau 13 : Cas d'utilisation « Ajouter un événement au calendrier »	77
Tableau 14 : Cas d'utilisation « Modifier un événement au calendrier »	78
Tableau 15 : Cas d'utilisation « Supprimer un événement au calendrier »	79
Tableau 16 : Cas d'utilisation « Voir le calendrier complet »	80
Tableau 17 : Cas d'utilisation « Valider un cours »	81
Tableau 18 : Cas d'utilisation « S'inscrire »	82
Tableau 19 : Cas d'utilisation « Créer un mot de passe et activer son compte »	83
Tableau 20 : Cas d'utilisation « Modifier ses informations personnelles »	84
Tableau 21 : Cas d'utilisation « S'inscrire à un cours collectif »	85
Tableau 22 : Cas d'utilisation « Modifier son inscription à un cours collectif »	86
Tableau 23 : Cas d'utilisation « Voir les cours validés »	87
Tableau 24 : Cas d'utilisation « Voir le calendrier »	87
Tableau 25 : Cas d'utilisation « Créer un cours collectif »	89
Tableau 26 : Cas d'utilisation « Modifier un cours collectif »	90
Tableau 27 : Cas d'utilisation « Supprimer un cours collectif »	91
Tableau 28 : Cas d'utilisation « Inscrire un moniteur »	92
Tableau 29 : Cas d'utilisation « modifier un moniteur »	93

Tableau 30 : Cas d'utilisation « supprimer un moniteur »	94
Tableau 31 : Cas d'utilisation « inscrire un secrétaire »	96
Tableau 32 : Cas d'utilisation « modifier un secrétaire »	97
Tableau 33 : Cas d'utilisation « supprimer un secrétaire »	98

Liste des Codes Source

Code Source 1 : Test unitaire pour le formulaire « création de secrétaire »	59
Code Source 2 : création d'une liste avec les données du formulaire.....	60
Code Source 3 : création du formulaire	60
Code Source 4 : Instanciation de l'objet Secrétaire	60
Code Source 5 : Soumission du formulaire.....	60
Code Source 6 : Test des données du formulaire	60
Code Source 7 : Création de la vue du formulaire	61

1

Introduction

1.1 Motivations et objectifs

Dans le cadre de ma formation à l'université de Fribourg, pour l'obtention du « Bachelor in Computer Sciences », je suis amené à produire un « Travail de Bachelor » qui met en pratique les cours dispensés pendant la formation.

Lors de mes propres cours d'autoécole, je me suis demandé comment était gérée l'autoécole puisque l'organisation semblait légèrement chaotique. Cela m'a amené à examiner leurs systèmes de près et à réfléchir comment l'adapter grâce à des technologies modernes, tels qu'un système informatisé d'information, avec l'utilisation d'une tablette dans la voiture et d'un téléphone portable récent pour les déplacements.

De la gestion du projet à la réalisation finale du projet lui-même, j'ai également appris à maîtriser une technologie et des programmes qui m'étaient inconnus, tels un Framework PHP (Symfony2), un autre Framework CSS (Bootstrap), sans oublier un Framework JavaScript (jQuery). Ces compétences me seront utiles pour le début de ma carrière professionnelle.

1.2 Organisation du présent travail

Par neutralité, tous les genres du présent travail seront au masculin.

Nous commençons par décrire l'entreprise sélectionnée. Des entretiens avec cette entreprise m'ont permis de mieux cerner le problème et d'identifier exactement quels sont les besoins et attentes de l'entreprise. Dans ce même chapitre, nous décrivons de manière détaillée les fonctionnalités qui doivent être implémentées pour répondre aux attentes.

Dans un deuxième temps, il m'a fallu réaliser et implémenter l'application décrite au chapitre précédent. Suite à quoi, nous détaillons dans ce deuxième chapitre chaque fonctionnalité implémentée avec des captures d'écran, pour que le lecteur puisse visualiser de manière précise l'application.

Pour terminer, l'implémentation qui fut faite de l'application est décrite. Pour cela, le modèle des données nécessaires à une telle application est présenté et les pistes de son développement futur selon les besoins formulés par l'entreprise sont expliquées. Pour plus de clarté, et afin de bien pouvoir identifier tous les problèmes et répondre à toutes les attentes, un modèle entité-association est dessiné et expliqué. Ce modèle a ensuite été transformé en modèle de base de données relationnelle.

2

Présentation de l'entreprise et sa problématique

Afin d'étudier les besoins de l'entreprise, l'entreprise en question et son fonctionnement sont brièvement expliquées : quelques notions de l'entreprise, ses activités, et comment elle s'est organisée pour arriver à ses fins. Après cela, à l'aide de quelques diagrammes UML¹, sont présentées les principales fonctionnalités que devrait fournir le nouveau système afin de répondre aux besoins cités.

2.1 Etat actuel

Dans un premier temps, nous allons explorer l'état actuel des choses pour une autoécole. Qu'est-ce qu'une autoécole ? Quels genres d'activités y sont exécutés ? Quelle est la gestion actuelle des processus de l'entreprise pour exécuter ces activités ? Voilà autant de questions qui seront adressées dans ce chapitre.

2.1.1 L'autoécole

Institutions privées, les autoécoles sont en général de petites entreprises indépendantes qui forment les nouveaux conducteurs aux règles de circulation et à avoir un comportement routier approprié. Les participants sont formés pour utiliser différents véhicules sur la voie publique. Des cours spéciaux sont disponibles afin de les faire interagir avec les différents usagers de la voirie et de les sensibiliser aux dangers de la route.

Pour ce faire, la formation est subdivisée en un certain nombre de catégories de permis de conduire. Si une personne veut obtenir un permis, le moniteur va la guider dans le processus d'apprentissage.

Nous allons détailler ici les différentes formations qui sont proposées par l'entreprise que nous étudions. Il s'agit de tous les permis proposés par l'Office de la circulation et de la navigation (OCN). Chaque permis est énoncé par une lettre associé ou non à un chiffre : soit A, A1, B, C, C1, D, D1, E, F, G, M et le permis bateau à moteurs (cat. A).

¹ <http://www.uml.org>, consulté en décembre 2015

2.1.2 Permis de conduire

Un permis de conduire est une licence autorisant la conduite d'un type de véhicule dans la voie publique. Ce genre d'autorisation s'obtient après la réussite d'une formation dans une autoécole.

Les procédures qui sont requises pour débiter une formation de permis de conduire sont présentées, ainsi que les étapes nécessaires pour leurs obtentions. Un diagramme d'activités présente ces étapes : celui du permis B, tandis que tous les autres sont en annexes, afin que le lecteur intéressé puisse s'y référer.

Les préliminaires

Avant de se former pour obtenir un permis de conduire, chaque candidat doit effectuer deux actions préliminaires détaillées ci-après.

- Examen de la vue

Pour tous les permis, un examen de la vue est nécessaire. Cet examen peut être passé chez un opticien reconnu par l'OCN selon les normes en vigueur correspondant à la catégorie du permis. Dans le cas où le candidat a déjà obtenu ce certificat, il reste valable 24 mois au maximum ; dans le cas contraire il doit être repassé. Si l'examen est négatif, l'élève doit consulter un médecin. Celui-ci peut encore fournir une autorisation avec un certain nombre de conditions qui devront être appliquées à chaque utilisation du véhicule. Si celui-ci décide qu'il ne peut l'autoriser, l'élève n'a pas le droit de passer son permis de conduire.

- Premiers secours

Le cours de premiers secours est un cours obligatoire de 10 heures qui présente l'essentiel des premières actions à entreprendre en cas de blessures, que ce soit dans un accident ou ailleurs. Cela entraîne à réagir de manière adéquate et efficace, tout en apprenant à identifier les situations à risque ainsi que les mesures de sécurité correspondantes.

Catégories enseignées

Les véhicules qui ont le droit de circuler sur la voie publique, sont catégorisés en différentes sortes. Ainsi chaque catégorie a un permis de conduire qui lui est propre, et la formation lui est propre : tous deux sont détaillés pour la situation en 2016. Puisque la loi évolue rapidement, une autoécole doit être capable de s'adapter aux nouvelles mesures et évoluer.

- Permis A

La catégorie A est le permis moto qui est divisée en trois catégories :

- A1 (illustré dans la Figure 49)

Le permis A1 est le permis moto d'une cylindrée de 125 cc² maximum et d'une puissance maximum de 11 Kw³. Si le permis est passé à 16 ans, il est limité à 50 cc et à 18 ans la limitation augmente automatiquement à 125 cc.

² Centimètres cubes

³ Kilowatts

Si l'élève a obtenu son permis B (voiture), avant de commencer celui-ci, il peut directement aller présenter les dossiers nécessaires à l'OCN, qui lui délivrera alors un permis provisoire, lui permettant de conduire seul une moto pour son apprentissage pendant quatre mois. Au cours de cette période, l'élève doit faire huit heures de cours d'autoécole obligatoires et obtient son permis définitif.

En revanche, si l'élève n'a pas obtenu son permis B, l'élève doit, avant de présenter les dossiers à l'OCN, faire un cours de premiers secours. L'OCN lui délivre ensuite un permis provisoire valable quatre mois. Une fois le permis provisoire obtenu, il doit en plus des huit heures de conduite, suivre un cours de sensibilisation à la conduite routière, suite à quoi il reçoit une prolongation de permis provisoire de 12 mois supplémentaires. Après quoi il doit passer un examen pratique.

- A25 (illustré dans la Figure 48)

Si l'élève a moins de 25 ans, il obtient un permis limité à 25 Kw. Le processus pour obtenir le permis est le même que pour la catégorie A1, sauf qu'il doit suivre 12 heures de cours de conduite obligatoires. Pour passer l'examen l'élève a besoin d'une moto entre 11 Kw et 25 Kw. Après deux ans, il peut passer à l'OCN pour transformer son permis en permis A+25.

- A+25 (illustré dans la Figure 48)

Si l'élève a plus de 25 ans, il obtient un permis non limité. Le processus est similaire à la catégorie A25, sauf qu'il doit passer l'examen avec une moto de plus de 35 Kw.

▪ Permis B (illustré dans la Figure 1)

Le permis B est le permis voiture. Ce permis est obligatoire pour obtenir le permis C et D. Pour prétendre à ce permis, le candidat doit être âgé d'au moins 18 ans.

Comme illustré dans la Figure 1, après avoir passé l'examen de la vue et un cours de premiers secours d'une durée de 10 heures, l'élève doit remplir les formulaires et dossiers et les soumettre à l'OCN ; l'élève reçoit alors un permis provisoire jaune, qui lui permet de se présenter à l'examen théorique. Une fois réussi, un nouveau permis provisoire, blanc cette fois, est délivré. A ce stade, l'élève doit suivre un cours de sensibilisation de huit heures tout en participant à des cours d'apprentissage pratique, soit avec un moniteur d'autoécole, soit avec une personne ayant obtenu son permis depuis plus de 2 ans et ayant plus de 23 ans. Aucune heure d'autoécole obligatoire n'est exigée pour passer l'examen pratique. Une fois réussi, l'élève obtient un permis valable pendant 3 ans, en format carte de crédit. Ce permis est à l'essai et le nouveau conducteur ne doit pas commettre de fautes graves pendant les trois années suivantes du permis à l'essai. De plus, il doit effectuer durant cette période un cours de perfectionnement en deux phases (L2 à Romont) qui dure deux jours. Après ces trois ans, l'élève reçoit finalement son permis définitif.

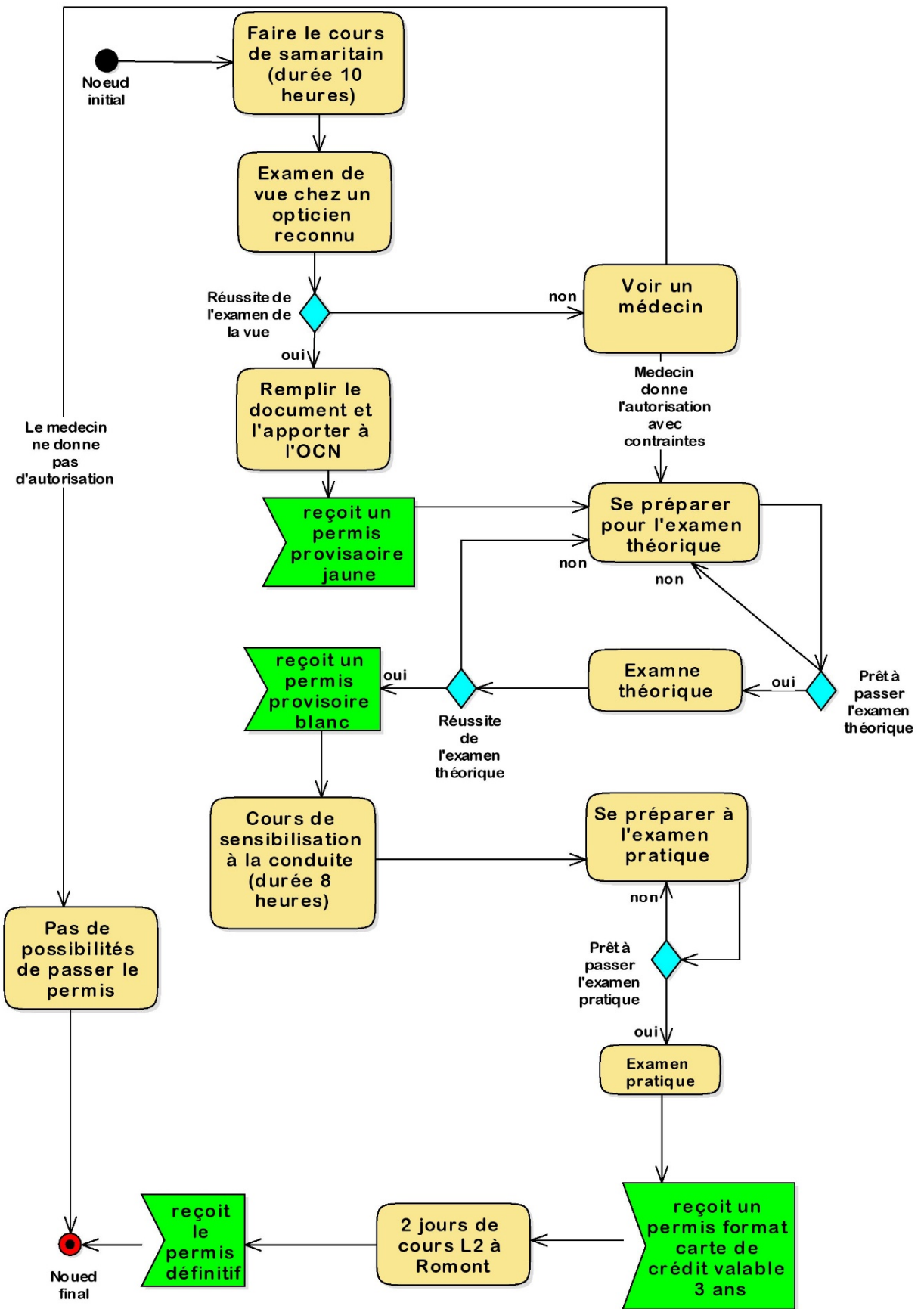


Figure 1 : Diagramme d'activités « permis B »

- Le permis C (illustré dans la Figure 50)

Le permis C est le permis camion. Il permet de conduire les véhicules de plus de 7500 KG. Pour pouvoir passer ce permis, il est nécessaire d'avoir déjà réussi le permis voiture (B).

Comme illustré sur la Figure 50, après avoir réussi l'examen de la vue, il faut encore passer un examen médical. Après avoir apporté les papiers nécessaires à l'OCN, l'élève doit se préparer pour l'examen théorique, suivi de l'examen pratique. A ce moment-là, l'élève reçoit un permis définitif, mais il ne peut pas conduire sans un certificat de capacités pour marchandises qu'il obtient après 5 jours de formation en autoécole et la réussite de l'examen. Après la formation de base, cinq jours de perfectionnement doivent être effectués tous les cinq ans pour garder le certificat.

- Le permis C1 (illustré dans la Figure 51)

Le permis C1 est un permis permettant de conduire un véhicule de maximum 7500 KG pour le déplacement de marchandises.

Comme illustré sur la Figure 51, le processus est le même que pour le permis camion, sauf que la théorie est adaptée, ainsi que le véhicule d'examen pratique.

- Le permis D (illustré dans la Figure 52)

Le permis D est le permis de car qui est défini comme un véhicule pouvant transporter plus de 16 personnes.

Comme illustré dans la Figure 52, il s'agit de la même procédure que pour le permis camion, sauf qu'il faut faire un certain nombre d'heures de conduite obligatoires. Ainsi si l'élève a déjà obtenu le permis camion et qu'il peut certifier (par une entreprise par exemple) la conduite d'un camion pendant plus de 500 heures, aucune heure d'autoécole n'est obligatoire. Si les 500 heures ne peuvent être certifiées, 24h de conduite en autoécole seront obligatoires. Si l'élève ne possède que le permis voiture, 52h de conduite en autoécole sont obligatoires.

- Le permis D1 (illustré dans la Figure 53)

Le permis D1 est un permis de minibus, défini comme un véhicule pouvant transporter un maximum de 16 personnes.

Comme illustré sur la Figure 53, le processus est le même que pour le permis camion, sauf que la théorie est adaptée, ainsi que le véhicule d'examen pratique.

- Le permis E (illustré dans la Figure 54)

Le permis E est un permis remorque. Un permis B permet de tracter une petite remorque pour autant qu'elle ne dépasse pas 750 KG ou que le poids combiné du véhicule tracteur et de la remorque ne dépasse pas 3500 KG. Si ce poids est dépassé, un permis remorque est nécessaire.

Comme illustré sur la Figure 54, après avoir passé l'examen de la vue et avoir apporté les papiers à l'OCN, un permis provisoire permet de conduire sans accompagnement pendant 24 mois. Avant la fin du délai, un examen pratique doit être obtenu.

- Le permis F (illustré dans la Figure 55)

Le permis F est un permis tracteur. Il permet de conduire un tracteur jusqu'à 45 km/h. Il faut avoir 16 ans pour être autorisé à le conduire.

Comme illustré dans la Figure 55, après avoir effectué l'examen de la vue et apporté les papiers à l'OCN, l'admission à l'examen est acceptée. Après avoir réussi l'examen théorique, l'élève reçoit un permis provisoire pour 12 mois. Après la réussite à l'examen pratique, le permis définitif lui est attribué.

- Le permis G (illustré dans la Figure 56)

Le permis G est un permis tracteur. Il permet de conduire un tracteur jusqu'à 30 km/h. Il faut avoir 14 ans pour être autorisé à le conduire.

Comme illustré dans la Figure 56, après avoir effectué l'examen de la vue et apporter les papiers à l'OCN, l'examen théorique peut être passé. A la réussite de cet examen, le permis définitif lui est accordé. Si les autorités doutent de l'aptitude de l'élève à conduire, elles peuvent demander qu'il fasse un examen pratique.

- Le permis M (illustré dans la Figure 57)

Le permis M est un permis vélomoteur. Il permet de conduire un vélomoteur. Il faut avoir 14 ans pour en faire la demande.

Comme illustré dans la Figure 57, après avoir effectué l'examen de la vue et apporté les papiers à l'OCN, l'élève peut passer l'examen théorique. A la réussite de cet examen, le permis définitif lui est accordé. Si les autorités doutent de l'aptitude de l'élève à conduire, elles peuvent demander qu'il fasse un examen pratique.

- Le permis bateau à moteur (cat. A) (illustré dans la Figure 58)

Le permis bateau à moteur permet de conduire un bateau à moteur de plus de 8 chevaux sur les lacs suisse et frontaliers.

Comme illustré dans la Figure 58, après avoir réussi l'examen de la vue et apporter la documentation à l'OCN, l'élève peut passer l'examen théorique. Par la suite, il peut passer l'examen pratique ; s'il est réussi, on lui délivre un permis définitif. Pour les cours d'apprentissage, il n'y a nul besoin de passer l'examen théorique d'abord ; cependant, une personne ayant le permis doit être aux côtés de l'élève.

2.1.3 Les cours

Comme on a pu le remarquer dans la section précédente, différents cours doivent être proposés afin de répondre aux besoins des candidats des différents permis de conduire. Pour y répondre, il existe plusieurs formes et sortes de cours selon la vitesse d'apprentissage désirée, l'attention que l'élève a besoin, les contraintes légales et les infrastructures disponibles.

- Les cours de sensibilisations

Ces cours collectifs sont obligatoires. Ils sensibilisent les élèves à la conduite sur la voie publique en quatre séances de deux heures.

- Cours 1 : Les partenaires
- Cours 2 : Les différentes routes

- Cours 3 : La mécanique
- Cours 4 : Alcool, drogues, médicaments
- Les cours de premiers secours

Ces cours collectifs sont obligatoires. Ils servent à sensibiliser les gens aux soins de premier secours à fournir aux blessés en cas d'accident. « Comment agir vite et bien », « sécuriser les personnes », « dans quel ordre agir » sont autant de thèmes qui sont abordés durant ce cours.

- Les cours de théorie

Les cours de théorie ont lieu pour enseigner aux élèves les connaissances nécessaires pour la réussite de l'examen théorique des différents permis de conduire. En général, ces cours sont collectifs et présentés par un seul moniteur sur place.

- Les cours de théorie privés

Des cours privés sont également organisés pour la théorie lorsque l'élève en fait la demande car il veut aller plus vite ou qu'il a besoin d'aide personnalisée. Un moniteur est sur place avec un ou plusieurs élèves selon la demande qui a été faite.

- Les cours OACP (Ordonnance sur l'admission à la circulation des chauffeurs professionnels)

Lors de la réussite du permis C, C1, D ou D1, afin de transporter des marchandises ou des personnes, un certificat de capacités est nécessaire.

Si le permis de conduire a été obtenu avant 2003, la personne obtient directement le certificat. Mais elle doit effectuer cinq jours de formation continue tous les cinq ans afin de valider son certificat.

Si le permis de conduire a été obtenu après 2003, la personne doit faire une formation de 12 jours et prouver les compétences acquises lors d'un examen. Pour pouvoir se présenter à l'examen final elle doit passer un examen sous forme QCM (questionnaire à choix multiples) à l'OCN. L'examen final a lieu au Colombier pour le transport des marchandises et à Renens pour le transport des personnes et s'effectue en trois parties orales et une partie pratique.

Ces cours enseignent aux élèves la conduite sûre et respectueuse de l'environnement, la manipulation correcte du véhicule et à avoir un bon comportement sur la route.

- Les cours pour entreprises

Certaines entreprises tiennent à offrir une formation continue à leurs conducteurs professionnels. De manière régulière, ces entreprises envoient leur chauffeurs suivre des cours dans une école de conduite pour s'assurer qu'ils ont conservé de bons réflexes et qu'ils apprennent les nouvelles règles.

- Les cours pratiques

Ces cours ont lieu en général seul avec le moniteur dans un véhicule. Celui-ci enseigne de manière pratique la conduite correcte du véhicule dans la pratique et au sein de la circulation routière. Ils ont lieu avec le véhicule correspondant à la catégorie du permis de conduire. Des pédales doubles permettent au moniteur de réagir et de corriger l'élève quand cela est nécessaire.

2.1.4 Etat actuel de la gestion de l'entreprise

Actuellement, plusieurs entreprises indépendantes organisent chacune leurs cours individuels. Pour les cours collectifs, elles créent des synergies en réunissant leurs élèves pour un cours en commun. Les moniteurs peuvent choisir les cours auxquels ils souhaitent participer et par conséquent y envoyer des élèves.

Ce système est actuellement géré à la main et sur papier, comme expliqué dans les paragraphes suivants.

Gestion des élèves :

Les élèves sont inscrits sur les formulaires papiers et stockés dans des classeurs, ce qui rend la recherche et le stockage peu aisés. D'autre part, il est difficile de gérer les élèves lorsqu'en déplacement, les dossiers étant au bureau. Il faut demander au client de patienter et le rappeler par la suite ; cela engendre une perte de temps, un coût supplémentaire et augmente le risque que l'élève aille voir ailleurs.

Gestion des cours :

L'inscription aux cours individuels se fait sur un agenda papier. Il arrive que des erreurs s'insinuent dans le calendrier : plusieurs élèves semblent alors avoir réservé le même cours. Il est parfois aussi difficile d'identifier un emplacement libre. Toute modification dans l'agenda est compliquée et devient rapidement illisible. L'agenda doit être constamment transporté d'un endroit à l'autre, ce qui engendre un énorme risque en cas de perte. Cet agenda risque également d'être oublié. Si un appel pour des informations ou pour une inscription à un cours a lieu lorsque le moniteur n'a pas l'agenda avec lui, celui-ci doit rappeler l'élève une fois l'agenda a portée de main, ce qui engendre une perte de temps. Il arrive régulièrement que les élèves oublient les horaires de leurs rendez-vous, ce qui engendre une nouvelle perte de temps pour chercher dans l'agenda les créneaux horaires à transmettre à l'élève. L'agenda est vite rempli et certaines informations ne peuvent pas y être contenues.

L'inscription aux cours collectifs a lieu dans la salle. Le secrétaire dresse un tableau et l'affiche dans la salle. Les moniteurs qui veulent inscrire leurs élèves se rendent dans la salle pour y inscrire le nom de chaque élève. Cela engendre des problèmes : d'une part une perte de temps, d'une autre part, les moniteurs ne connaissent pas d'avance le nombre de places disponibles et ils doivent donc reconfirmer aux élèves. Parfois ils ont oublié de confirmer et l'élève ne se présente alors pas ; au contraire, parfois il se présente et il y a trop de personnes dans la salle. De plus, l'élève ne paie que sur place les cours collectifs. Cela veut dire que s'il ne se présente pas, il a pris la place de quelqu'un qui avait envie de venir, et qui aurait donc payé, ce qui crée un manque à gagner pour l'entreprise.

Gestion des paiements :

Le moniteur doit se souvenir des élèves qui ont payé et ceux qui ne l'ont pas fait.

2.2 Proposition de logiciel répondant à la problématique

Afin de répondre à la problématique expliquée dans le chapitre précédent, voici une proposition d'application qui pourrait régler les problèmes cités précédemment. Afin de garantir une meilleure compréhension, les fonctionnalités sont d'abord représentées graphiquement grâce à des diagrammes UML et des cas d'utilisation, puis expliquées sommairement. Des tableaux détaillés sont en annexes et permettent au lecteur intéressé de voir plus de détails sur la fonctionnalité en question.

2.2.1 Base de données

Une base de données centralisée permet de répondre à la majeure partie des problèmes liés à la gestion des élèves. Les appareils mobiles, tablette ou smartphone, permettent d'y accéder depuis n'importe où. Il n'est donc plus nécessaire de faire de longues recherches pour trouver un élève et ses coordonnées. D'autre part, l'élève peut de lui-même changer ses données personnelles, sans devoir passer par le moniteur, ce qui fait gagner un temps considérable au moniteur.

2.2.2 Description des fonctionnalités du site

Afin d'expliquer le système, nous allons présenter les fonctionnalités en plusieurs parties. Nous commencerons par le cas général où l'utilisateur arrive sur le système, mais n'est pas encore authentifié. Le système aura trois sortes d'utilisateurs différents : le moniteur, le secrétaire et l'élève. Chacun a des fonctionnalités sur le système qui lui sont propres. Nous commencerons par les fonctionnalités du moniteur, suivies par celles de l'élève, pour terminer par celles de le secrétaire.

Afin de faciliter la compréhension d'un tel système, nous allons utiliser des diagrammes de cas d'utilisation. Chaque diagramme contiendra des cas d'utilisation (ovales bleus) qui résument en quelques mots ce que cette fonctionnalité permettra de faire. Ensuite chacun de ces ovales, qu'on appellera cas d'utilisation, sera décrit dans un petit tableau, avec quelques informations concernant ce qu'il fait, et les divers cas qui pourraient se présenter. Il sera présent dans les annexes et la référence sera notée dans chaque explication en italique, afin de permettre au lecteur intéressé de la retrouver facilement.

Authentification

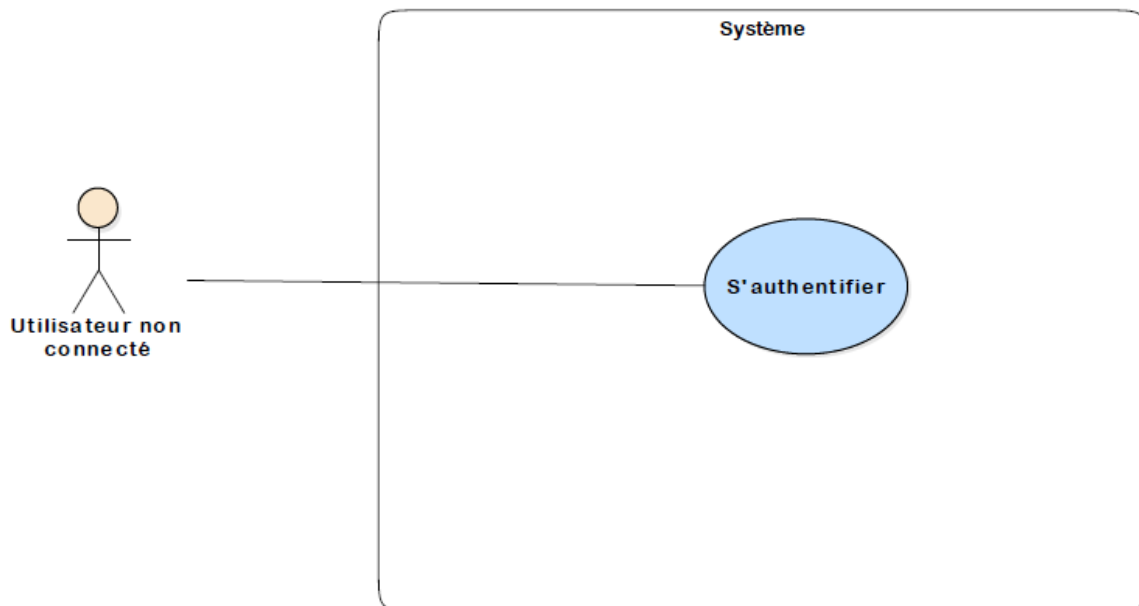


Figure 2 : cas d'utilisation « s'authentifier »

Dans le premier cas d'utilisation, expliqué dans le *Tableau 1*, le nouvel utilisateur arrivant sur le site doit s'identifier. Pour ce faire, il doit entrer un nom d'utilisateur et un mot de passe. Le système vérifie les informations et si elles correspondent à un utilisateur inscrit dans la base de données, l'utilisateur sera identifié sur le système.

Le moniteur

Tout d'abord les fonctionnalités du moniteur sont envisagées. Comme les cas d'utilisation de cet acteur sont trop nombreux pour apparaître lisiblement sur un diagramme, celui-ci sera divisé en deux diagrammes différents, chacun expliquant une partie du logiciel. Ainsi une partie sera consacrée aux élèves et à leur gestion et une autre à la gestion du calendrier.

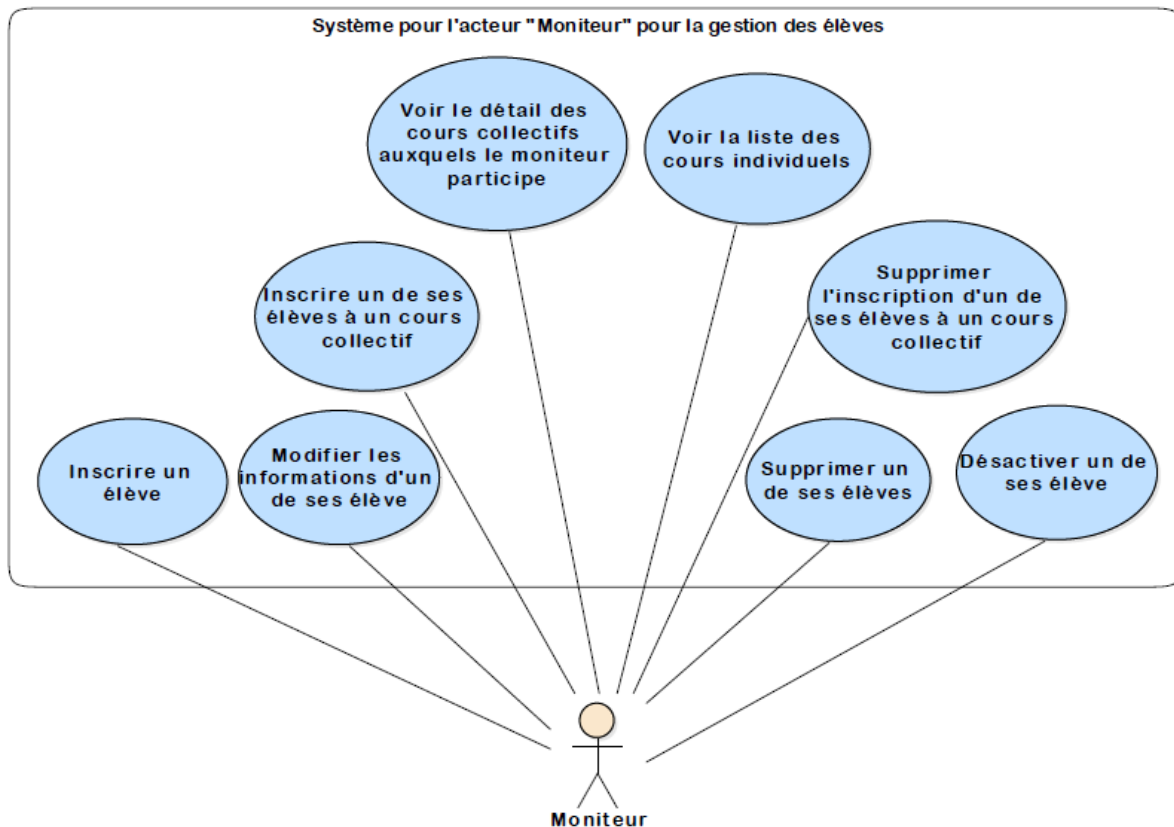


Figure 3 : cas d'utilisation pour l'acteur « moniteur », gestion des élèves

■ Gestion des élèves

Comme indiqué sur la Figure 3, un élève peut s'inscrire seul sur le système à partir du site du moniteur. Néanmoins il sera parfois nécessaire que le moniteur inscrive lui-même l'élève (*Tableau 2*) dans le système : dans ce cas le moniteur n'aura besoin que du nom, prénom et adresse email de l'élève. Après quoi, l'élève recevra un email avec un lien d'activation. En cliquant sur ce lien, l'élève devra entrer ses informations personnelles et créer un mot de passe. Cela permet au moniteur de gagner du temps au moment de l'inscription et d'éviter la saisie d'erreurs dans la transmission orale.

Une fois l'élève inscrit et activé, le moniteur pourra également modifier les informations personnelles de celui-ci (*Tableau 3*). Ce cas peut se présenter par exemple si l'élève change d'adresse ou de numéro de téléphone et le donne directement au moniteur. D'autre part, si l'élève n'a pas activé son compte et a perdu son lien d'activation, le moniteur doit pouvoir lui renvoyer.

Une fois que l'élève a fini les cours d'autoécole, la possibilité existe pour le moniteur de supprimer le dossier de l'élève, bien que cela ne soit pas recommandé (*Tableau 4*). En effet, il est plus avisé de ne pas supprimer un élève, mais simplement de le désactiver (*Tableau 5*). Ainsi les informations de l'élève resteront sur le système. Le moniteur pourra dans le futur rechercher ses anciens élèves, voir les cours qu'il a réalisés ou payés, etc. En cas de suppression ces informations seront perdues définitivement.

Nous avons classé les cours en deux catégories distinctes : les cours individuels, où le moniteur inscrit un élève (*Tableau 9*) à son cours, situation détaillée dans la section suivante, et les cours collectifs, où les élèves de plusieurs moniteurs distincts peuvent s'inscrire à ce cours.

Les élèves peuvent s'inscrire seuls au cours collectif, bien que le moniteur garde la possibilité de le faire aussi (*Tableau 6*). Chaque moniteur peut voir (*Tableau 8*) les cours collectifs que le secrétaire a créés et le moniteur y est inscrit comme participant du cours. Le moniteur peut en cliquant sur le cours, voir la liste de ses élèves. Il pourra ainsi inscrire un de ses élèves à ce cours ou le désinscrire (*Tableau 7*) de ce cours.

- Gestion du calendrier

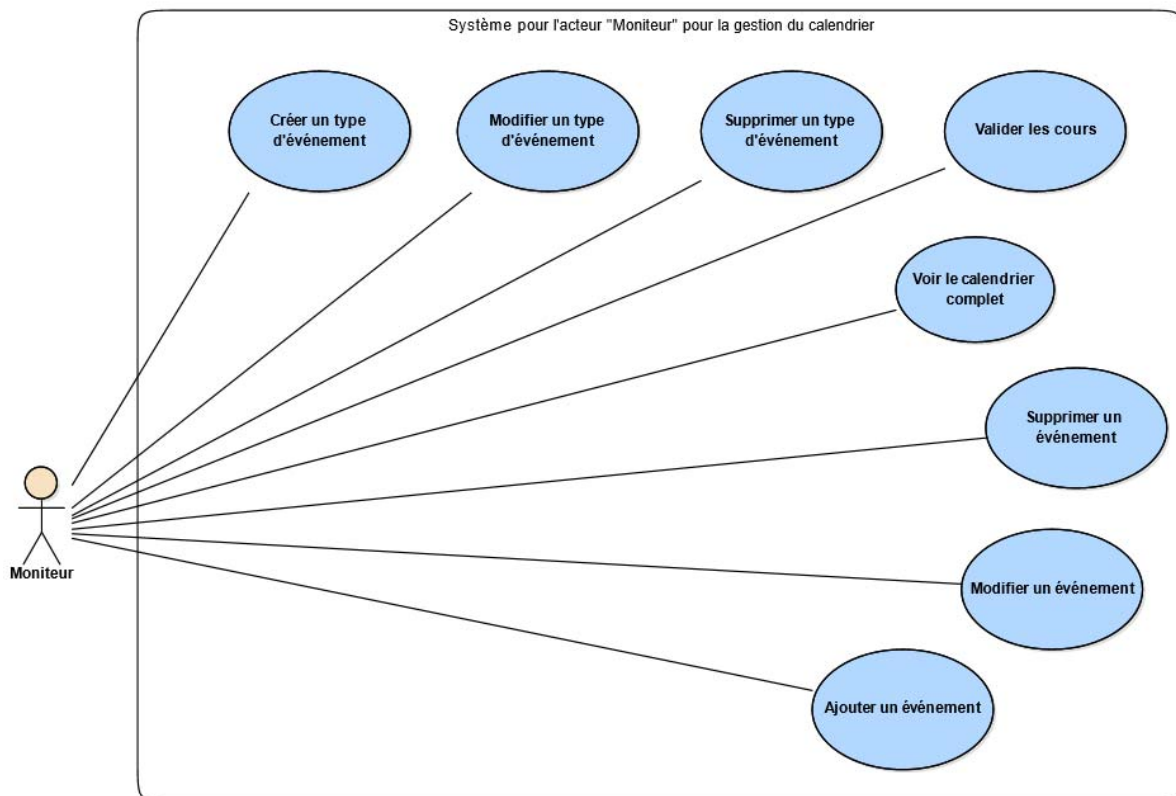


Figure 4 : Cas d'utilisation pour l'acteur « moniteur », gestion du calendrier

Parmi les cours individuels, nous pouvons les classer en plusieurs types de cours différents. Ainsi on crée un modèle de cours. Par exemple nous pourrions créer un cours « Pratique voiture », qui contiendrait un prix de base (qui pourra être modifié individuellement pour chaque cours par la suite), une description du cours (pour l'afficher sur le calendrier par exemple) et un champ définissant si le cours devra être facturé ou non. Le moniteur devra donc, avant de réserver un cours individuel, créer un type de cours (*Tableau 10*) y référant. Il lui est aussi possible de modifier le type (*Tableau 11*) ou de le supprimer (*Tableau 12*).

Pour pouvoir créer un cours individuel, le moniteur doit ajouter un événement au calendrier (*Tableau 13*) et choisir obligatoirement un élève et un type de cours, parmi une liste proposée. D'autre part, il devra déterminer l'horaire du cours, le lieu du rendez-vous avec l'élève et il

pourra y insérer des remarques. Un événement du calendrier pourra être modifié (Tableau 14) ou supprimé (Tableau 15).

Le moniteur doit pouvoir également voir le calendrier en entier (Tableau 16), avec les réservations de tous les élèves qui lui sont associés.

Après qu'un cours ait eu lieu, le moniteur devra le valider (Tableau 17) et mentionner si l'élève a payé directement et/ou s'il y a eu un changement dans le prix du cours.

L'élève

L'élève peut se connecter lui-même sur le système et peut effectuer un certain nombre d'actions. Celles-ci sont décrites dans cette section.

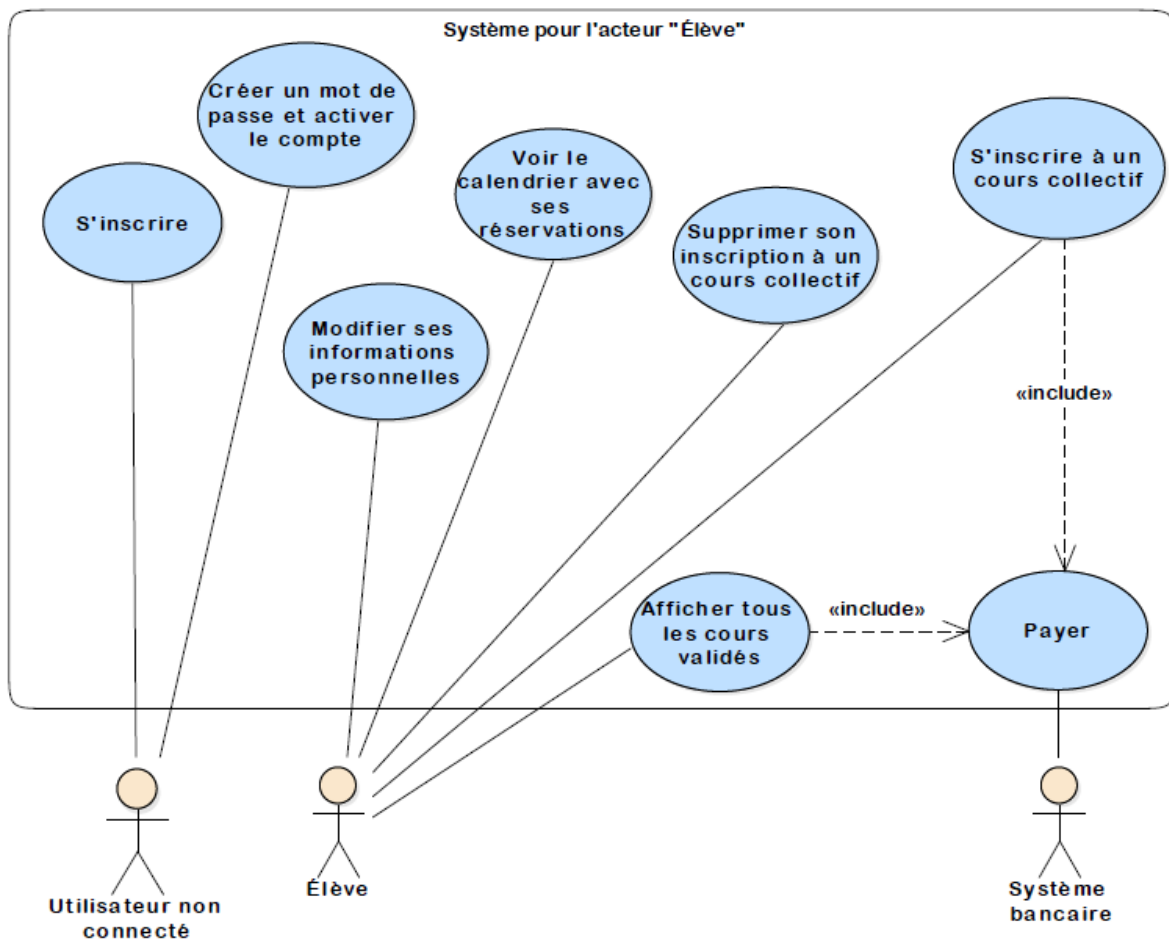


Figure 5 : Cas d'utilisation pour l'acteur « élève »

Un élève peut s'inscrire directement à partir du lien que lui a donné le moniteur (Tableau 18). Il doit entrer pour cela toutes ses données personnelles et créer un mot de passe. Suite à quoi, un email est envoyé avec un lien d'activation ; en cliquant sur celui-ci, l'élève active son compte.

Le moniteur peut également créer un compte pour l'élève : dans ce cas l'élève reçoit un email avec un lien d'activation, qui va lui permettre d'entrer ses données personnelles et créer un mot de passe (Tableau 19). Par la même occasion son compte sera activé.

L'élève peut changer ses informations personnelles (Tableau 20) si ses données ont changé, en cas de déménagement ou de nouveau numéro de téléphone.

Une fois que le secrétaire a annoncé la participation d'un moniteur à un cours collectifs, les élèves de ce dernier peuvent s'inscrire à ce cours (Tableau 21). Pour s'y inscrire, l'élève devra payer directement les frais du cours.

L'élève peut également supprimer son inscription à un cours collectif (Tableau 22). Pour cela des frais engendrés sont déduits directement et le solde du cours est ajouté au solde de son compte, qui lui permettra de commander à nouveau un autre cours. Si le délai de désinscription est passé, les frais augmentent et la même procédure est suivie.

L'élève peut également voir les cours que le moniteur a validés (Tableau 23). Les cours qui n'ont pas été annoncés comme payés, peuvent être directement soldés en cliquant sur une icône « payer ».

L'élève peut également consulter le calendrier (Tableau 24) avec tous les cours qu'il a réservés et ceux qu'il a déjà réalisés. Cela permet un gain de temps considérable, puisque le moniteur et le secrétaire n'ont pas besoin de répondre incessamment aux élèves qui ont oublié leurs dates de rendez-vous.

Le secrétaire

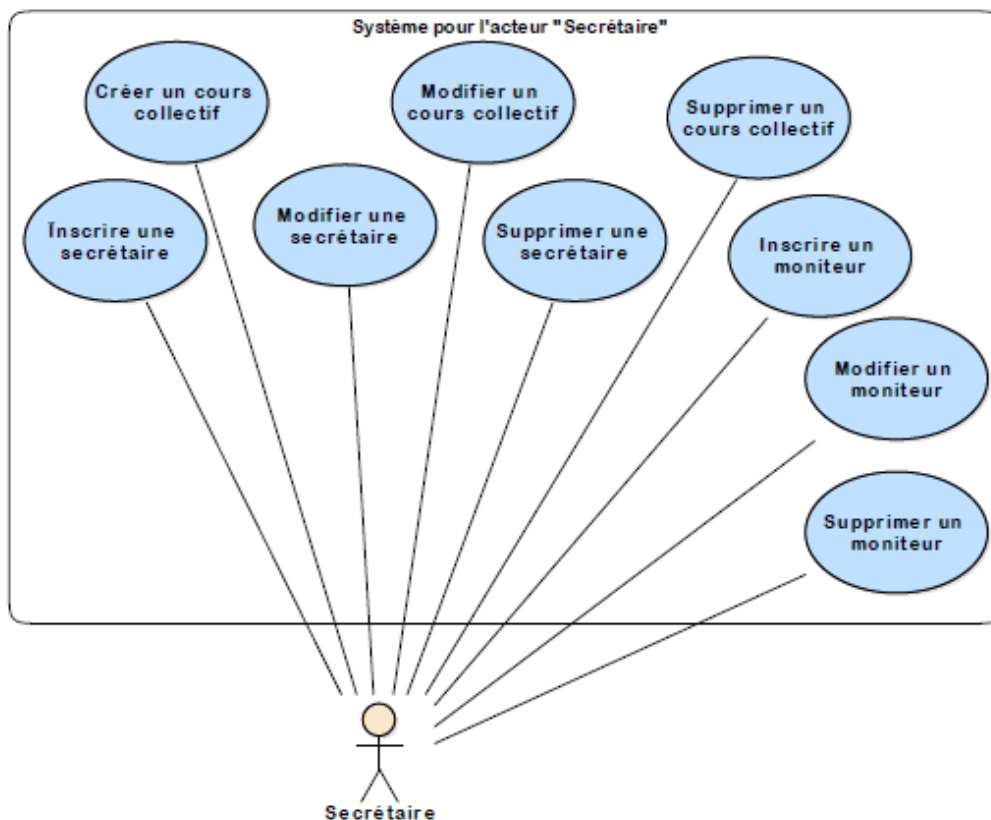


Figure 6 : Cas d'utilisation pour l'acteur « secrétaire »

Le secrétaire, quant à elle, peut créer (Tableau 25), modifier (Tableau 26) ou supprimer (Tableau 27) un cours collectif et inscrire de nouveaux moniteurs ou secrétaires.

Le secrétaire peut également inscrire (Tableau 28), modifier (Tableau 29) ou supprimer (Tableau 30) un moniteur.

Dans le même principe, le secrétaire peut également inscrire (Tableau 31), modifier (Tableau 32) ou supprimer (Tableau 33) un secrétaire.

3

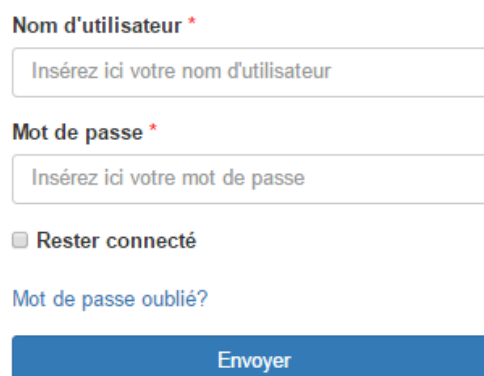
Présentation du prototype

Dans ce deuxième chapitre, nous allons présenter comment les besoins ont été transformés en une application fonctionnelle. Chaque fonctionnalité de l'application y est expliquée sommairement et il sera donné la référence du tableau du cas d'utilisation qui s'y réfère. Pour une meilleure compréhension le contenu est partagé en plusieurs parties. Ainsi, l'authentification sera d'abord expliquée, suivie des fonctions qui peuvent être réalisées par le moniteur, celles de l'élève et pour terminer celles de le secrétaire.

3.1 Authentification

Afin de répondre au cas d'utilisation expliqué dans le Tableau 1, lorsqu'un visiteur arrive sur le site, la seule option qui s'offre à lui est de s'identifier. Pour cela, comme on peut le voir sur la *Figure 7*, il doit entrer un nom d'utilisateur et un mot de passe qui vont lui permettre de s'identifier sur le site et d'accéder aux fonctions selon ses droits.

On peut également voir un lien « Mot de passe oublié ». Celui-ci permet, si un utilisateur a oublié son mot de passe de le récupérer. Il suffit à l'utilisateur en question d'entrer son adresse email, pour recevoir par email, un lien lui permettant de créer un nouveau mot de passe. A partir de ce moment, l'utilisateur ne pourra se connecter que par ce nouveau mot de passe.



Nom d'utilisateur *

Mot de passe *

Rester connecté

[Mot de passe oublié?](#)

Figure 7 : Capture d'écran « identification »

3.2 Fonctions du moniteur

Dans ce chapitre, il sera question d'étudier les fonctionnalités qui s'offrent au moniteur. Dans un premier temps, seront abordées les fonctions qu'a le moniteur pour gérer ses élèves, puis dans un deuxième temps, celles qui concernent la gestion des cours. Lorsque le moniteur s'identifie sur le système, il arrive sur sa page d'accueil (*Figure 9*), dans laquelle il obtient le lien qu'il doit fournir à ses élèves pour qu'ils s'inscrivent.

Un élève peut être inscrit sur le site de deux manières différentes. Soit l'élève peut s'y inscrire lui-même via le lien donné par le moniteur, soit le moniteur l'inscrit lui-même manuellement.

Afin de faciliter la navigation, un menu en haut de chaque page permet de rejoindre toutes les autres pages.



Figure 8 : Capture d'écran « menu »

3.2.1 Inscription d'un élève par l'élève lui-même

Chaque moniteur dispose d'un lien qu'il doit donner à ses élèves pour s'inscrire. Les moniteurs y ont accès en se connectant à leur espace privé (*Figure 7*). Le lien est affiché dans un champ de texte. Ce lien peut être copié en cliquant sur un bouton directement sur le champ, ou alors en faisant un simple copier-coller. Les élèves peuvent ensuite s'inscrire directement en utilisant ce lien.



Figure 9 : Capture d'écran « page d'accueil moniteur »

Lorsque l'élève clique sur le lien fourni par le moniteur, celui-ci obtient un formulaire (Figure 10).

The image shows a registration form with the following fields and callouts:

- Prénom ***: Insérez ici votre prénom
- Nom ***: Insérez ici votre nom
- Téléphone**: Insérez ici votre téléphone
- Téléphone portable**: Insérez ici votre téléphone portable
- Nationalité**: Insérez ici votre nationalité
- Adresse ***: Insérez ici votre adresse
- Numéro ***: Insérez ici votre numéro de rue
- Code Postal ***: Insérez ici votre code postal
- Ville ***: Insérez ici votre ville (circled in red)
- Pays ***: Insérez ici votre pays
- Email ***: Insérez ici votre adresse email
- Nom d'utilisateur ***: Insérez ici votre nom d'utilisateur
- Mot de passe ***: Insérez ici votre mot de passe
- Répéter le mot de passe ***: Répétez le mot de passe

Buttons: Envoyer (blue), Annuler (red)

Callout 1: Formulaire que l'élève doit remplir afin d'activer son compte

Callout 2: Les étoiles rouges indiquent que le champ est obligatoire

Figure 10 : Capture d'écran « formulaire d'inscription élève »

Le formulaire contient un certain nombre de champs, qui correspondent aux données que l'utilisateur doit entrer. Certains champs sont obligatoires et caractérisés par une petite étoile rouge, d'autres sont facultatifs.

Une gestion des erreurs permet d'avertir l'utilisateur que ce qu'il essaie de faire n'est pas possible.

Les erreurs sont affichées en rouge, avec un message d'erreur au dessous expliquant la raison de l'erreur (Figure 11) ; tant que l'erreur n'est pas corrigée, l'utilisateur ne peut pas soumettre le formulaire.

Email *

alexandre@monterroso.ch

! Cette adresse email existe déjà, merci d'en choisir une autre

Figure 11 : Capture d'écran « erreur du formulaire »

3.2.2 Gestion des élèves par le moniteur

Dans son espace membre, le moniteur a un espace où il peut gérer ses élèves :

The screenshot shows a user interface for managing students. It includes a table with columns for Nom, Prénom, Portable, Adresse email, Activé le, and Créé le. Callouts highlight specific features: a green '+ Ajouter' button, a dropdown menu for 'Voir les membres qui sont:', a search bar for 'Rechercher' with 'Nom' selected, and action icons (delete, edit, activate) for each student row.

Le moniteur clique pour ajouter un nouvel élève

Possibilité de trier les élèves

Champ de recherche permettant de trier les élèves par nom, prénom ou code postal

	Nom	Prénom	Portable	Adresse email	Activé le	Créé le
	NomEleve1	PrénomEleve1	1111111111	eleve1@eleve	Activé	01/04/2016
	NomEleve2	PrénomEleve2	1111111111	eleve2@eleve	Activé	05/04/2016

Supprimer l'élève

Modifier les informations de l'élève

Activer/Désactiver l'élève

Figure 12 : Capture d'écran « gestion des élèves »

Depuis cet espace (Figure 12), le moniteur peut gérer les élèves, que ce soit ajouter un nouvel élève (Tableau 2), supprimer un élève (Tableau 4), modifier les informations d'un élève (Tableau 3) ou activer/désactiver un compte (Tableau 5).

En cliquant sur le bouton « Ajouter », le moniteur peut inscrire un nouvel élève. Pour ce faire il ne doit inscrire que son nom, prénom et adresse email. L'élève reçoit directement un email avec un lien d'activation. Ce lien donne la possibilité à l'élève d'activer son compte, puis d'y

inscrire le reste des informations nécessaires, comme un identifiant et un mot de passe. Le formulaire à remplir est identique à celui où l'élève s'inscrit seul et vérifie les mêmes erreurs. L'avantage de ce système est que le moniteur peut très rapidement inscrire un nouvel élève et que c'est l'élève lui-même qui inscrit les informations restantes en ne perdant ainsi pas trop de temps durant le cours et en évitant les erreurs.

Par défaut, depuis la page de gestion des élèves, le moniteur voit uniquement les élèves actifs. S'il désire voir également les élèves qui ont été désactivés, ceux qui ne se sont jamais activés ou tout simplement tous les élèves, il suffit au moniteur de changer le choix sur la liste déroulante tout en haut de la page (Figure 12).

Dans le champ de recherche, en commençant à entrer le nom d'un élève, la liste affiche uniquement les élèves dont le nom commence par cette(ces) lettre(s). En cliquant sur le bouton précédant le champ, il est possible de faire le tri par le prénom ou par le code postal.

Le tableau résume tous les élèves qui sont associés au moniteur connecté. Pour une meilleure visibilité, le tableau est limité à 10 lignes, ensuite un système de pagination permet de voir les autres pages. Le système de pagination (Figure 13) n'apparaît que s'il y a plus de 10 entrées dans le tableau.

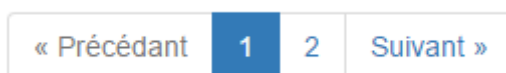


Figure 13 : Capture d'écran « pagination »

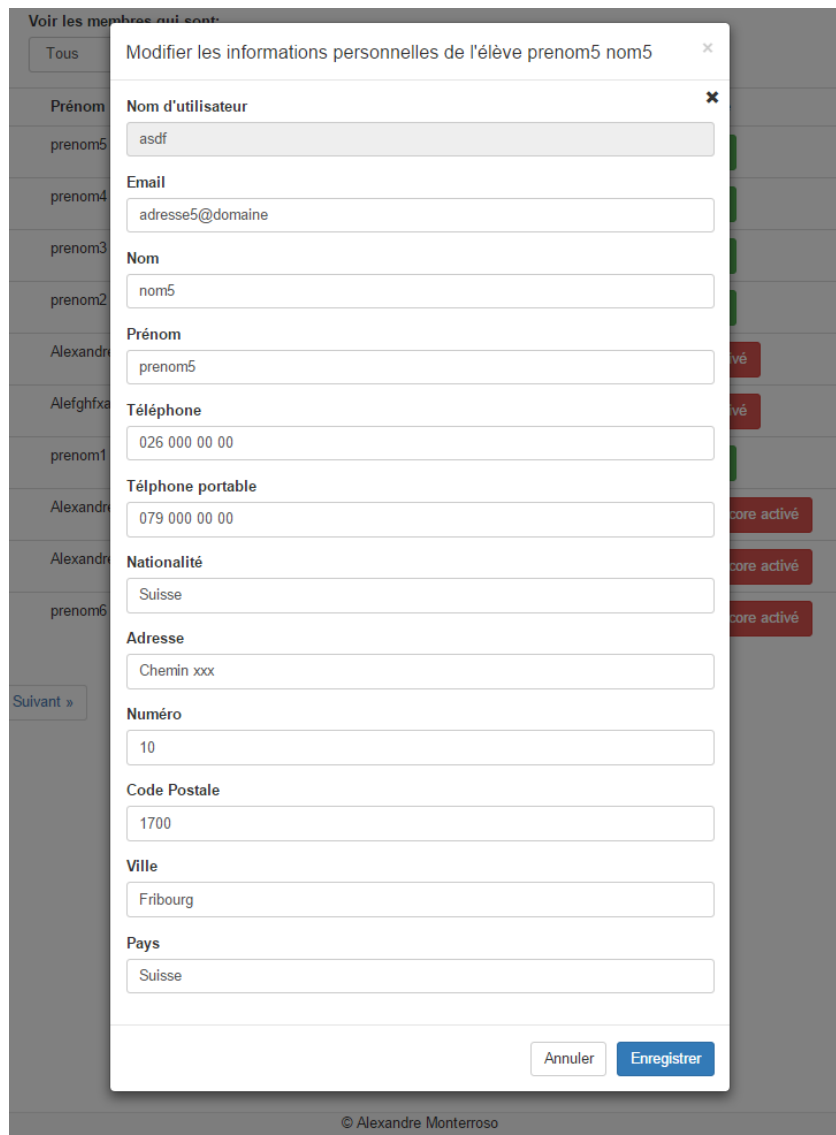
Pour supprimer un élève, il suffit de cliquer sur la petite croix en début de ligne. Une fenêtre modale s'ouvre⁴ (Figure 14) et avertit du danger de la suppression de l'élève. Le moniteur doit confirmer avant de continuer.



Figure 14 : capture d'écran « suppression d'un élève »

Pour modifier un élève, il suffit de cliquer sur le petit crayon à côté de la croix. Une fenêtre modale s'ouvre (Figure 15) et permet de modifier les informations.

⁴ <http://getbootstrap.com/javascript/#modals>, consulté en janvier 2016



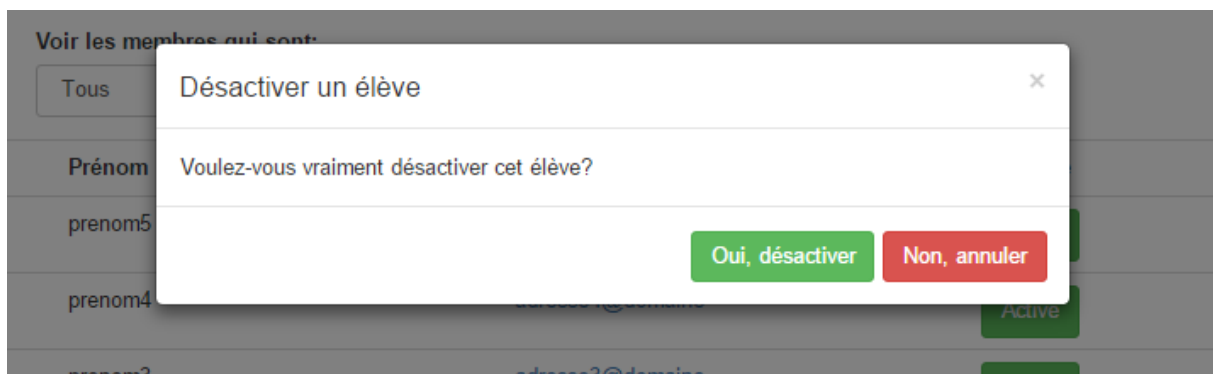
The screenshot shows a modal window titled "Modifier les informations personnelles de l'élève prenom5 nom5". The form contains the following fields:

- Nom d'utilisateur: asdf
- Email: adresse5@domaine
- Nom: nom5
- Prénom: prenom5
- Téléphone: 026 000 00 00
- Téléphone portable: 079 000 00 00
- Nationalité: Suisse
- Adresse: Chemin xxx
- Numéro: 10
- Code Postale: 1700
- Ville: Fribourg
- Pays: Suisse

At the bottom of the modal are two buttons: "Annuler" and "Enregistrer". The background shows a list of students with columns for name, email, and activation status.

Figure 15 : Capture d'écran « modification d'un élève »

Dans la colonne activation, on peut activer/désactiver un élève en cliquant sur le bouton et confirmant l'action dans la fenêtre modale (Figure 16) qui le demande. Si l'élève n'a jamais activé son compte, le moniteur pourra lui renvoyer un lien d'activation.



The screenshot shows a modal window titled "Désactiver un élève". The text inside the modal asks: "Voulez-vous vraiment désactiver cet élève?". At the bottom, there are two buttons: "Oui, désactiver" (green) and "Non, annuler" (red). The background shows the same student list as in Figure 15.

Figure 16 : Capture d'écran « Activation/Désactivation d'un élève »

3.2.3 Types de cours

Un type de cours est le moyen pour que le moniteur puisse choisir quelques informations de base d'un type de cours précis, comme le prix de base ou la couleur qu'il devra être affiché dans le calendrier. Ceci répond aux cas d'utilisation, dont la description est faite dans le Tableau 10 et Tableau 11.

Ajouter un type de cours

[+ Ajouter](#)







Titre	Description	Prix de base	Facture?	Couleur
 Pratique voiture	Ce cours a lieu pour apprendre a conduire une voiture	80.00 CHF	Oui	
 Pratique camion	Ce cours a lieu pour apprendre a conduire un camion	140.00 CHF	Oui	
 Cours théorique privé	Ce cours a lieu sur demande	50.00 CHF	Oui	

Figure 17 : Capture d'écran « Types de cours »

Comme on peut le voir sur la *Figure 17*, on peut ajouter les cours de la même manière que pour la gestion des élèves. Le moniteur fournit les informations suivantes : un titre du cours, une description du cours, un prix de base du cours (qui pourra être changé individuellement si nécessaire), si le cours doit être facturé ou non et la couleur dans laquelle le cours apparaîtra dans le calendrier.

3.2.4 Calendrier

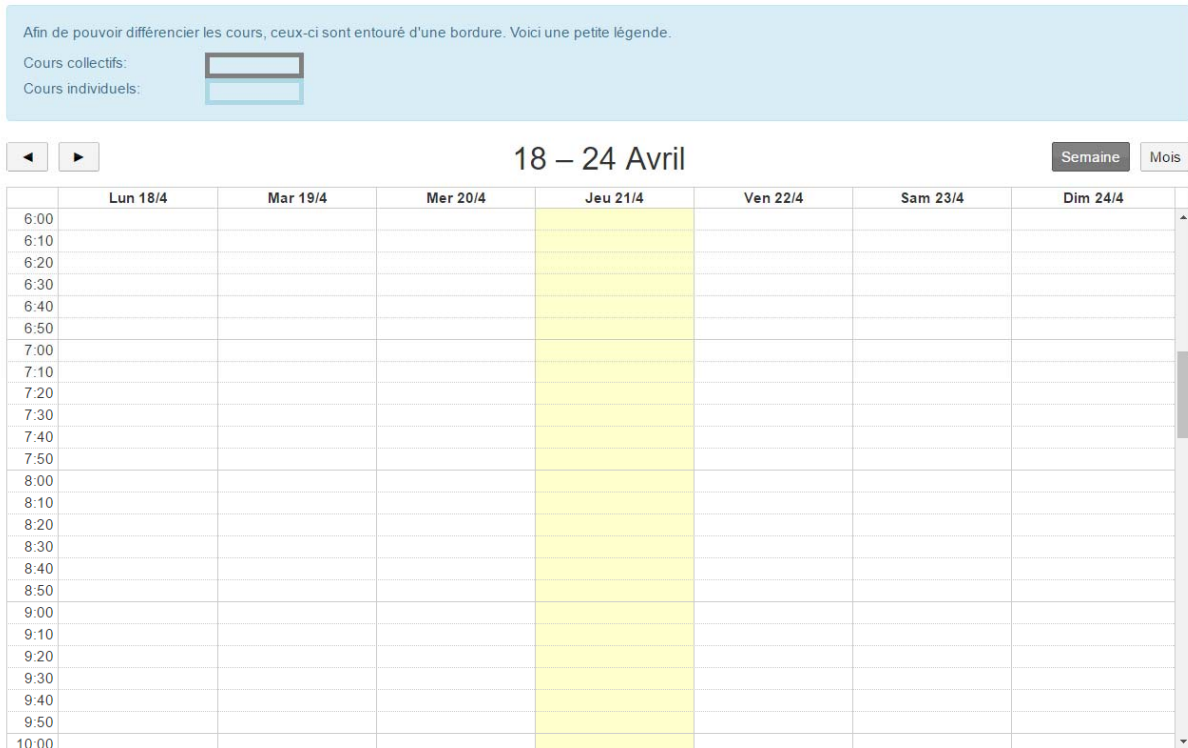


Figure 18 : Capture d'écran « calendrier »

Comme on peut le voir sur la *Figure 18*, un calendrier est affiché avec une vue sur toute la semaine. Le calendrier a un découpage par 10 minutes, ce qui permet au moniteur de planifier ses cours comme il l'entend. Pour réserver un cours le moniteur n'aura qu'à cliquer et glisser sur la tranche horaire qu'il désire. A ce moment-là une fenêtre modale s'ouvre et lui demande d'inscrire les informations nécessaires pour le cours (Tableau 13).

The screenshot shows a web application interface for creating an individual course. A modal dialog box titled "Créer un cours individuel" is open, allowing the user to define course details. The background shows a calendar grid with time slots from 6:00 to 10:00 and days of the week. The dialog box contains the following fields:

- Type de cours:** A dropdown menu with "Pratique voiture" selected.
- Eleves:** A dropdown menu with "Monterroso Alexandre" selected.
- Début du cours:** A date and time selector showing "18 avr. 2016 07:00".
- Fin du cours:** A date and time selector showing "18 avr. 2016 10:00".
- Lieu de rendez-vous:** A text input field containing "Gare de Fribourg".
- Remarques:** A text input field with the placeholder "Veuillez entrer ici vos remarques".
- Enregistrer la modification:** A blue button to save the changes.

Figure 19 : Capture d'écran « Création d'un nouveau cours »

Le moniteur doit choisir parmi la liste de ses types de cours et parmi la liste de ses élèves. Il peut définir plus précisément l'heure de début et de fin, qui ont été pré-remplies selon sa sélection sur le calendrier. Puis il indique le lieu de rendez-vous où l'élève devra attendre le moniteur et pour finir, il peut également noter des remarques éventuelles.

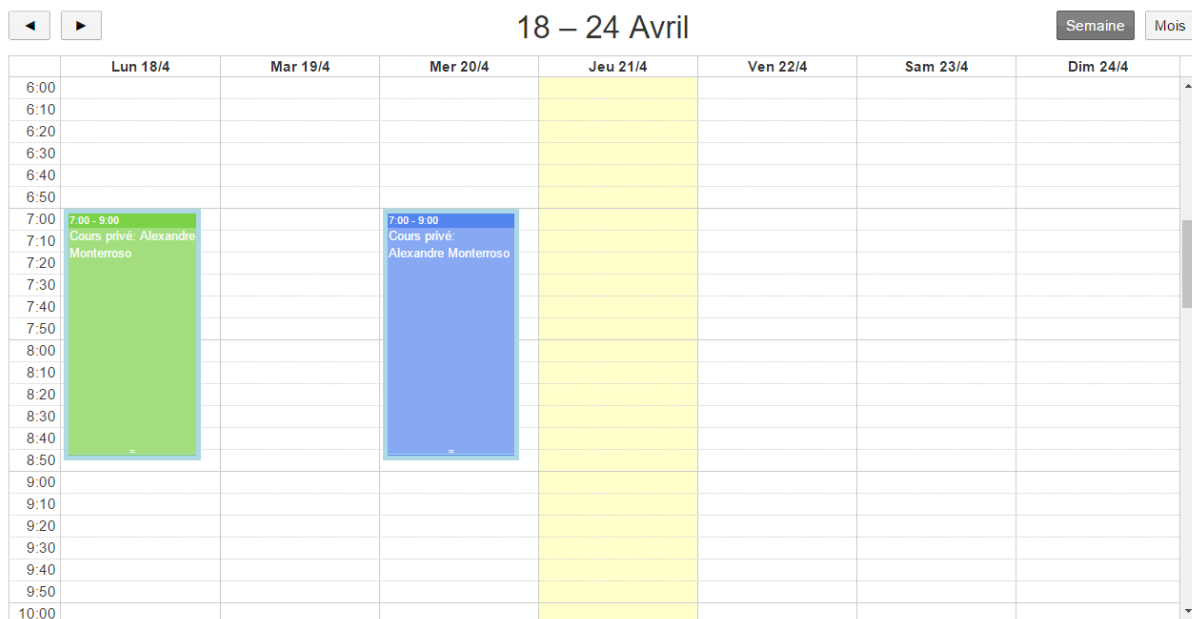


Figure 20 : Capture d'écran « calendrier avec cours »

Une fois enregistré, les cours sont directement affichés sur le calendrier (Figure 20), sans besoin d'actualiser la page. Le cours s'enregistre dynamiquement. En cliquant sur le cours, on a la possibilité de le modifier ou de le supprimer. De plus il est également possible d'agrandir/rétrécir le cours directement depuis le calendrier ou de le déplacer.

3.2.5 Valider un cours

Une fois le cours passé, le moniteur doit valider le cours (Tableau 17), pour dire si le cours a eu lieu ou non, mais aussi si le cours a été directement payé lors du cours ou doit encore être payé.

Monterroso Alexandre (du 18-04-2016 à 07:00 au 18-04-2016 à 09:00)

Description du cours
Ce cours a lieu pour apprendre a conduire une voiture

Durée du cours
2h00

Prix final du cours

Le cours a été payé?
 Non
 Oui, le cours a eu lieu Non, le cours n'a pas eu lieu

Monterroso Alexandre (du 20-04-2016 à 07:00 au 20-04-2016 à 09:00)

Figure 21 : Capture d'écran « valider un cours »

Dans l'onglet validation, une liste des cours passés est affichée (*Figure 21*). En cliquant sur un cours, le moniteur voit la durée du cours, et le prix de celui-ci, calculé à partir du prix de base et de la durée du cours. Le moniteur peut entrer le prix final si celui-ci doit être différent et valider le cours ou annoncer que le cours n'a pas eu lieu.

3.2.6 Liste des cours

Lorsque les cours ont été validés, ils apparaissent dans la liste des cours.

The screenshot shows a web interface for course management. At the top, there are two date pickers: 'Début de la période' (01.03.2016) and 'Fin de la période' (30.04.2016). Below them is a dropdown menu for 'Eleves' with 'Alexandre Monterroso' selected. The main part of the interface is a table with the following data:

Nom	Prénom	Début du cours	Fin du cours	Prix final	Payé?
Monterroso	Alexandre	18-04-2016 07:00	18-04-2016 09:00	160.00 CHF	Non
Monterroso	Alexandre	20-04-2016 07:00	20-04-2016 09:00	280.00 CHF	Oui

At the bottom right, there is a summary: Total: 440 CHF, Déjà payé: 280 CHF, Reste à payer: 160 CHF.

Figure 22 : Capture d'écran « liste des cours »

La liste des cours (Tableau 9) est affichée avec les informations nécessaires. Il est également précisé si le cours a été déjà payé ou non. Un total est calculé et ce qu'il reste à payer est également affiché.

Il est possible de trier la liste des cours. Par défaut, la période des cours correspond au mois précédent, vu que le moniteur veut en principe faire une facturation pour le mois qui précède. Mais le début et la fin de la période peuvent être changés très facilement. Il est également possible de choisir si on affiche tous les cours de ces périodes ou seulement les cours d'un seul élève. Pour cela on choisit un élève dans la liste.

3.3 Fonctions de le secrétaire

Dans ce chapitre nous allons expliquer les différentes fonctions qui sont offertes à le secrétaire.

3.3.1 Gestion des moniteurs

Un secrétaire peut gérer les moniteurs, inscrire un nouveau moniteur (Tableau 28), modifier le moniteur (Tableau 29) ou en supprimer un (Tableau 30). Depuis son espace client, le secrétaire peut accéder à un tableau identique à la gestion des élèves (*Figure 12*) décrit plus haut.

3.3.2 Gestion des secrétaires

Comme pour le moniteur, le secrétaire peut gérer les autres secrétaires depuis son espace privé (Tableau 31, Tableau 32, Tableau 33).

3.4 Fonctions de l'élève

Les fonctions qui sont offertes aux élèves sont diverses et expliquées dans ce chapitre.

3.4.1 Calendrier

L'élève peut voir également son calendrier avec tous les rendez-vous qu'il a fixés avec son moniteur. Il ne peut par contre rien modifier, puisque seul le moniteur peut modifier le contenu du calendrier.

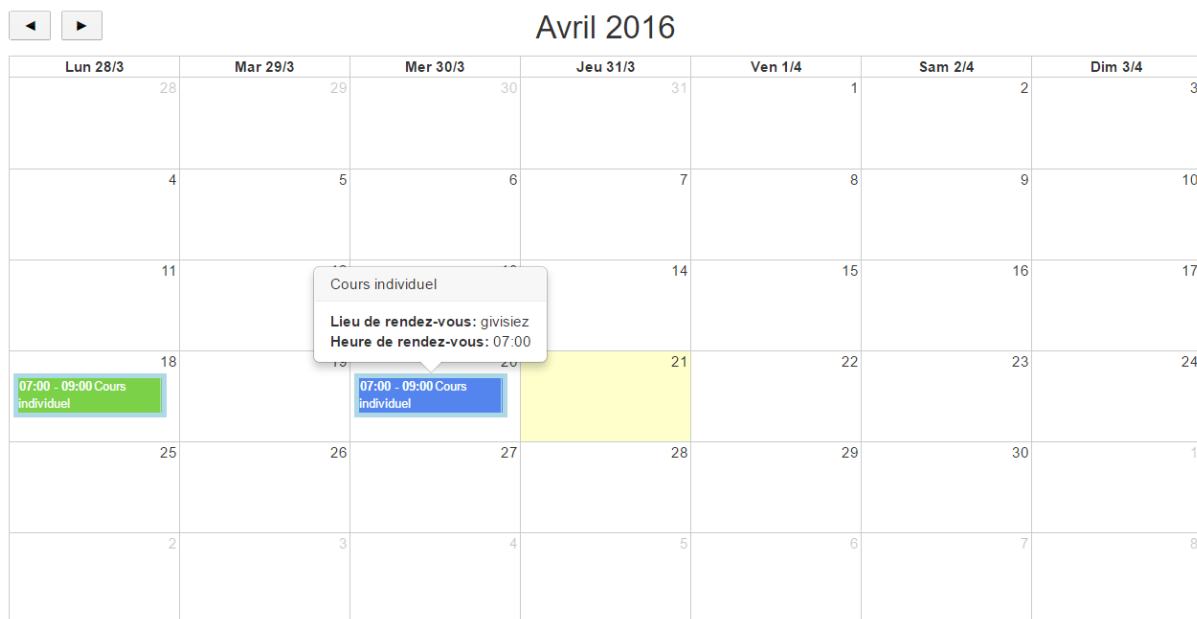


Figure 23 : Capture d'écran « calendrier de l'élève »

Comme on peut le voir sur la Figure 23, l'élève voit les rendez-vous fixés dans une vue du mois entier. S'il passe la souris par-dessus son cours, quelques informations supplémentaires comme l'heure exacte du rendez-vous et le lieu du rendez-vous s'affichent.

3.4.2 Liste des cours

L'élève peut également voir la liste complète des cours qu'il a réalisés (Figure 24) avec le prix que le cours a coûté, la mention si le cours a été payé ou non, ainsi que le total qu'il a déjà payé et ce qu'il lui reste à payer.

Nom	Prénom	Début du cours	Fin du cours	Prix final	Payé?
NomEleve1	PrénomEleve1	18-04-2016 07:00	18-04-2016 09:00	160.00 CHF	Non
NomEleve1	PrénomEleve1	20-04-2016 07:00	20-04-2016 09:00	280.00 CHF	Oui

Total: 440.00 CHF
Déjà payé: 280.00 CHF
Reste à payer: 160.00 CHF

Figure 24 : Capture d'écran « liste des cours de l'élève »

4

Présentation de l'implémentation

Le chapitre 4 présente l'implémentation des données avec un modèle entité-association, puis sa transformation en base de données relationnelles. Le Framework qui a été utilisé pour cette implémentation, est également présenté avant de clore ce chapitre par l'étude de cas pratique avec la réalisation d'un test unitaire pour un formulaire du site.

4.1 Modèle entité-association

Afin de pouvoir mieux présenter le modèle des données, le modèle entité-association est expliqué et présenté : celui-ci permet de comprendre les liens qui existent entre les différentes entités et leurs associations.

Le modèle des données se compose de trois parties : les utilisateurs, les cours individuels et les cours collectifs. Pour une meilleure visibilité, ceux-ci sont dessinés en différentes couleurs sur le modèle entité-association (Figure 25) ainsi que sur le modèle de base de données relationnelles (Figure 26) : en rouge pour les utilisateurs, en bleu pour les cours collectifs et en vert pour les cours individuels.

Utilisateurs (rouge) :

Le système contient plusieurs utilisateurs. Un utilisateur peut être soit un élève, soit un moniteur, soit un secrétaire. Il ne peut s'agir qu'une de ces trois possibilités.

Un élève n'est lié qu'à un seul moniteur. Uniquement ce moniteur aura accès aux données de cet élève ou pourra l'inscrire à un cours que ce soit individuel ou collectif. Cependant, un moniteur peut avoir plusieurs élèves.

Cours :

Il existe deux sortes de cours, les cours individuels et les cours collectifs. Un cours ne peut être qu'une de ces deux possibilités.

Cours individuels (vert) :

Un élève peut être inscrit à un ou plusieurs cours individuels, tandis qu'un cours individuel ne contient qu'un seul élève.

Un cours individuel a exactement un type de cours, tandis qu'un type de cours peut être dans plusieurs cours individuels.

Un type de cours contient au plus un permis, tandis qu'un permis peut être contenu dans plusieurs types de cours.

Un type de cours est créé par exactement un moniteur, tandis qu'un moniteur peut créer plusieurs types de cours.

Un élève peut posséder plusieurs permis et un permis peut être possédé par plusieurs élèves.

Cours collectifs (bleu) :

Un élève peut être inscrit à un ou plusieurs cours collectifs, tandis qu'un cours collectif contient plusieurs élèves.

Un cours collectif contient au moins un moniteur, tandis qu'un moniteur peut être inscrit dans plusieurs cours collectifs.

D'après ces contraintes, nous avons créé un diagramme entité-association qui décrit le modèle de données :

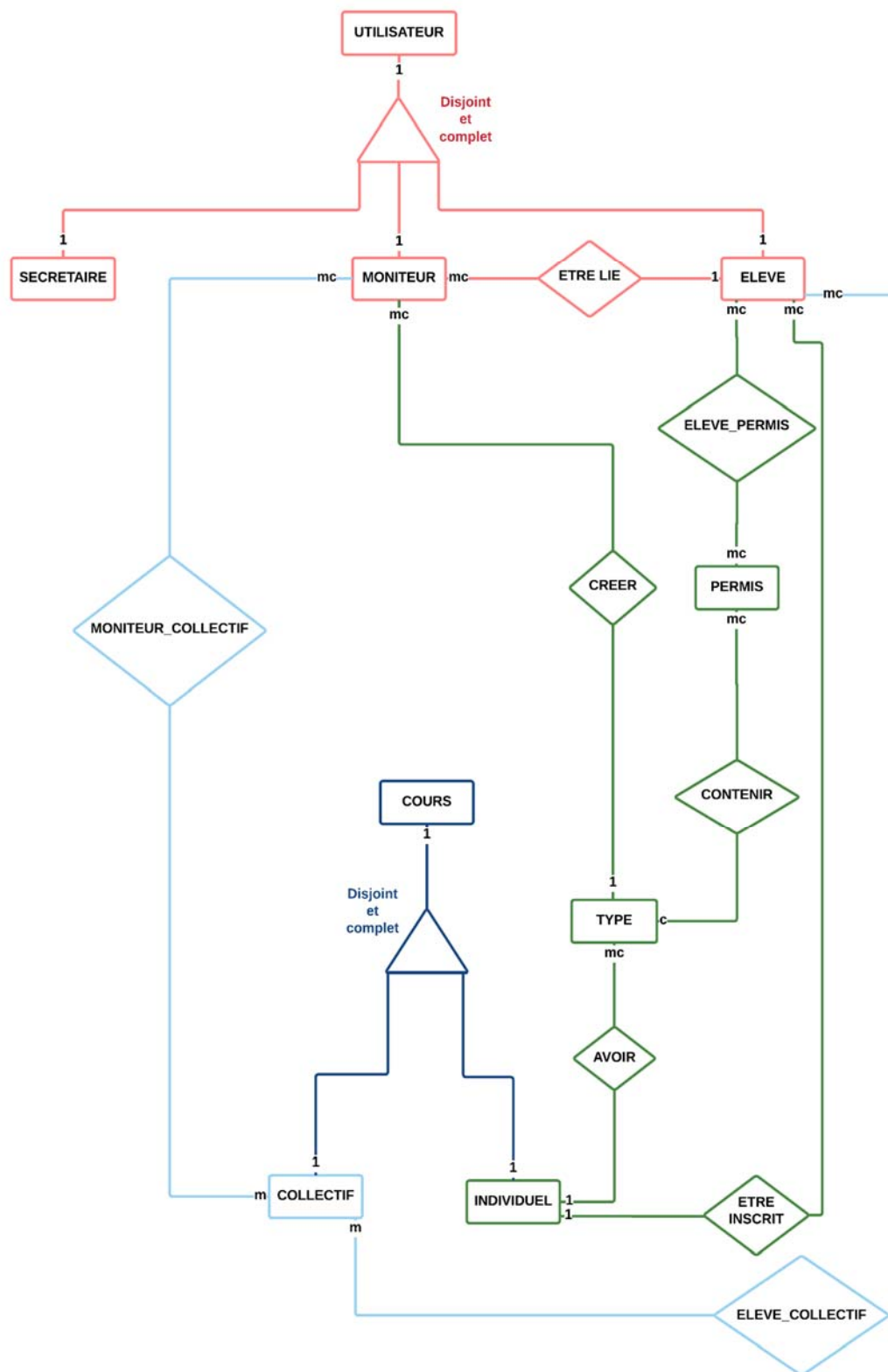


Figure 25 : Modèle entité-association

4.2 Transformation des entités en tables

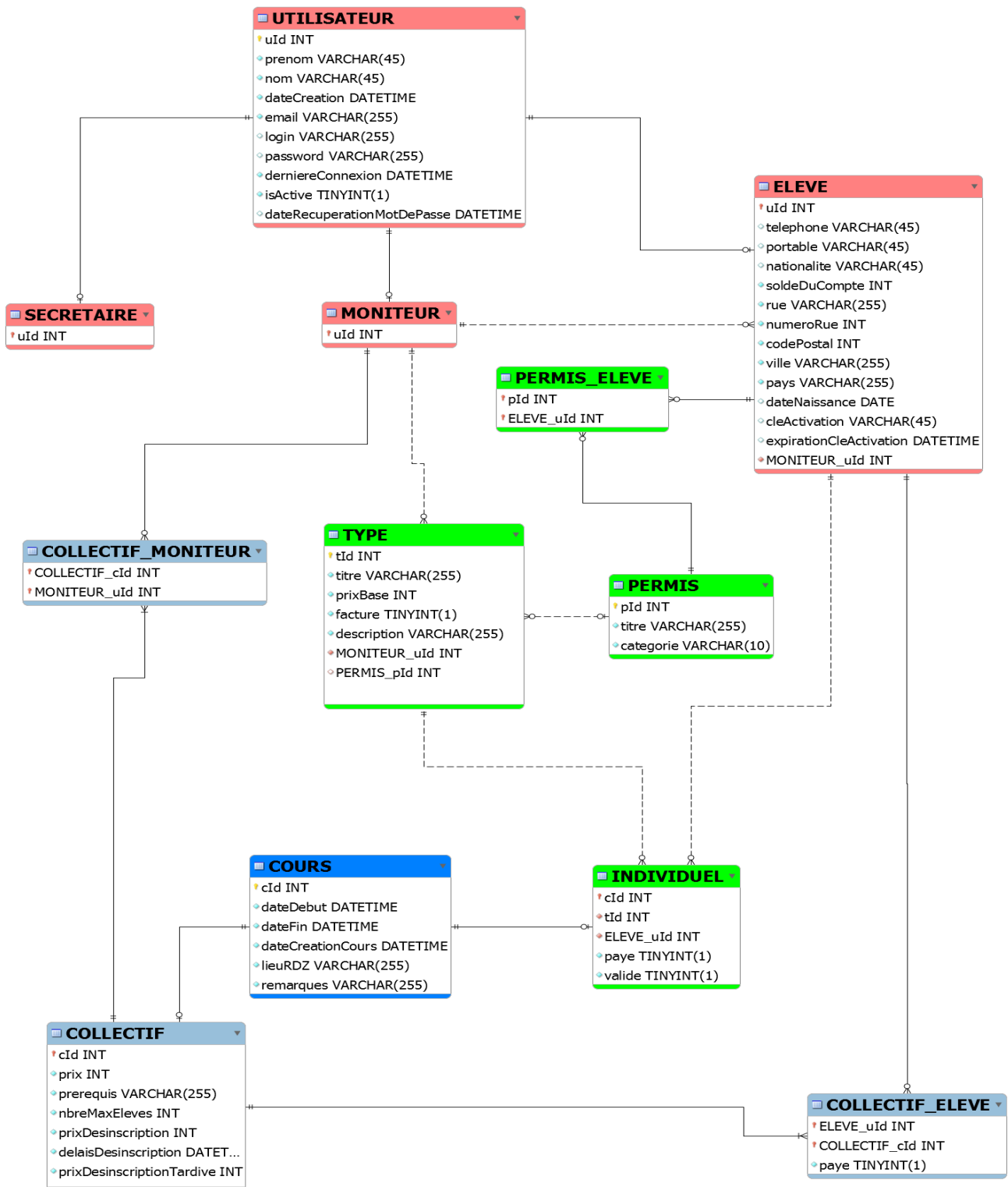


Figure 26 : modèle de bases de données relationnelle

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	uld	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes

Figure 29 : Table « Secrétaire »

- Moniteur

Comme pour le secrétaire, le moniteur ne contient aucun attribut (Figure 30), mais ceux-ci pourront être ajoutés par la suite, si cela s'avère nécessaire.

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	uld	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes

Figure 30 : Table « Moniteur »

4.2.2 Cours

- Cours

La table cours (Figure 31) contient les informations générales que tous les cours doivent avoir, comme la date de début du cours, la date de fin, la date de création du cours, le lieu du rendez-vous et les remarques éventuelles.

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	cld	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
2	dateDebut	datetime			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
3	dateFin	datetime			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
4	dateCreationCours	datetime			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
5	lieuRDZ	varchar(255)	utf8_general_ci		Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
6	remarques	varchar(255)	utf8_general_ci		Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes

Figure 31 : Table « Cours »

- Individuel

Les cours individuels (Figure 32) ont obligatoirement un type de cours. La table contient également l'élève pour lequel le cours est créé. Il contient par la même occasion l'information qui indique si le cours a été payé et si le cours a été validé par le moniteur après le cours (cela permet au moniteur de confirmer que le cours a eu lieu).

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	cld	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
2	tld	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
3	ELEVE_uld	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
4	paye	tinyint(1)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
5	valide	tinyint(1)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes

Figure 32 : Table « Individuel »

- Collectif

Un cours collectif (Figure 33) contient certaines informations nécessaires comme le prix du cours, les prérequis nécessaires pour s'inscrire au cours (qui sont juste affichés et pas

Une table type de cours (*Figure 38*) individuel sert à enregistrer les valeurs de base qui seront utiles dans la génération d'un cours individuel lié à un permis. Il contient donc un titre, un prix de base (qui peut être modifié par la suite), une mention pour savoir si ce type doit être facturé (un type privé par exemple ne doit pas forcément être payé, mais doit apparaître comme cours dans la base de données pour réserver une plage horaire) et une description.

Il se peut qu'un type ne contienne pas de permis (cela peut être le cas par exemple dans un type « privé »). Dans les autres cas, on doit inscrire l'id du permis auquel le type de cours est relié. La table contient également l'id du moniteur qui a créé le type de cours.

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	id	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
2	titre	varchar(255)	utf8_general_ci		Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
3	prixBase	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
4	facture	tinyint(1)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
5	description	varchar(255)	utf8_general_ci		Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
6	MONITEUR_uid	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
7	PERMIS_pld	int(11)			Oui	NULL		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes

Figure 38 : Table « Type de cours »

4.3 Contraintes

Certaines contraintes sont nécessaires pour rendre la base de données robuste et consistante. Si elle ne l'était pas, des éléments pourraient ne pas être associés avec d'autres ; par conséquent elle deviendrait non consistante. Ces différentes contraintes sont détaillées dans les sections suivantes.

OnDELETE

Les entrées dans la base de données peuvent être supprimées. Afin qu'il ne reste pas d'éléments qui ne sont pas accessibles après une suppression dans une autre table liée à celle-ci ou qui deviendraient non consistants après une modification, il faut ajouter des contraintes *onDelete*, expliquées de manière détaillée.

Plusieurs contraintes peuvent être appliquées aux différentes clés. Pour chaque clé on pourra avoir :

- Cascade

Cette contrainte effacera également l'élément lié.

- *Restrict*

Cette contrainte empêchera que l'élément ne soit effacé.

- *Set Null*

Cette contrainte appliquera la valeur *NULL* à l'attribut qui contient la clé supprimée.

- *No action*

C'est la valeur par défaut, aucune contrainte n'est appliquée.

- Elève (Figure 39) :

- **uId** : En effaçant un utilisateur, il faut effacer également les éléments de la table élève qui correspondent. On appliquera donc une contrainte *onDelete CASCADE*.

- **MONITEUR_uId** : En effaçant un moniteur, il faut effacer tous les élèves qui y sont associés. On appliquera donc une contrainte *onDelete CASCADE*.

Relations			
Colonne	Contrainte de clé étrangère (INNODB)		
uld	mydb	utilisateur	uld
	Nom de la contrainte: fk_ELEVE_UTILISATEUR1		
	ON DELETE	CASCADE	
	ON UPDATE	NO ACTION	
telephone	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
portable	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
nationalite	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
soldeDuCompte	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
rue	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
numeroRue	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
codePostal	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
ville	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
pays	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
dateNaissance	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
cleActivation	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
expirationCleActivation	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
MONITEUR_uld	mydb	moniteur	uld
	Nom de la contrainte: fk_ELEVE_MONITEUR1		
	ON DELETE	CASCADE	
	ON UPDATE	NO ACTION	

Figure 39 : Contraintes « Elève »

- Moniteur (Figure 40):
 - **uld** : En effaçant un utilisateur, il faut effacer également les éléments de la table moniteur qui correspondent. On appliquera donc une contrainte *onDelete CASCADE*.

Relations			
Colonne	Contrainte de clé étrangère (INNODB)		
uld	mydb	utilisateur	uld
	Nom de la contrainte: fk_MONITEUR_UTILISATEL		
	ON DELETE	CASCADE	
	ON UPDATE	NO ACTION	

Figure 40 : Contraintes « Moniteur »

- Secrétaire :

- **uId** : En effaçant un utilisateur, il faut effacer également les éléments de la table secrétaire qui correspondent. On appliquera donc une contrainte *onDelete CASCADE*.

Relations			
Colonne	Contrainte de clé étrangère (INNODB)		
uld	mydb ▼	utilisateur ▼	uld ▼
Nom de la contrainte <input type="text" value="fk_SECRETAIRE_UTILISAT"/>			
ON DELETE	CASCADE ▼		
ON UPDATE	NO ACTION ▼		

Figure 41 : Contraintes « Secrétaire »

- Type (Figure 42):
 - **MONITEUR_uId** : En effaçant un moniteur il faut également effacer tous les types qui sont associés à ce moniteur. On appliquera donc une contrainte *onDelete CASCADE*.
 - **PERMIS_pId** : En effaçant un permis, il faut mettre la valeur du permis à *NULL*. On appliquera donc une contrainte *onDelete SET NULL*.

Relations			
Colonne	Contrainte de clé étrangère (INNODB)		
tld	mydb		
titre	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
prixBase	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
facture	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
description	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
MONITEUR_uld	mydb	moniteur	uld
	Nom de la contrainte fk_TYPE_MONITEUR1		
	ON DELETE	CASCADE	
	ON UPDATE	NO ACTION	
PERMIS_pld	mydb	permis	pld
	Nom de la contrainte fk_TYPE_PERMIS1		
	ON DELETE	SET NULL	
	ON UPDATE	NO ACTION	
archive	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		

Figure 42 : Contraintes « Type »

- PERMIS_ELEVE (Figure 43):
 - **pld** : En effaçant un permis, il faut également effacer les entrées le concernant dans PERMIS_ELEVE. On appliquera donc une contrainte *onDelete CASCADE*.
 - **ELEVE_uld** : En effaçant un élève, il faut également effacer les entrées le concernant dans PERMIS_ELEVE. On appliquera donc une contrainte *onDelete CASCADE*.

Relations		
Colonne	Contrainte de clé étrangère (INNODB)	
pId	mydb	permis
		pId
	Nom de la contrainte: fk_PERMIS_has_ELEVE_PE	
	ON DELETE	CASCADE
	ON UPDATE	NO ACTION
ELEVE_uId	mydb	eleve
		uId
	Nom de la contrainte: fk_PERMIS_has_ELEVE_EL	
	ON DELETE	CASCADE
	ON UPDATE	NO ACTION

Figure 43 : Contraintes « Permis-Elève »

- Collectif (Figure 44) :
 - **cId** : En effaçant un cours, il faut également effacer le cours collectif y faisant référence. On appliquera donc une contrainte *onDelete cascade*.

Relations		
Colonne	Contrainte de clé étrangère (INNODB)	
cId	mydb	cours
		cId
	Nom de la contrainte: fk_COLLECTIF_COURS1	
	ON DELETE	CASCADE
	ON UPDATE	NO ACTION
prix	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas	
prerequis	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas	
nbreMaxElevés	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas	
prixDesinscription	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas	
delaisDesinscription	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas	
prixDesinscriptionTardive	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas	

Figure 44 : Contraintes « Cours collectif »

- COLLECTIF_ELEVE (Figure 45) :
 - **ELEVE_uId** : En effaçant un élève, il faut aussi effacer toutes ses participations aux différents cours collectifs. Par conséquent il faut appliquer une contrainte *onDelete CASCADE*.
 - **COLLECTIF_cId** : En effaçant un cours collectif, il faut également effacer les inscriptions des élèves à ce cours. On appliquera par conséquent une contrainte *onDelete CASCADE*.

Relations		
Colonne	Contrainte de clé étrangère (INNODB)	
ELEVE_uId	mydb ▼	eleve ▼
		uId ▼
	Nom de la contrainte fk_ELEVE_has_COLLECTIF	
	ON DELETE	CASCADE ▼
	ON UPDATE	NO ACTION ▼
COLLECTIF_cId	mydb ▼	collectif ▼
		cId ▼
	Nom de la contrainte fk_ELEVE_has_COLLECTIF	
	ON DELETE	CASCADE ▼
	ON UPDATE	NO ACTION ▼
paye	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas	

Figure 45 : Contraintes « Collectif-Elève »

- COLLECTIF_MONITEUR (Figure 46):
 - **COLLECTIF_cId** : En effaçant un cours collectif, il faut également effacer les participations des moniteurs à ce cours. Par conséquent il faut appliquer une contrainte *onDelete CASCADE*.
 - **MONITEUR_uId** : En effaçant un moniteur, il faut aussi effacer ses participations aux différents cours collectifs. On appliquera par conséquent une contrainte *onDelete CASCADE*.

Relations		
Colonne	Contrainte de clé étrangère (INNODB)	
COLLECTIF_cld	mydb ▼	collectif ▼ cld ▼
	Nom de la contrainte fk_COLLECTIF_has_MONITI	
	ON DELETE	CASCADE ▼
	ON UPDATE	NO ACTION ▼
MONITEUR_uld	mydb ▼	moniteur ▼ uld ▼
	Nom de la contrainte fk_COLLECTIF_has_MONITI	
	ON DELETE	CASCADE ▼
	ON UPDATE	NO ACTION ▼

Figure 46 : Contraintes « Collectif-Moniteur »

- Individuel (Figure 47):
 - **cId** : En effaçant un cours, il faut également effacer le cours individuel qui y fait référence. On appliquera donc une contrainte *onDelete CASCADE*.
 - **tId** : On ne doit pas pouvoir effacer un type, si un cours individuel existe utilisant ce type. On appliquera donc une contrainte *onDelete RESTRICT*.
 - **ELEVE_uId** : En effaçant un élève, il faut également effacer les cours individuels qui contiennent cet élève. On appliquera par conséquent une contrainte *onDelete CASCADE*.

Relations			
Colonne	Contrainte de clé étrangère (INNODB)		
cld	mydb	cours	cld
	Nom de la contrainte: fk_INDIVIDUEL_COURS1		
	ON DELETE	CASCADE	
	ON UPDATE	NO ACTION	
tld	mydb	type	tld
	Nom de la contrainte: fk_Individuel_Type1		
	ON DELETE	CASCADE	
	ON UPDATE	CASCADE	
ELEVE_uld	mydb	eleve	uld
	Nom de la contrainte: fk_INDIVIDUEL_ELEVE1		
	ON DELETE	CASCADE	
	ON UPDATE	NO ACTION	
paye	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		
valide	Aucun index n'est défini ! Créez-en un dans le dialogue plus bas		

Figure 47 : Contraintes « Individuel »

4.4 Description du Framework Symfony 2

Les fonctions principales du Framework PHP⁵ utilisé pour le développement de l'application sont expliquées dans cette section.

4.4.1 Framework

Un Framework est un ensemble d'outils aidant à la conception d'un logiciel ou d'un site web. Il permet d'organiser la conception autour de ses principes, d'introduire de bonnes pratiques, évitant qu'on doive à chaque nouvelle conception repartir de zéro.

Comparé à une simple librairie, le Framework pose des bases : la programmation se fait à l'intérieur du Framework, tandis qu'avec une librairie, la conception originale est programmée puis des librairies externes sont utilisées. Un Framework est un outil qui n'est pas forcément spécialisé dans une fonction, mais qui intègre plusieurs librairies spécialisées.

Un Framework a l'avantage de guider dans l'architecture logicielle, obligeant l'utilisateur à respecter certains principes, comme des patrons de conception⁶.

Durant ce travail, Symfony2 est utilisé comme Framework afin de profiter de son architecture et de ses différents composants. Ce Framework PHP permet d'utiliser un modèle-vue-contrôleur, qui permet de bien séparer l'affichage du code. Il offre un ensemble de composants, découplés et réutilisables, afin d'utiliser les bonnes pratiques et la

⁵ <https://secure.php.net>, consulté en mars 2016

⁶ https://fr.wikipedia.org/wiki/Patron_de_conception, consulté en mars 2016

standardisation de la programmation en PHP. Le principe de base du Framework est qu'il n'est pas nécessaire de réinventer la roue. Ainsi comme certains composants ont déjà été développés, les réutiliser est plus efficace que les développer.

D'autre part, l'utilisation du Framework permet de maintenir plus facilement le logiciel. Une nouvelle équipe peut ainsi plus facilement s'y intégrer.

4.4.2 Modèle-vue-contrôleur

Le modèle-vue-contrôleur (MVC) est un patron de conception qui permet d'organiser le code. Il sépare la partie qui correspond à l'organisation des données, celle qui contient l'interface avec l'utilisateur et celle qui contient la programmation à proprement parler. Elle permet d'avoir un code bien séparé et très lisible ce qui le rend facile à maintenir.

Le modèle MVC est présent dans le Framework Symfony2. La séparation est clairement définie afin de faciliter l'utilisation et la programmation.

- Modèle

Le modèle correspond à la manière dont les données sont stockées. En général il s'agit de la base de données.

Symfony2 utilise Doctrine⁷ pour la gestion de la base de données. Bien que ce ne soit pas obligatoire, cela facilite grandement l'abstraction des données en utilisant des objets pour les gérer.

- Contrôleur

Le contrôleur est la partie algorithmique qui fait des actions sur les données. C'est la partie principale du code PHP.

Dans Symfony2 il s'agit de la logique de l'application. Aucun code HTML n'y est présent, le contrôleur retourne les données nécessaires après avoir fait les actions nécessaires à la vue.

- *Template*

Le *Template* est la vue dans Symfony2, l'affichage vu par l'utilisateur. Symfony2 utilise le langage TWIG⁸ afin de rajouter les données aux balises HTML. TWIG est là pour exprimer la représentation et non pour faire des actions sur les données.

La puissance de TWIG est qu'il permet de faire de l'héritage : ainsi un *Template* de base est utilisé par toutes les pages et est complété à chacune d'entre elles par le contenu de la page en question. Ce *Template* de base peut tant contenir le menu du site que le graphisme de la page.

⁷ <http://www.doctrine-project.org>, consulté en mars 2016

⁸ <http://twig.sensiolabs.org>, consulté en mars 2016

4.4.3 Configuration

La première partie du Framework contient tout ce qui concerne la configuration du Framework. Il a besoin d'un certain nombre de paramètres comme l'adresse de la base de données, le système utilisé pour envoyer des emails, etc.

Concernant les emails, afin de ne pas créer à chaque fois un nouveau système d'envoi d'emails, avec toutes les complications que cela engendre au niveau des mécanismes anti-spam entre autres, Symfony2 intègre une librairie externe « Swift Mailer⁹ ». Cette librairie propose plusieurs moyens pour le transport d'emails. Nous avons choisi d'utiliser le serveur de Gmail¹⁰ pour le transport des emails. Pour cela il a fallu créer un compte Gmail dédié à cette utilisation.

4.4.4 Le routage

Les URL d'un site internet sont une partie importante du site. Les URLS du type <http://www.monsite.com/id=22&name=oscar> ne sont pas très jolies, et pour un site professionnel, elles ne sont pas du meilleur effet. Tout bon site doit donc se doter d'un bon système de routage. De plus, si l'on veut changer ce qui est écrit dans l'URL, il faut alors mettre à jour de nombreux liens sur le site, ce qui risque de causer des erreurs.

Pour éviter cela, Symfony 2 s'est doté d'un puissant système de routage. Un fichier *routing* contient toutes les routes qui lient un chemin d'une URL vers un contrôleur qui contiendra le code métier et fera appel ensuite à une vue.

4.4.5 Contrôleur

Un contrôleur est appelé à chaque fois que l'utilisateur entre une route qui lui est associé. Le contrôleur effectue un certain nombre d'actions en PHP, puis renvoie une vue à l'utilisateur. Le contrôleur peut renvoyer un certain nombre d'arguments à la vue, pour qu'elle affiche les données que le contrôleur a traitées et qu'il a, par exemple, obtenues depuis la base de données.

4.4.6 *Template* de base

TWIG contient un système pouvant faire de l'héritage assez simplement. Ainsi, nous utiliserons cette fonctionnalité afin de pouvoir faire un *Template* de base contenant les éléments qui ne vont pas être modifiés au cours de l'exécution du programme, telle que le graphisme de l'application ou le menu du site.

A chaque exécution d'une route, le *Template* de base est appelé et le contenu correspondant à la route est ajouté à la page en question.

⁹ <http://swiftmailer.org>, consulté en mars 2016

¹⁰ <http://www.gmail.com>, consulté en mars 2016

Durant le projet, pour garantir de meilleurs affichage et ergonomie, un certain nombre d'autres technologies ont été utilisées, dont voici les principales.

- **Bootstrap**¹¹

Bootstrap facilite la mise en page du site. Grâce à la puissance de Bootstrap, la mise en page s'adapte automatiquement aux différentes tailles d'écran. L'application devient par conséquent *responsive*¹² et l'affichage est uniformisé.

- **JQuery**¹³

JQuery est une librairie JavaScript¹⁴, qui permet de manipuler les éléments d'une page HTML rapidement et facilement. Elle permet aussi de gérer facilement les événements de la page.

- **Ajax**¹⁵

Ajax intègre JavaScript ce qui permet de communiquer avec le serveur de manière asynchrone : le navigateur du client communique avec le serveur en arrière-plan et ainsi, il n'est plus nécessaire de recharger toute la page. Uniquement le contenu qu'il faut modifier sera envoyé au navigateur. Cette technologie sera beaucoup utilisée lors du projet afin de dynamiser l'application. L'utilisateur pouvant faire beaucoup d'actions sans que le rechargement de la page ne lui soit visible.

4.4.7 Aide au développement de Symfony2

Symfony2 contient d'autre part, un certain nombre d'outils qui peuvent être utiles pour le développement d'applications. Il s'agit de tout un environnement dont seules les fonctions principales sont expliquées.

Symfony2 contient un mini serveur web, permettant d'exécuter du code PHP en local, sans avoir besoin d'installer tout un serveur web.

L'application peut être exécutée de deux manières différentes. Pendant le développement, une version permet d'avoir des erreurs précises, lesquelles ne sont pas souhaitables lors d'une mise en production, afin notamment de ne pas fournir trop d'informations au niveau de la sécurité à quelqu'un de malveillant. Lors d'une mise en production, la deuxième version, elle, ne contient que des erreurs génériques, sans aucune information.

Symfony2 contient un fichier log pour le développement et un autre pour la mise en production. Tous les événements et erreurs y sont répertoriés, pour que le développeur puisse voir les erreurs exactes qui se sont produites.

De plus, Symfony2 contient également une barre de débogage. A partir de cette barre, toutes les informations sur le développement et l'aide au développement sont visibles. Cette barre n'est disponible que dans la version développement. Elle sera supprimée dans la version de production, essentiellement pour des raisons de sécurité.

¹¹ <http://getbootstrap.com>, consulté en mars 2016

¹² http://www.w3schools.com/html/html_responsive.asp, consulté en mars 2016

¹³ <https://jquery.com>, consulté en mars 2016

¹⁴ <http://www.w3schools.com/js>, consulté en mars 2016

¹⁵ <http://api.jquery.com/jquery.ajax>, consulté en mars 2016

4.5 Etude de cas : Tests unitaires

Dans cette section, la création d'un test unitaire pour un formulaire est étudiée en détail pour analyser la mise en pratique.

Pour toute application, une nécessité fondamentale après chaque implémentation est de la tester. Afin d'éviter des comportements indésirables, il faut tester chaque fonctionnalité de l'application, avec plusieurs données différentes, afin de voir si l'application se comporte tel qu'on le désire.

Pour cette phase de test, le Symfony2 intègre une librairie de tests unitaires¹⁶ en PHP. Cette librairie s'appelle PHPUnit¹⁷.

Cela permet de tester chaque classe du programme GestAuto et chaque méthode afin de garantir, qu'en cas de modification du code source, que ce soit pour de la maintenance, l'ajout de nouvelles fonctionnalités ou autres, cela continuera à fonctionner tel qu'avant, sans rajouter des problèmes à d'autres endroits de l'application qui fonctionnaient très bien auparavant.

Un exemple de test unitaire est expliqué sur une classe de formulaire de l'application.

```
1 class SecretaireCreationTypeTest extends TypeTestCase {
2
3     public function testSubmitValidData() {
4         $formData = array(
5             'username' => 'user',
6         );
7
8         $type = new \AppBundle\Form\SecretaireCreationType();
9         $form = $this->factory->create($type);
10        $object = new Secretaire();
11
12        $object->setUsername($formData['username']);
13
14        $form->submit($formData);
15
16        $this->assertTrue($form->isSynchronized());
17        $this->assertEquals($object, $form->getData());
18
19        $view = $form->createView();
20        $children = $view->children;
21
22        foreach (array_keys($formData) as $key) {
23            $this->assertArrayHasKey($key, $children);
24        }
25    }
26
27 }
```

Code Source 1 : Test unitaire pour le formulaire « création de secrétaire »

¹⁶ https://fr.wikipedia.org/wiki/Test_unitaire, consulté en mars 2016

¹⁷ <https://phpunit.de>, consulté en mars 2016

Le Code Source 1 est un test unitaire pour le formulaire de création d'une nouvelle secrétaire. Nous allons tester la soumission des données à travers ce formulaire. Nous utilisons pour cela, la fonction de PHPUnit « testSubmitValidData() » qui permet de tester la soumission du formulaire.

```
1     $formData = array(  
2         'username' => 'user',  
3     );
```

Code Source 2 : création d'une liste avec les données du formulaire

Il faut d'abord créer une liste contenant les données que nous souhaitons envoyer au formulaire (Code Source 2). Dans cet exemple, le formulaire ne contient qu'un seul champ, username, ainsi on crée une liste avec pour le champ « username », la valeur « user ».

```
4     $type = new \AppBundle\Form\SecrtaireCreationType();  
5     $form = $this->factory->create($type);
```

Code Source 3 : création du formulaire

Nous instancions ici (Code Source 3) l'objet SecrtaireCreationType, qui contient tout le formulaire. Puis nous créons le formulaire. De plus cela nous permet de vérifier si le formulaire compile correctement.

```
6     $object = new Secrtaire();  
7     $object->setUsername($formData['username']);
```

Code Source 4 : Instanciation de l'objet Secrétaire

Nous instancions ensuite (Code Source 4) l'objet secrétaire, auquel nous lui attribuons les données de notre formulaire, dans le cas présent l'username.

```
8     $form->submit($formData);  
9     $this->assertTrue($form->isSynchronized());
```

Code Source 5 : Soumission du formulaire

Nous allons ensuite soumettre le formulaire (Code Source 5) et tester que le formulaire a bien été envoyé et reçu.

```
10    $this->assertEquals($object, $form->getData());
```

Code Source 6 : Test des données du formulaire

Nous testons ici (Code Source 6) si les données reçues du formulaire contiennent bien les données qui lui ont été envoyées.

```
11     $view = $form->createView();
12     $children = $view->children;
13
14     foreach (array_keys($formData) as $key) {
15         $this->assertArrayHasKey($key, $children);
16     }
```

Code Source 7 : Création de la vue du formulaire

Et pour terminer, nous allons tester (Code Source 7) si la création de la vue du formulaire correspond bien au formulaire que nous voulons créer.

Nous avons ainsi testé la plupart des comportements basiques d'un formulaire.

5

Conclusions

5.1 Rétrospective

Dans notre société actuelle, chaque entreprise doit devenir performante afin de pouvoir gagner des parts de marché, et se démarquer face à la concurrence.

La gestion manuelle de l'entreprise, a certes été la seule gestion connue dans le passé, néanmoins aujourd'hui des technologies un peu plus modernes permettent de gagner du temps. En s'obstinant à gérer le tout de manière manuelle, une entreprise perd de l'argent, du temps et au final des clients.

La gestion de l'entreprise indépendante, moins grande, a certes besoin d'une gestion moins complexe. La gestion manuelle est encore aujourd'hui très utilisée dans ce genre de petits commerces de proximités (serrurerie, cordonnier, petites entreprises du bâtiment...).

Cependant une autoécole gère de nombreux acteurs et créneaux horaires, une digitalisation des processus permettrait de rendre plus efficace toute la gestion administrative.

C'est pour toutes ces raisons, qu'une proposition de logiciel a été faite. L'objectif du travail était de réaliser un projet, de sa conception à son implémentation, avec à terme un prototype permettant de montrer la viabilité d'un tel projet.

Pour le développement du logiciel, nous avons commencé par discuter avec l'entreprise afin de trier les informations et poser le cadre du projet, avec les différentes fonctionnalités que devaient fournir le projet. Il en a été déduit qu'il faut pouvoir faire la gestion du calendrier directement en ligne, pouvoir gérer le calendrier à partir d'une interface graphique simple à utiliser depuis laquelle nous pourrions y insérer les différents cours. La gestion des élèves doit également pouvoir être réalisé en ligne simplement, tout en perdant le moins de temps possible pour la créer et la mettre à jour. Une gestion de la facturation permettrait également de faciliter la gestion et gagner un temps considérable.

Puis nous avons mis au point la conception de ce projet, avec une analyse de la base de données, une conception de celle-ci, puis au final l'implémentation d'un prototype qui regroupe la plupart des fonctionnalités qui ont été décrites et demandées par les entreprises en question.

Le prototype peut être utilisé tel quel ; néanmoins certaines améliorations doivent être effectuées, soit dans le cadre de la sécurité (sauvegarde, etc.), soit au niveau des dernières fonctionnalités qui n'ont pas été implémentées.

L'avantage de ce projet est qu'il peut facilement être réutilisé par d'autres entreprises similaires bien que les fonctionnalités soient propres à celle d'une autoécole. Nous pourrions imaginer une adaptation facile pour un secrétariat d'un cabinet médical avec plusieurs docteurs par exemple. Le calendrier permet, dans la même optique, d'être réutilisé pour beaucoup d'autres applications, on pourrait citer par exemple le pointage horaire pour les employés d'une entreprise.

5.2 Difficultés rencontrées lors du projet

En débutant un tel projet, je savais que j'aurais des difficultés dans la préparation du projet et de la conception. Mais un tel défi était nécessaire pour étendre mes compétences en gestion de projet. En effet, la récolte des informations et le traitement de celles-ci, pour en arriver à quelque chose d'utilisable a pris passablement de temps et d'énergie.

Une fois cette phase terminée, il a fallu apprendre à utiliser un certain nombre de technologies, telles que le Framework Symfony2, Bootstrap ou encore jQuery. C'était une expérience très intéressante et instructive, bien qu'elle ait pris beaucoup de temps.

La dernière grande difficulté rencontrée a été celle de l'écriture du rapport final. En effet, l'organisation des informations a également été une grande étape à franchir.

5.3 Améliorations potentielles

Le prototype n'est par définition qu'une première version avec ses limitations. Il faudrait encore le retravailler afin de le rendre complètement opérationnel. Le projet est réalisé de manière à ce qu'on puisse rajouter de nouvelles fonctionnalités assez facilement. Le modèle MVC permet entre autres à tout nouveau programmeur de s'intégrer et comprendre facilement le projet. Voici une liste non exhaustive des principales fonctionnalités qui n'ont pas été implémentées dans ce prototype :

- Finalisation de la réservation des cours collectifs par les élèves eux-mêmes (paiement en ligne directement) ;
- Désinscription des cours collectifs avec le coût que cela engendre ;
- Paiement directement en ligne des cours non encore payés ;
- Réalisation de factures mensuelles (en PDF par exemple).
- Envoi automatiques des factures

6

Annexes

Titre	S'authentifier	
Description	Un utilisateur arrive sur le système et doit s'authentifier	
Acteurs	Utilisateur non connecté	
Pré-conditions	L'utilisateur n'est pas encore connecté sur le système	
Post-conditions	En cas de succès	L'utilisateur est correctement authentifié
	En cas d'échec	L'utilisateur n'est pas authentifié et ne peut continuer sur le système
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. L'utilisateur se trouve devant la page de login b. L'utilisateur entre un nom d'utilisateur et un mot de passe c. Le système vérifie que le pair nom d'utilisateur et mot de passe existe bien dans le système d. Le système connecte l'utilisateur à l'application e. L'utilisateur est correctement authentifié 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici 	
	Le nom d'utilisateur ou le mot de passe sont invalides	
	<ul style="list-style-type: none"> a. L'utilisateur est averti b. L'utilisateur doit corriger les informations qu'il a entrées c. Le cas d'utilisation continue à l'étape c du scénario principal 	
Besoins non-	Ergonomie	

fonctionnels	L'authentification doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone
	Vérification des erreurs côtés client
	Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page

Tableau 1 : cas d'utilisation « s'authentifier »

Titre	Inscrire un élève sur le système	
Description	Un moniteur inscrit un élève sur le système	
Acteurs	Moniteur	
Pré-conditions	Le moniteur est authentifié sur le système	
Post-conditions	En cas de succès	<ul style="list-style-type: none"> - L'élève est inscrit sur le système - L'élève est averti par email de la création du compte pour lui et reçoit un lien pour activer son compte et créer un mot de passe
	En cas d'échec	L'élève n'est pas inscrit sur le système
Scénario principal aboutissant au succès	<ol style="list-style-type: none"> a. Le moniteur remplit les champs demandés et clique sur le bouton de validation b. Le système vérifie les champs c. L'élève est averti par email de la création du compte, reçoit un lien de validation et de création du mot de passe d. Le moniteur se déconnecte du système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	<ol style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici 	
	Une des informations n'est pas valide	
	<ol style="list-style-type: none"> a. Le moniteur est averti b. Le moniteur est prié de changer les informations non valides c. Le cas d'utilisation continue à l'étape b du scénario principal 	

		Le moniteur veut changer de page ou quitter le navigateur avant de valider
		<ul style="list-style-type: none"> a. Le système demande une confirmation b. Si le moniteur valide, le cas d'utilisation se termine ici c. Sinon le cas d'utilisation continue à l'étape b du scénario principal
Besoins fonctionnels	non-	Ergonomie
		L'inscription doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone
		Vérification des erreurs côtés client
		Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page
		Sécurité
		Les données n'ont pas besoin d'être cryptés

Tableau 2 : cas d'utilisation « inscrire un élève »

Titre	Modifier un élève sur le système	
Description	Un moniteur modifie un élève sur le système	
Acteurs	Moniteur	
Préconditions	<ul style="list-style-type: none"> - Le moniteur est authentifié sur le système - L'élève est déjà inscrit sur le système 	
Post conditions	En cas de succès	<ul style="list-style-type: none"> - Les informations de l'élève sont modifiées sur le système - L'élève est averti par email de la modification
	En cas d'échec	Les informations de l'élève ne sont pas modifiées sur le système
Scénario principal	<ul style="list-style-type: none"> a. Le moniteur remplit les champs demandés et clique sur le bouton de validation 	

aboutissant au succès	<ul style="list-style-type: none"> b. Le système vérifie les champs c. L'élève est avertie par email de la modification de ses informations d. Le moniteur se déconnecte du système
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici
	Une des informations n'est pas valide
	<ul style="list-style-type: none"> a. Le moniteur est averti b. Le moniteur est prié de changer les informations non valides c. Le cas d'utilisation continue à l'étape b du scénario principal
	Le moniteur veut changer de page ou quitter le navigateur avant de valider
	<ul style="list-style-type: none"> a. Le système demande une confirmation b. Si le moniteur valide, le cas d'utilisation se termine ici c. Sinon le cas d'utilisation continue à l'étape b du scénario principal
Besoins non-fonctionnels	Ergonomie
	La modification doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone
	Vérification des erreurs côté client
	Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page
	Sécurité
	Les données n'ont pas besoin d'être cryptés

Tableau 3 : Cas d'utilisation « modifier un élève »

Titre	Supprimer un élève sur le système
Description	Un moniteur supprime un élève sur le système

Acteurs	Moniteur	
Préconditions	<ul style="list-style-type: none"> - Le moniteur est authentifié sur le système - L'élève est déjà inscrit sur le système 	
Post conditions	En cas de succès	<ul style="list-style-type: none"> - Les informations de l'élève sont supprimées sur le système - L'élève est averti par email de la suppression
	En cas d'échec	Les informations de l'élève ne sont pas supprimées sur le système
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le moniteur clique sur le bouton de suppression b. Un avertissement doit être validé c. L'élève est avertie par email de la suppression de son compte d. Le moniteur se déconnecte du système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici 	
Besoins non-fonctionnels	Ergonomie	
	La suppression doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons doivent apparaître lors de l'affichage sur tablette ou smartphone	

Tableau 4 : Cas d'utilisation « supprimer un élève »

Titre	Désactiver un élève sur le système	
Description	Un moniteur désactive un élève sur le système	
Acteurs	Moniteur	
Préconditions	<ul style="list-style-type: none"> - Le moniteur est authentifié sur le système - L'élève est déjà inscrit sur le système 	
Post conditions	En cas de succès	<ul style="list-style-type: none"> - L'élève est désactivé sur le système - L'élève est averti par email de la désactivation

	En cas d'échec	Les informations de l'élève ne sont pas désactivées sur le système
Scénario principal aboutissant au succès		<ul style="list-style-type: none"> a. Le moniteur clique sur le bouton de désactivation b. Un avertissement doit être validé c. L'élève est averti par email de la désactivation de son compte d. Le moniteur se déconnecte du système
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
		<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici
Besoins non-fonctionnels	Ergonomie	
		La désactivation doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons doivent apparaître lors de l'affichage sur tablette ou smartphone

Tableau 5 : Cas d'utilisation « désactiver un élève »

Titre	Inscrire un de ses élèves à un cours collectif	
Description	Un moniteur inscrit un de ses élèves à un cours auquel le moniteur a été annoncé par le secrétaire	
Acteurs	Moniteur	
Pré-conditions	<ul style="list-style-type: none"> a. Le moniteur est authentifié. b. Le cours est déjà dans le système c. L'élève est présent dans le système d. Le moniteur a été annoncé par le secrétaire comme moniteur du cours 	
Post-conditions	En cas de succès	<ul style="list-style-type: none"> - L'élève est inscrit au cours - L'élève est averti par email de son inscription au cours
	En cas d'échec	L'élève n'est pas inscrit sur le système
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le moniteur choisit le cours désiré parmi la liste des cours qui lui sont proposés (uniquement des cours ou il apparaît comme moniteur) b. Le moniteur choisit le ou les élèves à inscrire parmi une liste 	

	de ses élèves c. Le ou les élèves sont avertis par email d. Le moniteur se déconnecte du système
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie
	a. Le système affiche une erreur b. Le cas d'utilisation se termine ici
	Une des informations n'est pas valide
	a. Le moniteur est averti b. Le moniteur est prié de changer les informations non valides c. Le cas d'utilisation continue à l'étape b du scénario principal
Besoins non-fonctionnels	Ergonomie
	L'inscription doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone

Tableau 6 : Cas d'utilisation « inscrire un de ses élèves à un cours collectif »

Titre	Supprimer l'inscription d'un de ses élèves à un cours collectif	
Description	Un moniteur supprime l'inscription d'un de ses élèves à un cours	
Acteurs	Moniteur	
Pré-conditions	a. Le moniteur est authentifié. b. Le cours est déjà dans le système c. L'élève est présent dans le système et inscrit au cours d. Le moniteur a été annoncé par le secrétaire comme moniteur du cours	
Post-conditions	En cas de succès	- L'élève est supprimé du cours - L'élève est averti par email de la suppression du cours - Si l'élève a payé, le solde est ajouté dans son compte
	En cas d'échec	L'élève n'est pas inscrit sur le système
Scénario principal aboutissant au succès	a. Le moniteur choisit le cours désiré parmi la liste des cours qui lui sont proposés (uniquement des cours ou il apparaît comme moniteur) b. Le moniteur choisit le ou les élèves à supprimer parmi une liste de ses élèves	

	<ul style="list-style-type: none"> c. Le ou les élèves sont avertis par email d. Le moniteur se déconnecte du système
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici
	Le cours était payé et le délai de désinscription n'est pas passé
	La somme qu'il a payé ôté du prix de désinscription du cours lui est ajouté dans son solde restant sur son compte, il pourra avec cela payer un autre cours.
	Le cours était payé et le délai de désinscription est passé
	La somme qu'il a payé ôté du prix de désinscription tardif du cours lui est ajouté dans son solde restant sur son compte, il pourra avec cela payer un autre cours.
Besoins non-fonctionnels	Ergonomie
	La suppression doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone

Tableau 7 : Cas d'utilisation « supprimer un de ses élèves à un cours collectif »

Titre	Voir le détail des cours collectif auxquels le moniteur participe	
Description	Un moniteur désire voir la liste des élèves d'un cours collectif ou il est moniteur	
Acteurs	Moniteur	
Pré-conditions	<ul style="list-style-type: none"> a. Le moniteur est authentifié. b. Le cours est déjà dans le système c. Le moniteur a été annoncé par le secrétaire comme moniteur du cours 	
Post-conditions	En cas de succès	La liste des élèves participants est affichée

	En cas d'échec	La liste n'est pas affichée
Scénario principal aboutissant au succès		<ul style="list-style-type: none"> a. Le moniteur choisit un cours collectif parmi la liste (uniquement des cours auxquels il est annoncé comme moniteur) b. Le système affiche la liste des élèves participants à ce cours (uniquement nom et prénom) avec indication de quel moniteur il est lié et y ajoute l'information si le cours a été payé ou non. c. A côté de chaque élève une case permet de dire si le cours a été payé ou non. Un simple clic dessus permet de changer le statut du paiement. d. Le moniteur se déconnecte du système
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
		<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici
Besoins non-fonctionnels	Ergonomie	
		La liste doit s'adapter à la taille de l'écran.

Tableau 8 : Cas d'utilisation « voir détail cours collectif »

Titre	Voir la liste des cours individuels	
Description	Un moniteur désire voir la liste de ses cours individuels	
Acteurs	Moniteur	
Pré-conditions	<ul style="list-style-type: none"> a. Le moniteur est authentifié. b. Le cours est déjà dans le système 	
Post-conditions	En cas de succès	La liste des cours des affiché
	En cas d'échec	La liste n'est pas affichée
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le moniteur choisit un élève parmi une liste de tous ses élèves actifs b. La liste des cours individuel de l'élève choisi qui ont été validés est affichée c. A côté de chaque cours une case permet de dire si le cours a été payé ou non. Un clic dessus permet de changer le statut du 	

		<p>payement.</p> <p>d. Le moniteur se déconnecte du système</p>
Déroulement alternatif		La connexion avec le serveur ne peut pas être établie
		<p>a. Le système affiche une erreur</p> <p>b. Le cas d'utilisation se termine ici</p>
Besoins non-fonctionnels		Ergonomie
		La liste doit s'adapter à la taille de l'écran

Tableau 9 : Cas d'utilisation « voir liste cours individuels »

Titre	Créer un type de cours	
Description	Un moniteur crée un type de cours qui lui est propre	
Acteurs	Moniteur	
Pré-conditions	Le moniteur est authentifié sur le système	
Post-conditions	En cas de succès	Un type de cours est créé
	En cas d'échec	Le type de cours n'est pas créé
Scénario principal aboutissant au succès	<p>a. Le moniteur remplit les champs demandés et clique sur le bouton de validation</p> <p>b. Le système vérifie les champs</p> <p>c. Le type de cours est créé</p> <p>d. Le moniteur se déconnecte du système</p>	
Déroulement alternatif		La connexion avec le serveur ne peut pas être établie
		<p>a. Le système affiche une erreur</p> <p>b. Le cas d'utilisation se termine ici</p>
		Une des informations n'est pas valide
		<p>a. Le moniteur est averti</p> <p>b. Le moniteur est prié de changer les informations non valides</p> <p>c. Le cas d'utilisation continue à l'étape b du scénario principal</p>
Besoins non-	Ergonomie	

fonctionnels	La création du type de cours doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l’affichage sur tablette ou smartphone
	Vérification des erreurs côtés client
	Les erreurs sont aussi vérifiées côté client, afin que l’utilisateur en soit rapidement averti sans actualisation de la page
	Sécurité
	Les données n’ont pas besoin d’être cryptés

Tableau 10 : Cas d’utilisation « créer un type de cours »

Titre	Modifier un type de cours	
Description	Un moniteur modifie un type de cours qu’il a créé précédemment	
Acteurs	Moniteur	
Pré-conditions	<ul style="list-style-type: none"> - Le moniteur est authentifié sur le système - Le type de cours existe dans la base de données 	
Post-conditions	En cas de succès	Un type de cours est modifié
	En cas d’échec	Le type de cours n’est pas modifié
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le moniteur remplit les champs demandés et clique sur le bouton de validation b. Le système vérifie les champs c. Le type de cours est modifié d. Le moniteur se déconnecte du système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d’utilisation se termine ici 	
	Une des informations n’est pas valide	
	<ul style="list-style-type: none"> a. Le moniteur est averti 	

		<ul style="list-style-type: none"> b. Le moniteur est prié de changer les informations non valides c. Le cas d'utilisation continue à l'étape b du scénario principal
Besoins fonctionnels non-	Ergonomie	
		La modification du type de cours doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone
	Vérification des erreurs côtés client	
		Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page
	Sécurité	
		Les données n'ont pas besoin d'être cryptés

Tableau 11 : Cas d'utilisation « modifier un type de cours »

Titre	Supprimer un type de cours	
Description	Un moniteur supprime un type de cours	
Acteurs	Moniteur	
Pré-conditions	<ul style="list-style-type: none"> - Le moniteur est authentifié sur le système - Le type de cours existe dans la base de données 	
Post-conditions	En cas de succès	Un type de cours est supprimé
	En cas d'échec	Le type de cours n'est pas supprimé
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le moniteur clique sur le bouton de suppression b. Le système affiche un avertissement et demande une confirmation, les données prédéfinies ne seront plus inscrites et il faudra les redonner lors de chaque validation de cours qui avait ce type de permis et qui n'aurait pas encore été validé c. Le type de cours est supprimé d. Le moniteur se déconnecte du système 	

Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	a. Le système affiche une erreur b. Le cas d'utilisation se termine ici	
	Une des informations n'est pas valide	
	a. Le moniteur est averti b. Le moniteur est prié de changer les informations non valides c. Le cas d'utilisation continue à l'étape b du scénario principal	
Besoins fonctionnels non-	Ergonomie	
	La suppression du type de cours doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone	
	Vérification des erreurs côtés client	
	Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page	
	Sécurité	
	Les données n'ont pas besoin d'être cryptés	

Tableau 12 : Cas d'utilisation « supprimer un type de cours »

Titre	Ajouter un évènement	
Description	Un moniteur crée un événement sur le calendrier. Cela peut être différentes choses selon le type de cours choisit. Pour la plupart des fois il s'agira d'un cours individuel. Il est aussi possible de créer un événement privé par exemple.	
Acteurs	Moniteur	
Pré-conditions	Le moniteur est authentifié	
Post-conditions	En cas de	L'évènement est créé

	succès	
	En cas d'échec	L'événement n'est pas créé
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le moniteur choisit une plage horaire sur le calendrier b. Une fenêtre modale s'ouvre et demande d'entrer les informations c. Si un élève est ajouté, l'élève reçoit un email lui confirmant le cours d. Le moniteur se déconnecte du système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici 	
Besoins non-fonctionnels	Ergonomie	
	La création de l'événement doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone. La fenêtre modale est bien adaptée à la taille de l'écran.	
	Vérification des erreurs côté client	
	Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page	

Tableau 13 : Cas d'utilisation « Ajouter un événement au calendrier »

Titre	Modifier un événement	
Description	Un moniteur modifie un événement du calendrier	
Acteurs	Moniteur	
Pré-conditions	<ul style="list-style-type: none"> a. Le moniteur est authentifié b. Le cours est déjà créé 	
Post-conditions	En cas de succès	L'événement est modifié

	En cas d'échec	L'événement n'est pas modifié
Scénario principal aboutissant au succès		<ul style="list-style-type: none"> a. Le moniteur choisit un événement sur le calendrier b. Une fenêtre modale s'ouvre et demande de modifier les informations affichées c. Si un élève est concerné, l'élève reçoit un email lui confirmant la modification d. Le moniteur se déconnecte du système
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
		<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici
Besoins non-fonctionnels	Ergonomie	
	La modification de l'événement doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone. La fenêtre modale est bien adaptée à la taille de l'écran.	
	Vérification des erreurs côtés client	
	Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page	

Tableau 14 : Cas d'utilisation « Modifier un événement au calendrier »

Titre	Supprimer un évènement	
Description	Un moniteur supprime un événement du calendrier	
Acteurs	Moniteur	
Pré-conditions	<ul style="list-style-type: none"> a. Le moniteur est authentifié b. Le cours est déjà créé 	
Post-conditions	En cas de succès	L'événement est supprimé
	En cas d'échec	L'événement n'est pas supprimé

Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le moniteur choisit un événement sur le calendrier b. Une fenêtre modale s'ouvre et le moniteur clique sur le bouton de suppression c. Si un élève est concerné, l'élève reçoit un email lui confirmant la suppression d. Le moniteur se déconnecte du système
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici
Besoins non-fonctionnels	Ergonomie
	La suppression de l'événement doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone. La fenêtre modale est bien adaptée à la taille de l'écran.

Tableau 15 : Cas d'utilisation « Supprimer un événement au calendrier »

Titre	Voir le calendrier complet	
Description	Le moniteur affiche le calendrier en entier contenant les réservations de tous ses élèves et celles qu'il a fait lui-même	
Acteurs	Moniteur	
Pré-conditions	Le moniteur est authentifié	
Post-conditions	En cas de succès	Le calendrier est affiché
	En cas d'échec	Le calendrier n'est pas affiché
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le moniteur clique sur le bouton b. Le calendrier est affiché c. Le moniteur se déconnecte du système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	a. Le système affiche une erreur	

	b. Le cas d'utilisation se termine ici
Besoins fonctionnels non-	Ergonomie
	Le calendrier doit s'afficher en premier lieu avec une vue sur la semaine courante en entier. Il doit s'adapter à la taille de l'écran.

Tableau 16 : Cas d'utilisation « Voir le calendrier complet »

Titre	Valider un cours	
Description	Un moniteur valide un cours	
Acteurs	Moniteur	
Pré-conditions	<ul style="list-style-type: none"> - Le moniteur est authentifié sur le système - Le cours existe dans le système - La date du cours est passée 	
Post-conditions	En cas de succès	Le cours est validé
	En cas d'échec	Le cours n'est pas validé
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Une liste des cours non validés et déjà passé est affichée à l'écran b. Le moniteur clique sur un cours c. Une fenêtre modale s'ouvre avec les informations pré remplis grâce au type de cours. Le moniteur peut modifier les informations et cliquer sur valider d. Le moniteur se déconnecte du système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici 	
	Une des informations n'est pas valide	
	<ul style="list-style-type: none"> a. Le moniteur est averti b. Le moniteur est prié de changer les informations non valides c. Le cas d'utilisation continue à l'étape c du scénario principal 	
Besoins non-	Ergonomie	

fonctionnels	La validation du cours doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l’affichage sur tablette ou smartphone. La fenêtre modale est bien adaptée à la taille de l’écran.
	Vérification des erreurs côtés client
	Les erreurs sont aussi vérifiées côté client, afin que l’utilisateur en soit rapidement averti sans actualisation de la page

Tableau 17 : Cas d’utilisation « Valider un cours »

ELEVES

Titre	S’inscrire	
Description	Un élève s’inscrit directement depuis le site d’un moniteur	
Acteurs	Moniteur	
Pré-conditions	Aucune	
Post-conditions	En cas de succès	Un nouvel élève est créé
	En cas d’échec	Le nouvel élève n’est pas créé
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. L’utilisateur remplit les champs du formulaire b. Le système vérifie les champs c. L’élève reçoit un email avec un lien de validation d. L’élève clique sur le lien, son compte est validé et il est connecté sur le système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d’utilisation se termine ici 	
	Une des informations n’est pas valide	
	<ul style="list-style-type: none"> a. L’utilisateur est averti b. L’utilisateur est prié de changer les informations non valides 	

		c. Le cas d'utilisation continue à l'étape b du scénario principal
		L'élève ne clique pas sur le lien avant la date d'expiration du lien
		a. Le système averti l'élève qui reçoit un nouveau lien par email b. Le cas d'utilisation continue à l'étape d du scénario principal
Besoins fonctionnels non-	non-	Ergonomie
		L'inscription doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone.
		Vérification des erreurs côtés client
		Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page

Tableau 18 : Cas d'utilisation « S'inscrire »

Titre	Créer un mot de passe et activer le compte	
Description	Un moniteur a inscrit un élève et l'élève a reçu un lien pour créer un nom d'utilisateur et un mot de passe. Ce lien va également activer le compte	
Acteurs	Elève	
Pré-conditions	<ul style="list-style-type: none"> - L'élève a reçu le lien - L'élève a cliqué sur le lien 	
Post-conditions	En cas de succès	Le compte est activé
	En cas d'échec	Le compte n'est pas activé
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. L'utilisateur choisit un nom d'utilisateur et un mot de passe b. Le système vérifie les champs c. L'élève est activé d. L'élève est connecté sur le système 	
Déroulement	La connexion avec le serveur ne peut pas être établie	

alternatif	a. Le système affiche une erreur b. Le cas d'utilisation se termine ici
	Une des informations n'est pas valide
	a. L'utilisateur est averti b. L'utilisateur est prié de changer les informations non valides c. Le cas d'utilisation continue à l'étape b du scénario principal
	L'élève ne clique pas sur le lien avant la date d'expiration du lien
	a. Le système averti l'élève qui reçoit un nouveau lien par email b. Le cas d'utilisation continue à l'étape d du scénario principal
Besoins fonctionnels non-	Ergonomie
	La création du mot de passe doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone.
	Vérification des erreurs côtés client
	Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page

Tableau 19 : Cas d'utilisation « Créer un mot de passe et activer son compte »

Titre	Modifier ses informations personnelles	
Description	Un élève modifie ses informations personnelles	
Acteurs	Elève	
Pré-conditions	L'élève est authentifié	
Post-conditions	En cas de succès	Les informations sont modifiées
	En cas d'échec	Les informations ne sont pas modifiées
Scénario principal aboutissant au succès	a. L'élève modifie les champs b. Le système vérifie les champs c. L'élève se déconnecte du système	

Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	c. Le système affiche une erreur d. Le cas d'utilisation se termine ici	
	Une des informations n'est pas valide	
	d. L'utilisateur est averti e. L'utilisateur est prié de changer les informations non valides f. Le cas d'utilisation continue à l'étape b du scénario principal	
Besoins fonctionnels non-	Ergonomie	
	La modification des informations doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone.	
	Vérification des erreurs côté client	
	Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page	

Tableau 20 : Cas d'utilisation « Modifier ses informations personnelles »

Titre	S'inscrire à un cours collectif	
Description	Un élève souhaite s'inscrire à un cours collectif	
Acteurs	Elève	
Pré-conditions	L'élève est authentifié	
Post-conditions	En cas de succès	L'élève est inscrit au cours
	En cas d'échec	L'élève n'est pas inscrit au cours
Scénario principal aboutissant au succès	a. L'élève choisit un cours parmi les cours collectifs que son moniteur propose b. L'élève confirme qu'il veut réserver ce cours et le stock restant est déduit de la base de données c. L'élève clique sur le bouton payer et se voit rediriger sur la page de la banque	

	<ul style="list-style-type: none"> d. Le paiement est confirmé par la banque e. L'inscription de l'élève est confirmée et l'élève reçoit un email de confirmation f. L'élève se déconnecte du système
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici
	La banque ne confirme pas le paiement
	<ul style="list-style-type: none"> a. L'inscription est annulée b. Le stock restant est augmentée c. L'élève est averti et doit recommencer la procédure
	La banque ne confirme pas le paiement dans un délai de 2 heures
	<ul style="list-style-type: none"> a. L'inscription est annulée b. Le stock restant est augmentée c. L'élève est averti et doit recommencer la procédure d. Le cas d'utilisation se termine ici
Besoins non-fonctionnels	Ergonomie
	L'inscription doit pouvoir se faire facilement depuis une tablette ou un smartphone

Tableau 21 : Cas d'utilisation « S'inscrire à un cours collectif »

Titre	Modifier son inscription à un cours collectif	
Description	Un élève souhaite modifier son inscription à un cours collectif	
Acteurs	Elève	
Pré-conditions	<ul style="list-style-type: none"> - L'élève est authentifié - L'élève est inscrit au cours collectif 	
Post-conditions	En cas de succès	L'élève est inscrit au cours
	En cas d'échec	L'élève n'est pas inscrit au cours
Scénario principal	<ul style="list-style-type: none"> a. L'élève choisit un cours parmi les cours collectifs ou il est inscrit 	

aboutissant au succès	<ul style="list-style-type: none"> b. L'élève confirme qu'il veut annuler son inscription au cours. Le prix de désinscription ou le prix de désinscription tardif est affiché. L'élève est averti que l'argent ne lui sera pas remboursé, mais qui lui permettra de réserver un autre cours. c. Le stock est mis à jour d. Son compte est mis à jour e. La désinscription est effective f. L'élève se déconnecte du système
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici
Besoins non-fonctionnels	Ergonomie
	La modification doit pouvoir se faire facilement depuis une tablette ou un smartphone

Tableau 22 : Cas d'utilisation « Modifier son inscription à un cours collectif »

Titre	Afficher tous les cours validés	
Description	Un élève veut afficher tous les cours qu'il a faits et que le moniteur a validés	
Acteurs	Elève	
Pré-conditions	L'élève est authentifié	
Post-conditions	En cas de succès	La liste est affichée
	En cas d'échec	La liste n'est pas affichée
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. L'élève clique pour voir la liste des cours b. La liste est affichée et il est ajouté un bouton payé à côté des cours qu'il n'a pas payé c. L'élève se déconnecte du système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	a. Le système affiche une erreur	

		b. Le cas d'utilisation se termine ici
Besoins fonctionnels	non-	Ergonomie
		Le tableau doit pouvoir se voir facilement depuis une tablette ou un smartphone. La liste s'adapte à la taille de l'écran.

Tableau 23 : Cas d'utilisation « Voir les cours validés »

Titre	Voir le calendrier avec ses réservations	
Description	Un élève veut voir le calendrier ou sont affichés tous les cours qu'il a réservé	
Acteurs	Elève	
Pré-conditions	- L'élève est authentifié	
Post-conditions	En cas de succès	Le calendrier est affiché
	En cas d'échec	Le calendrier n'est pas affiché
Scénario principal aboutissant au succès	a. L'élève clique pour voir le calendrier b. Le calendrier est affiché avec tous les cours au nom de l'élève c. L'élève se déconnecte du système	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	a. Le système affiche une erreur b. Le cas d'utilisation se termine ici	
Besoins fonctionnels	non-	Ergonomie
		Le calendrier doit pouvoir se voir facilement depuis une tablette ou un smartphone. Le calendrier s'adapte à la taille de l'écran.

Tableau 24 : Cas d'utilisation « Voir le calendrier »

SECRETARE

Titre	Créer un cours collectif	
Description	Un secrétaire décide de créer un cours collectif	
Acteurs	Secrétaire	
Préconditions	- Le secrétaire est authentifiée sur le système	
Post conditions	En cas de succès	Le cours est créé
	En cas d'échec	Le cours n'est pas créé
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le secrétaire remplit les champs demandés et clique sur le bouton de validation b. Le système vérifie les champs c. Les moniteurs concernés par le cours sont averti par email de la création de celui-ci d. Le secrétaire se déconnecte du système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici 	
	Une des informations n'est pas valide	
	<ul style="list-style-type: none"> a. Le secrétaire est avertie b. Le secrétaire est priée de changer les informations non valides c. Le cas d'utilisation continue à l'étape b du scénario principal 	
	Le secrétaire veut changer de page ou quitter le navigateur avant de valider	
	<ul style="list-style-type: none"> a. Le système demande une confirmation b. Si le secrétaire valide, le cas d'utilisation se termine ici c. Sinon le cas d'utilisation continue à l'étape b du scénario principal 	
Besoins non-fonctionnels	Ergonomie	
	La création doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone	

	Vérification des erreurs côtés client
	Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page
	Sécurité
	Les données n'ont pas besoin d'être cryptés

Tableau 25 : Cas d'utilisation « Créer un cours collectif »

Titre	Modifier un cours collectif	
Description	Un secrétaire décide de modifier un cours collectif	
Acteurs	Secrétaire	
Préconditions	<ul style="list-style-type: none"> - Le secrétaire est authentifiée sur le système - Aucun élève n'est inscrit - Le cours existe 	
Post conditions	En cas de succès	Le cours est modifié
	En cas d'échec	Le cours n'est pas modifié
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le secrétaire modifie les champs demandés et clique sur le bouton de validation b. Le système vérifie les champs c. Les moniteurs concernés par le cours sont averti par email de la modification de celui-ci d. Le secrétaire se déconnecte du système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici 	
	Une des informations n'est pas valide	
	<ul style="list-style-type: none"> a. Le secrétaire est avertie b. Le secrétaire est priée de changer les informations non valides c. Le cas d'utilisation continue à l'étape b du scénario principal 	

		Le secrétaire veut changer de page ou quitter le navigateur avant de valider
		<ul style="list-style-type: none"> a. Le système demande une confirmation b. Si le secrétaire valide, le cas d'utilisation se termine ici c. Sinon le cas d'utilisation continue à l'étape b du scénario principal
Besoins fonctionnels non-		Ergonomie
		La modification doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone
		Vérification des erreurs côtés client
		Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page
		Sécurité
		Les données n'ont pas besoin d'être cryptés

Tableau 26 : Cas d'utilisation « Modifier un cours collectif »

Titre	Supprimer un cours collectif	
Description	Un secrétaire décide de créer un cours collectif	
Acteurs	Secrétaire	
Préconditions	<ul style="list-style-type: none"> - Le secrétaire est authentifiée sur le système - Aucun élève n'est inscrit - Le cours existe 	
Post conditions	En cas de succès	Le cours est supprimé
	En cas d'échec	Le cours n'est pas supprimé
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le secrétaire clique sur le bouton de suppression b. Le système avertie le secrétaire que cette action sera définitive c. Les moniteurs concernés par le cours sont averti par email de 	

	la suppression de celui-ci d. Le secrétaire se déconnecte du système
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie
	a. Le système affiche une erreur b. Le cas d'utilisation se termine ici
Besoins non-fonctionnels	Ergonomie
	La suppression doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons doivent apparaître lors de l'affichage sur tablette ou smartphone

Tableau 27 : Cas d'utilisation « Supprimer un cours collectif »

Titre	Inscrire un moniteur	
Description	Un secrétaire inscrit un nouveau moniteur sur le système	
Acteurs	Secrétaire	
Préconditions	Le secrétaire est authentifiée sur le système	
Post conditions	En cas de succès	Le nouveau moniteur est inscrit
	En cas d'échec	Le moniteur n'est pas inscrit
Scénario principal aboutissant au succès	a. Le secrétaire remplit les champs demandés et clique sur le bouton de validation b. Le système vérifie les champs c. Le moniteur est averti par email de la création de son compte d. Le secrétaire se déconnecte du système	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	a. Le système affiche une erreur b. Le cas d'utilisation se termine ici	
	Une des informations n'est pas valide	
	a. Le secrétaire est avertie b. Le secrétaire est priée de changer les informations non valides	

		c. Le cas d'utilisation continue à l'étape b du scénario principal
		Le secrétaire veut changer de page ou quitter le navigateur avant de valider
		d. Le système demande une confirmation e. Si le secrétaire valide, le cas d'utilisation se termine ici f. Sinon le cas d'utilisation continue à l'étape b du scénario principal
Besoins fonctionnels	non-	Ergonomie
		L'inscription doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone
		Vérification des erreurs côtés client
		Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page
		Sécurité
		Les données n'ont pas besoin d'être cryptés

Tableau 28 : Cas d'utilisation « Inscrire un moniteur »

Titre	Modifier un moniteur	
Description	Un secrétaire modifie un moniteur sur le système	
Acteurs	Secrétaire	
Préconditions	<ul style="list-style-type: none"> - Le secrétaire est authentifiée sur le système - Le moniteur existe sur le système 	
Post conditions	En cas de succès	Le moniteur est modifié
	En cas d'échec	Le moniteur n'est pas modifié
Scénario principal	<ul style="list-style-type: none"> a. Le secrétaire modifie les champs demandés et clique sur le bouton de validation b. Le système vérifie les champs 	

aboutissant au succès	<ul style="list-style-type: none"> c. Le moniteur est averti par email de la modification de son compte d. Le secrétaire se déconnecte du système
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici
	Une des informations n'est pas valide
	<ul style="list-style-type: none"> a. Le secrétaire est avertie b. Le secrétaire est priée de changer les informations non valides c. Le cas d'utilisation continue à l'étape b du scénario principal
	Le secrétaire veut changer de page ou quitter le navigateur avant de valider
	<ul style="list-style-type: none"> a. Le système demande une confirmation b. Si le secrétaire valide, le cas d'utilisation se termine ici c. Sinon le cas d'utilisation continue à l'étape b du scénario principal
Besoins fonctionnels non-	Ergonomie
	La modification doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone
	Vérification des erreurs côtés client
	Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page
	Sécurité
	Les données n'ont pas besoin d'être cryptés

Tableau 29 : Cas d'utilisation « modifier un moniteur »

Titre	Supprimer un moniteur
Description	Un secrétaire supprime un moniteur sur le système
Acteurs	Secrétaire

Préconditions	- Le secrétaire est authentifiée sur le système	
Post conditions	En cas de succès	Le moniteur est supprimé
	En cas d'échec	Le moniteur n'est pas supprimé
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le secrétaire clique sur le bouton de suppression b. Le système avertit le secrétaire que cet action sera définitive et que tous les élèves du moniteur seront également supprimés. c. Le moniteur est averti par email de la suppression de son compte d. Le secrétaire se déconnecte du système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici 	
	Un des élèves n'a pas payé	
	<ul style="list-style-type: none"> a. Le secrétaire est avertie b. Le secrétaire doit confirmer la suppression malgré que tous les élèves n'aient pas payé c. Si le secrétaire confirme, le cas d'utilisation continue à l'étape c du scénario principal d. Sinon le cas d'utilisation se termine ici 	
Besoins non-fonctionnels	Ergonomie	
	La suppression doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons doivent apparaître lors de l'affichage sur tablette ou smartphone	

Tableau 30 : Cas d'utilisation « supprimer un moniteur »

Titre	Inscrire un secrétaire
Description	Un secrétaire inscrit une nouvelle secrétaire sur le système
Acteurs	Secrétaire
Préconditions	Le secrétaire est authentifiée sur le système

Post conditions	En cas de succès	La nouvelle secrétaire est inscrite
	En cas d'échec	Le secrétaire n'est pas inscrite
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le secrétaire remplit les champs demandés et clique sur le bouton de validation b. Le système vérifie les champs c. La nouvelle secrétaire est avertie par email de la création de son compte d. Le secrétaire se déconnecte du système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici 	
	Une des informations n'est pas valide	
	<ul style="list-style-type: none"> a. Le secrétaire est avertie b. Le secrétaire est priée de changer les informations non valides c. Le cas d'utilisation continue à l'étape b du scénario principal 	
	Le secrétaire veut changer de page ou quitter le navigateur avant de valider	
	<ul style="list-style-type: none"> g. Le système demande une confirmation h. Si le secrétaire valide, le cas d'utilisation se termine ici i. Sinon le cas d'utilisation continue à l'étape b du scénario principal 	
Besoins non-fonctionnels	Ergonomie	
	L'inscription doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors de l'affichage sur tablette ou smartphone	
	Vérification des erreurs côté client	
	Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page	
	Sécurité	
Les données n'ont pas besoin d'être cryptés		

Tableau 31 : Cas d'utilisation « inscrire un secrétaire »

Titre	Modifier un secrétaire	
Description	Un secrétaire modifie une autre secrétaire sur le système	
Acteurs	Secrétaire	
Préconditions	<ul style="list-style-type: none"> - Le secrétaire est authentifiée sur le système - Le secrétaire existe sur le système 	
Post conditions	En cas de succès	Le secrétaire est modifiée
	En cas d'échec	Le secrétaire n'est pas modifiée
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le secrétaire modifie les champs demandés et clique sur le bouton de validation b. Le système vérifie les champs c. Le secrétaire est avertie par email de la modification de son compte d. Le secrétaire se déconnecte du système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	<ul style="list-style-type: none"> a. Le système affiche une erreur b. Le cas d'utilisation se termine ici 	
	Une des informations n'est pas valide	
	<ul style="list-style-type: none"> a. Le secrétaire est avertie b. Le secrétaire est priée de changer les informations non valides c. Le cas d'utilisation continue à l'étape b du scénario principal 	
	Le secrétaire veut changer de page ou quitter le navigateur avant de valider	
	<ul style="list-style-type: none"> a. Le système demande une confirmation b. Si le secrétaire valide, le cas d'utilisation se termine ici c. Sinon le cas d'utilisation continue à l'étape b du scénario principal 	
Besoins non-fonctionnels	Ergonomie	
	La modification doit pouvoir se faire facilement depuis une tablette ou un smartphone. De grands boutons et champs doivent apparaître lors	

	de l'affichage sur tablette ou smartphone
	Vérification des erreurs côtés client
	Les erreurs sont aussi vérifiées côté client, afin que l'utilisateur en soit rapidement averti sans actualisation de la page
	Sécurité
	Les données n'ont pas besoin d'être cryptés

Tableau 32 : Cas d'utilisation « modifier un secrétaire »

Titre	Supprimer un secrétaire	
Description	Un secrétaire supprime une autre secrétaire sur le système	
Acteurs	Secrétaire	
Préconditions	- Le secrétaire est authentifiée sur le système	
Post conditions	En cas de succès	Le secrétaire est supprimée
	En cas d'échec	Le secrétaire n'est pas supprimée
Scénario principal aboutissant au succès	<ul style="list-style-type: none"> a. Le secrétaire clique sur le bouton de suppression b. Le système avertit le secrétaire que cette action sera définitive. c. Le secrétaire est avertie par email de la suppression de son compte d. Le secrétaire se déconnecte du système 	
Déroulement alternatif	La connexion avec le serveur ne peut pas être établie	
	<ul style="list-style-type: none"> c. Le système affiche une erreur d. Le cas d'utilisation se termine ici 	
Besoins non-fonctionnels	Ergonomie	
	La suppression doit pouvoir se faire facilement depuis une tablette ou	

	un smartphone. De grands boutons doivent apparaître lors de l'affichage sur tablette ou smartphone
--	--

Tableau 33 : Cas d'utilisation « supprimer un secrétaire »

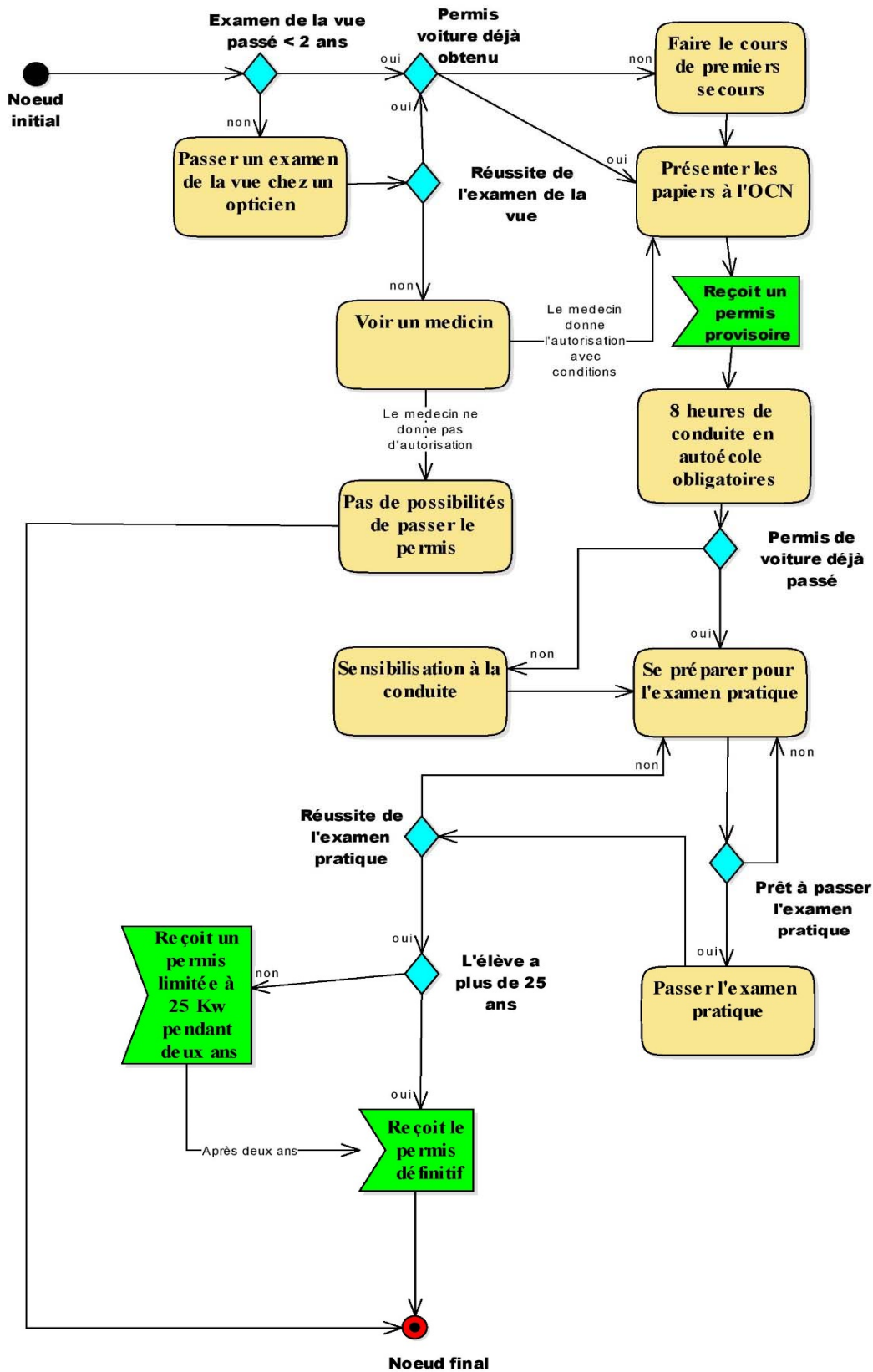


Figure 48 : Diagramme d'activités « permis A »

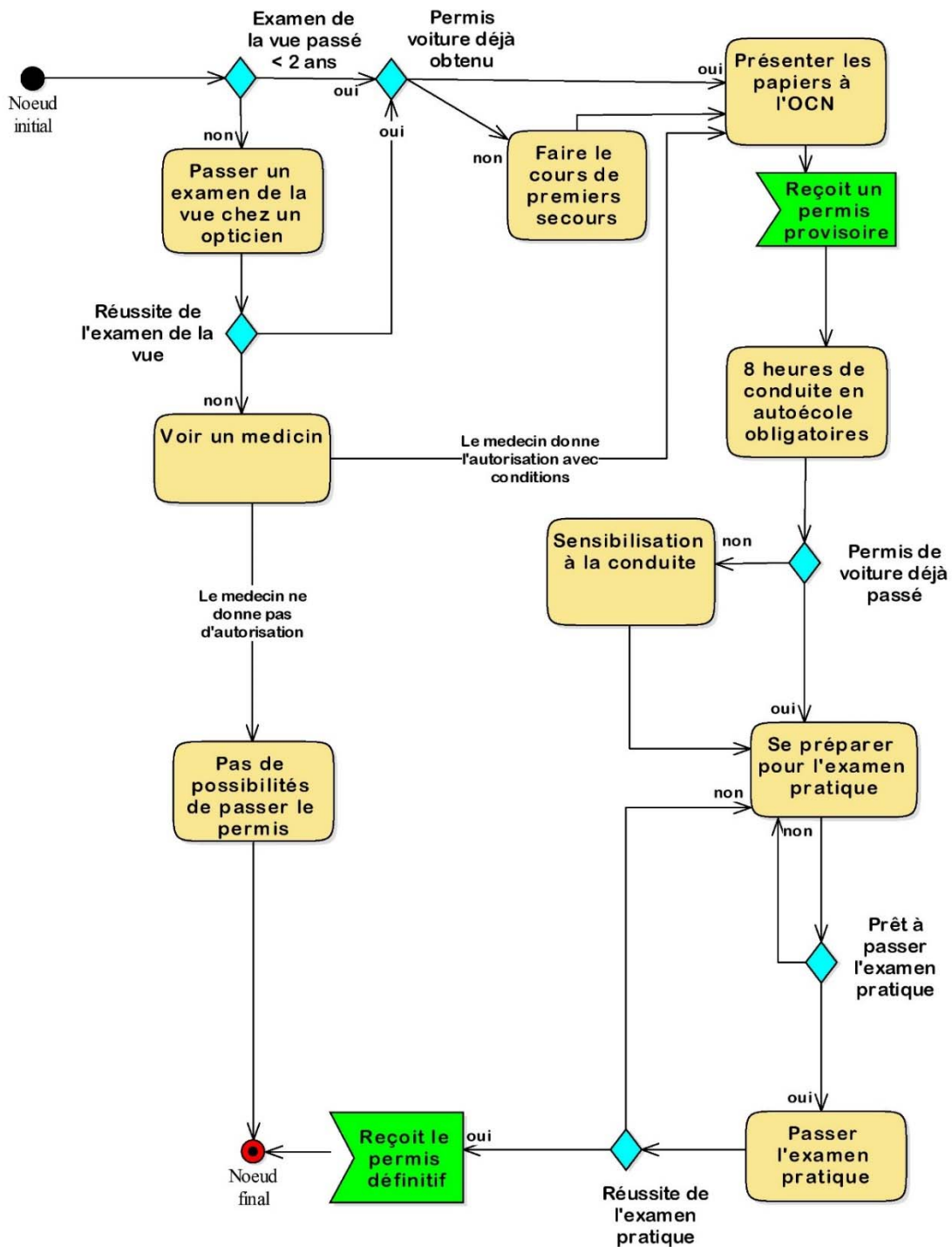


Figure 49 : Diagramme d'activités "permis A1"

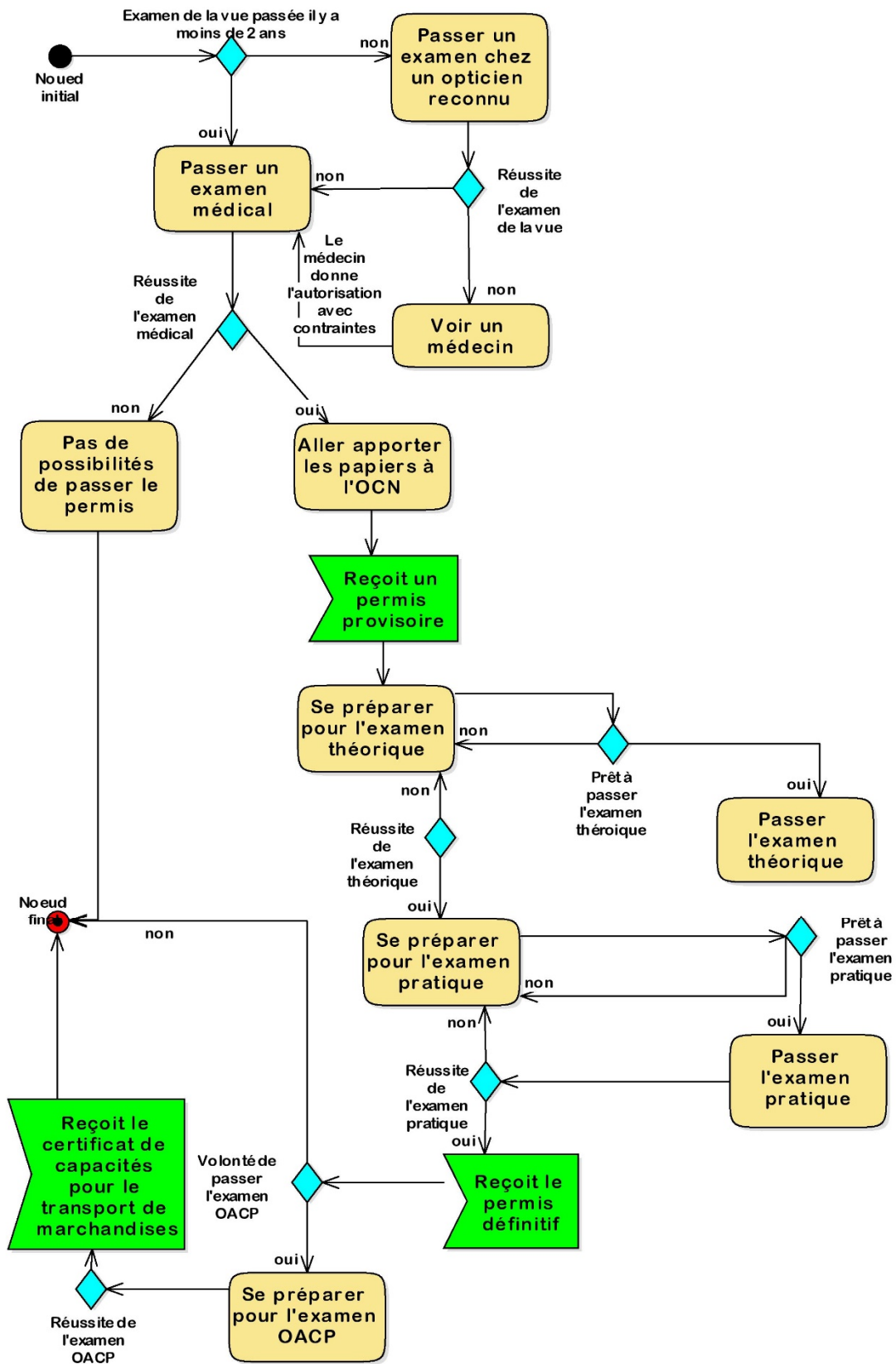


Figure 50 : Diagramme d'activités « permis C »

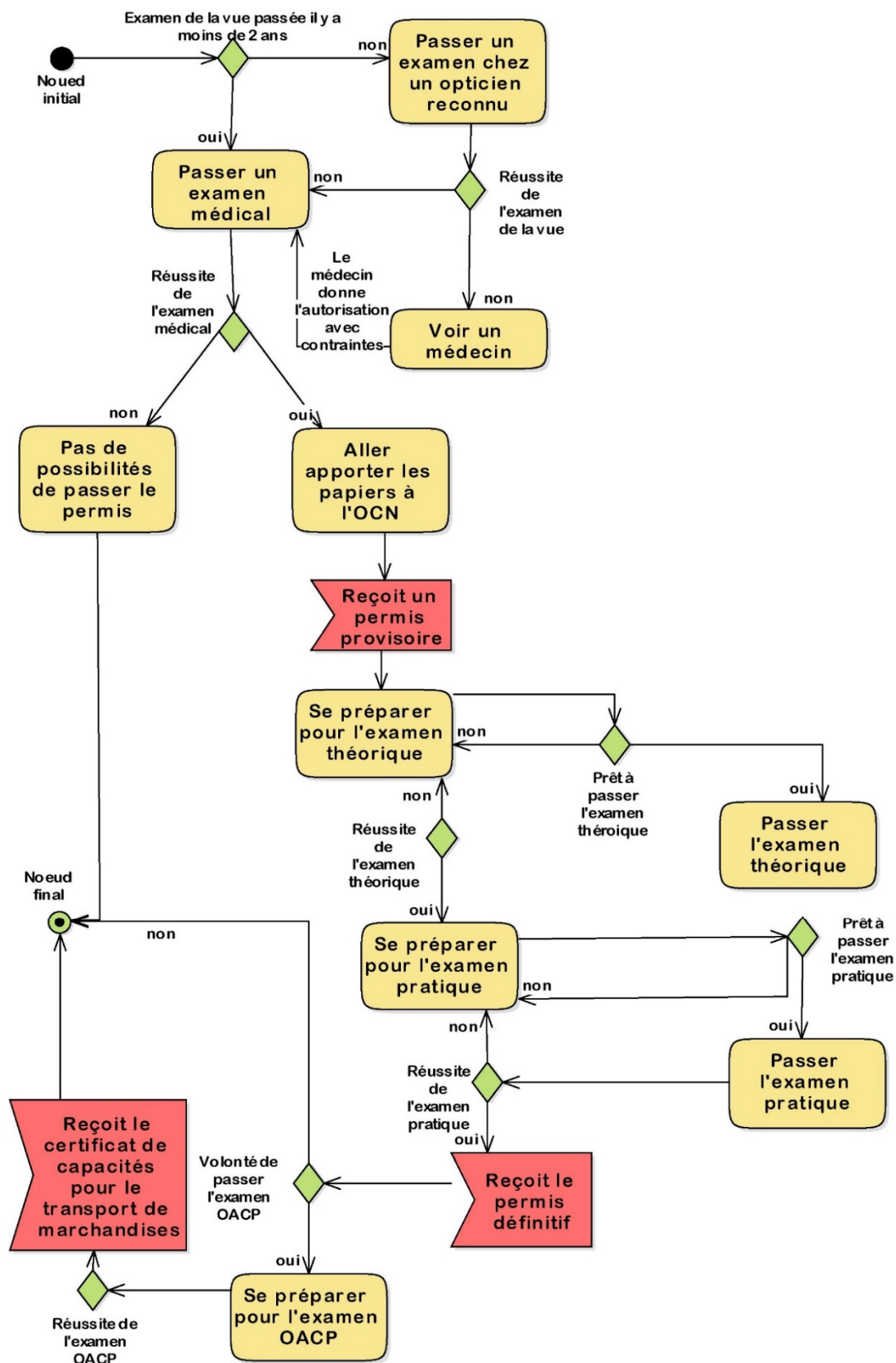


Figure 51 : Diagramme d'activités « permis C1 »

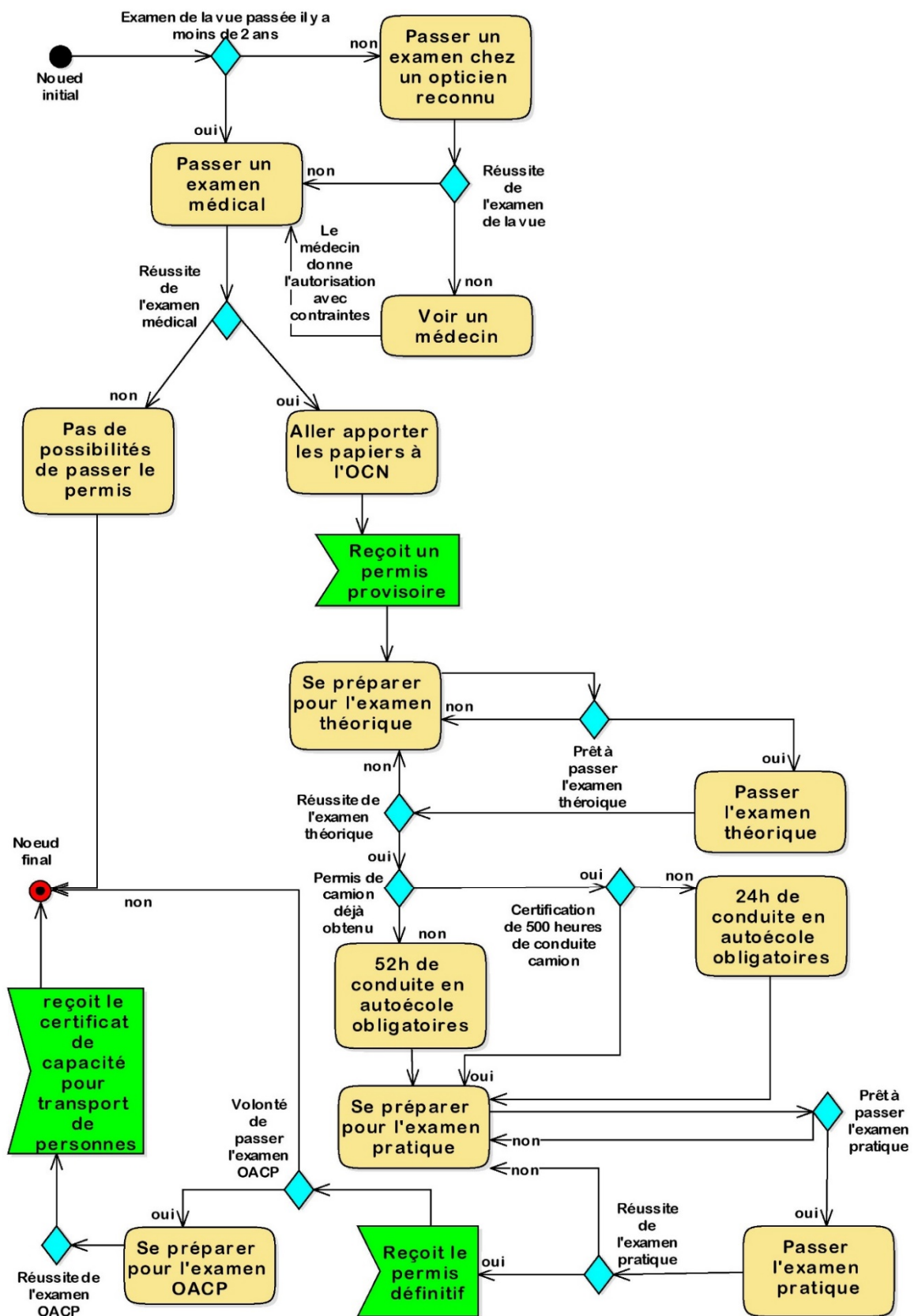


Figure 52 : Diagramme d'activités « permis D »

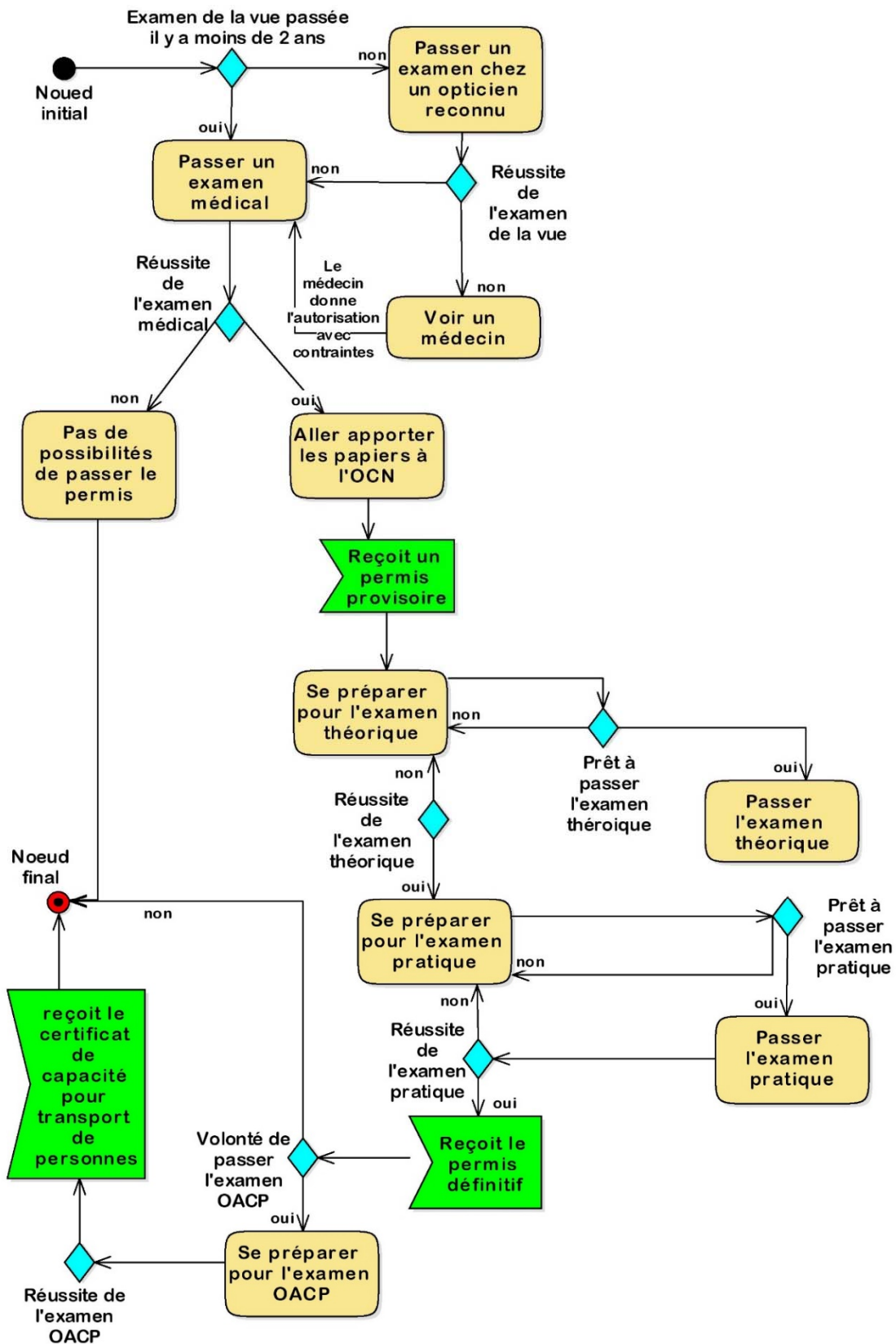


Figure 53 : Diagramme d'activités « permis D1 »

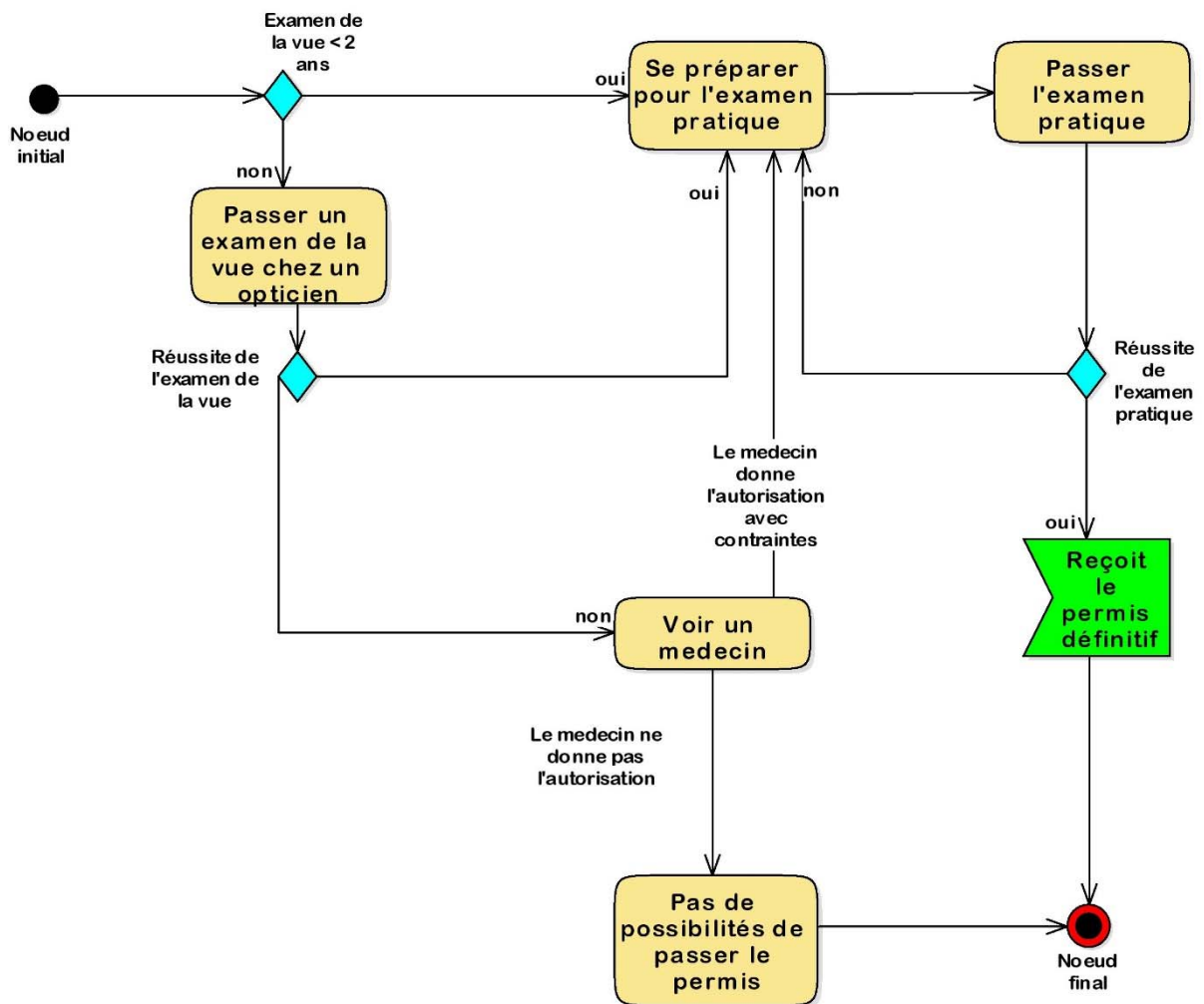


Figure 54 : Diagramme d'activités « permis E »

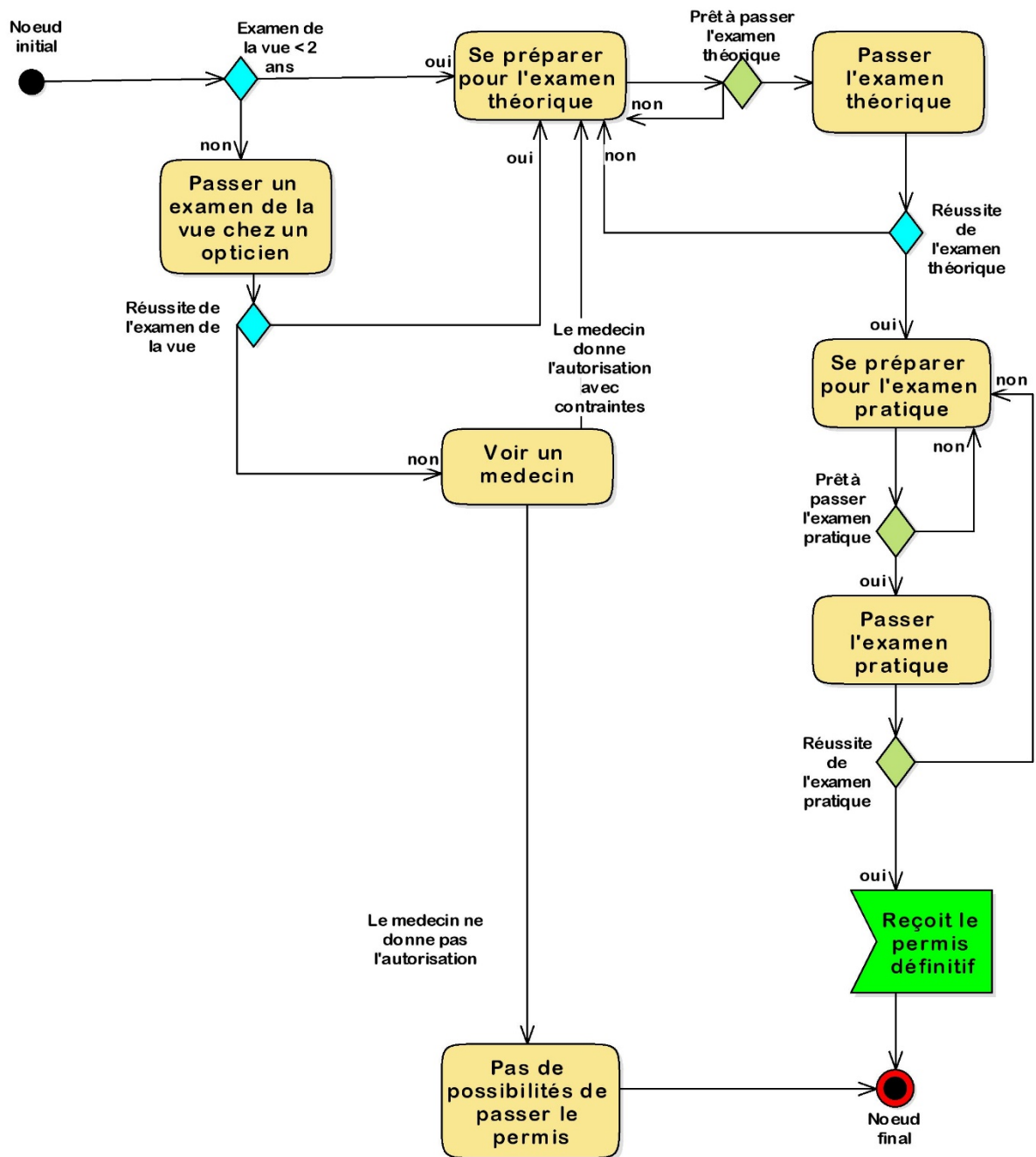


Figure 55 : Diagramme d'activités « permis F »

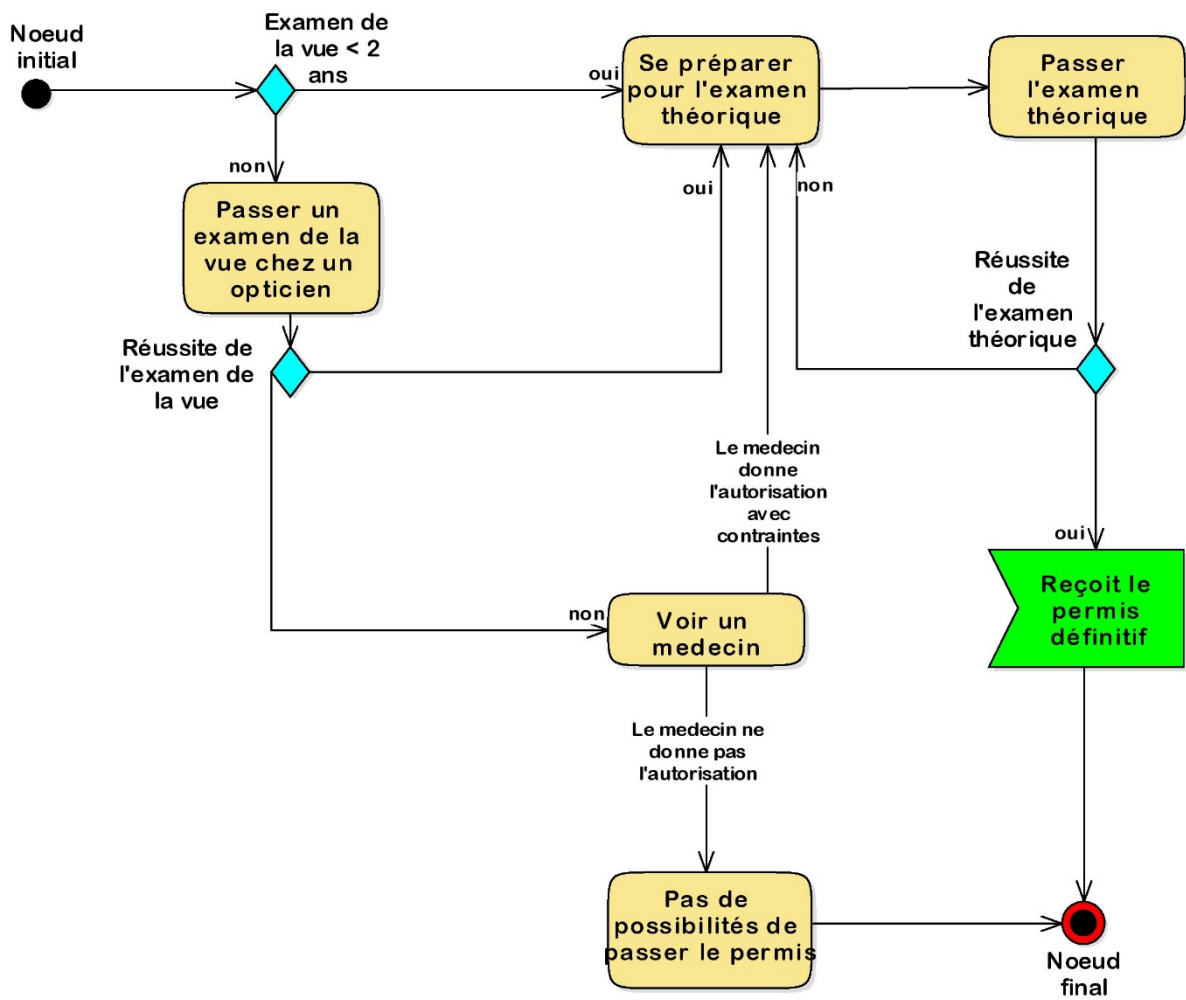


Figure 56 : Diagramme d'activités « permis G »

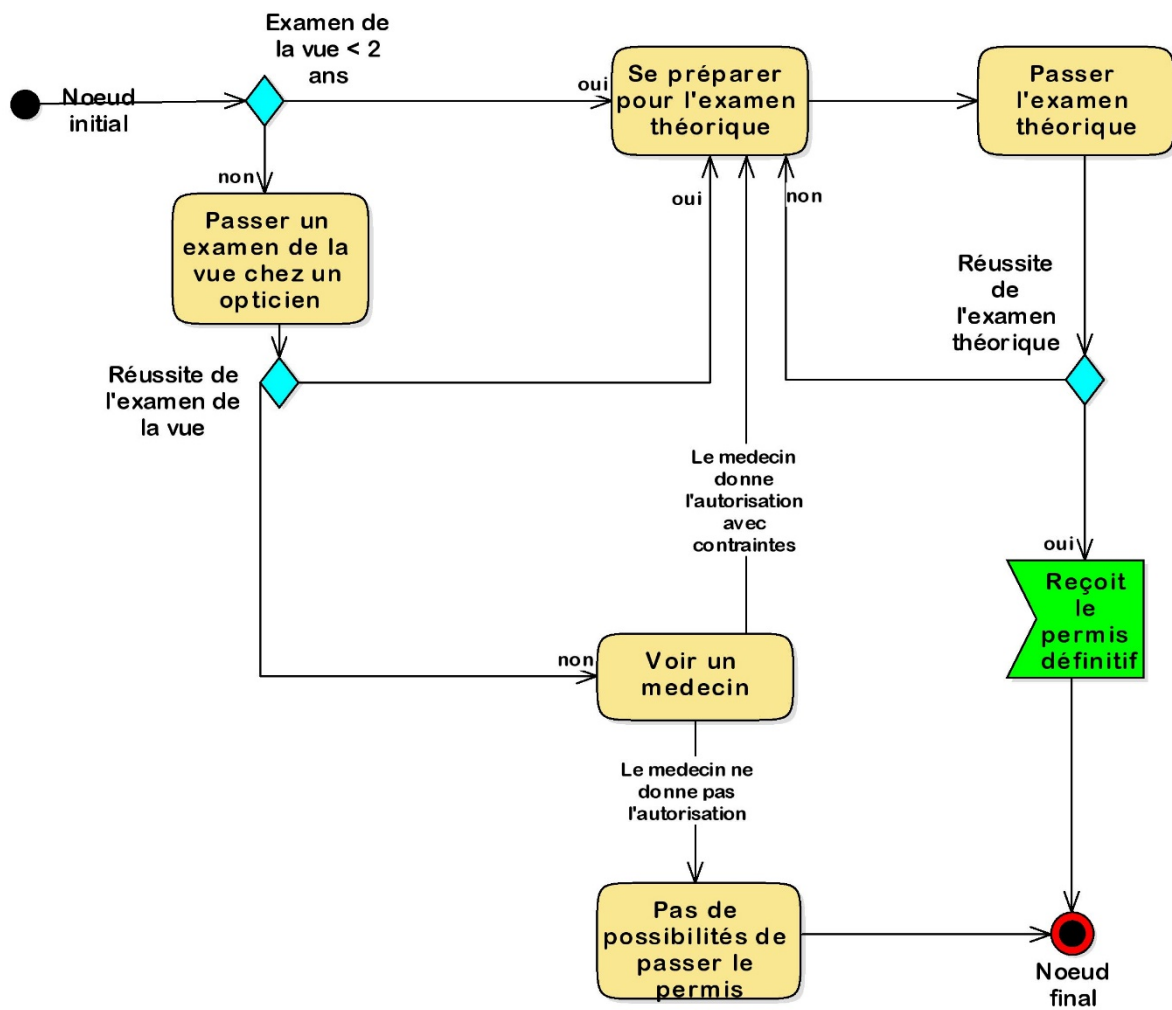


Figure 57 : Diagramme d'activités « permis M »

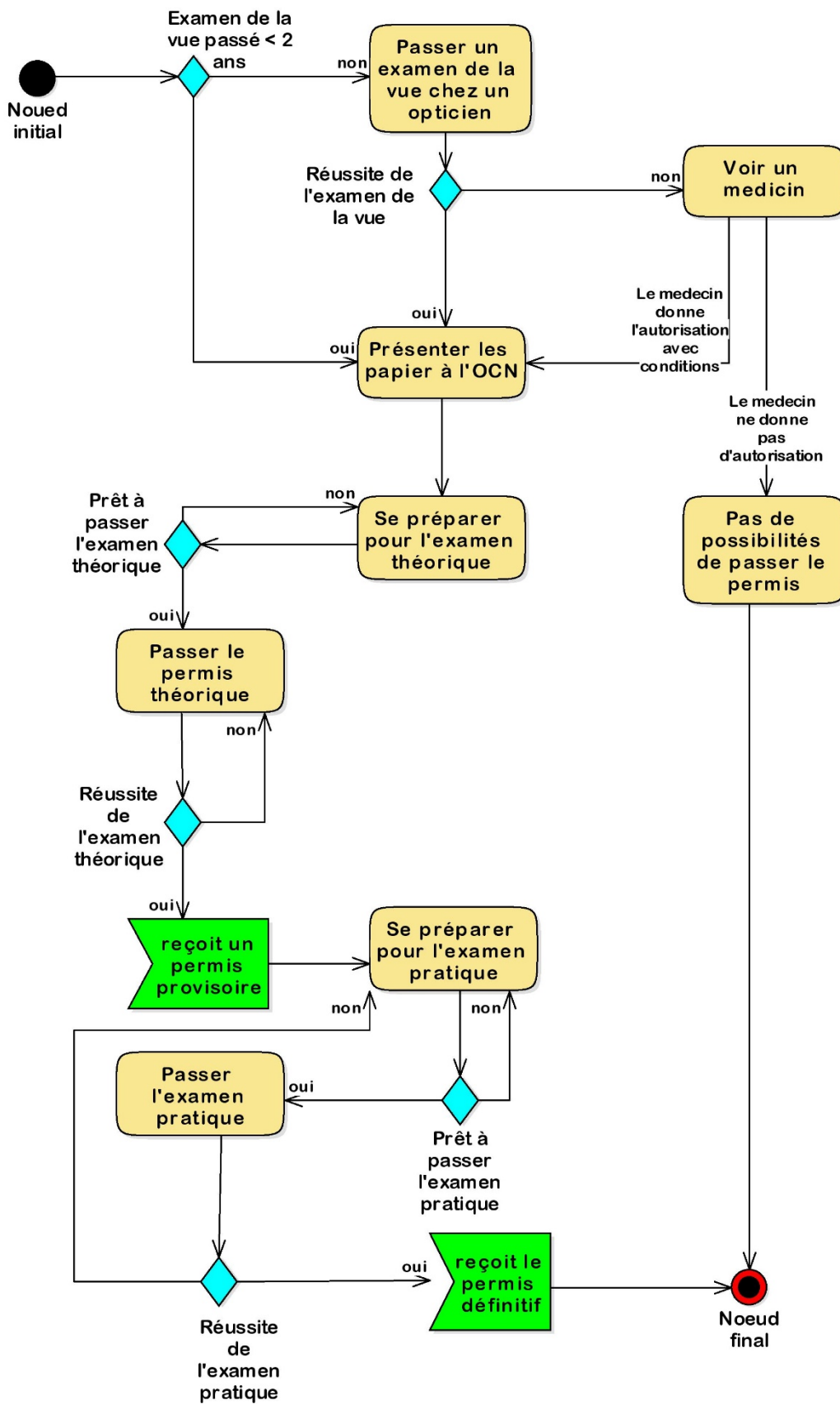


Figure 58 : Diagramme d'activités « permis bateau cat. A »

7

Ressources

7.1 Bibliographie

- Andreas Meier, Introduction pratique aux bases de données relationnelles, Paris, Springer, 2006.
- Pascal Roques, UML en action: de l'analyse des besoins à la conception en Java, Paris, Eyrolles, 2000.
- Pascal Roques, UML 2 par la pratique : études de cas et exercices corrigés, Paris, Eyrolles, 2006.
- Chantal Morley, Jean Hugues, Bernard Leblanc, UML 2 pour l'analyse d'un système d'information, Paris, Dunod, 2008.
- Pascal Roques, Le cahier du Programmeur : UML modéliser un site e-commerce, Paris, Eyrolles, 2003.
- Larry Ullman, E-commerce sans peine avec PHP et MySQL, Paris, Editions First, 2011
- Développez.com, <http://www.developpez.com>, consulté en décembre 2015.
- Documentation Symfony, <https://symfony.com>, consulté en mars 2016.
- Documentation Doctrine, <http://www.doctrine-project.org>, consulté en mars 2016.
- Documentation PHP, <https://secure.php.net>, consulté en mars 2016.
- Documentation TWIG, <http://twig.sensiolabs.org>, consulté en mars 2016.