Université Toulouse Jean Jaurès

Département Documentation, Archives, Médiathèque et Edition

Projet e-Get

*Dossier de présentation*

*Tarik BOUHJAR*

*Anaïs CALMON*

*Abigail MIRA*

*Waugoh OUEREGA*

Année universitaire 2017-2018

# Sommaire

[**Introduction 3**](#_Toc509360525)

[**1. Présentation du projet 4**](#_Toc509360526)

[Notre commanditaire 4](#_Toc509360527)

[Acteurs de l’équipe 4](#_Toc509360528)

[Commande du projet 6](#_Toc509360529)

[**2. Analyse de l’existant 7**](#_Toc509360530)

[Matrice 3QOCCP 7](#_Toc509360531)

[Logiciels existants au sein de l’Université 10](#_Toc509360532)

[Le logiciel InfoCOM\_ET 11](#_Toc509360533)

[Planning prévisionnel 12](#_Toc509360534)

[Matrice SWOT 13](#_Toc509360535)

[**3. Cahier des charges 15**](#_Toc509360536)

[Spécifications fonctionnelles 15](#_Toc509360537)

[Spécifications techniques 15](#_Toc509360538)

[**4. Conception et mise en œuvre 16**](#_Toc509360539)

[Développement 16](#_Toc509360540)

[Arborescence 20](#_Toc509360541)

[Graphique 22](#_Toc509360542)

[Modèle de données 25](#_Toc509360543)

[Langages et outils 26](#_Toc509360544)

[**5. Présentation du prototype 27**](#_Toc509360545)

[Point de vue graphisme 27](#_Toc509360546)

[**Conclusion 28**](#_Toc509360547)

[**Annexes 29**](#_Toc509360548)

# Introduction

Le présent dossier a été réalisé dans le cadre des projets transversaux du master 2 Ingénierie de l’information numérique, au DDAME, sis à l’Université Toulouse Jean Jaurès.

Notre projet a été commandité par M. Taoufiq DKAKI, directeur adjoint du DDAME et responsable de notre master. En charge de la création des emplois du temps, il a sollicité la réalisation de ce projet pour mettre au jour une nouvelle application lui permettant d’améliorer ce processus.

Le but du projet est donc la conception et la réalisation d’une application, orientée web, pour la création, la gestion et la diffusion des emplois du temps.

Nous sommes quatre étudiant-e-s du master I2N en charge du projet, qui va nous permettre de développer nos compétences tout au long du processus de création de l’application : modélisation, graphisme, codage, base de données, etc.

Nous avons fait le choix d’appeler notre projet « e-Get », d’une part pour rappeler la dimension électronique et la gestion en ligne de l’application que nous devons créer, le « Get » se faisant l’acronyme de « Gestion des emplois du temps ». La philosophie du terme *get* anglais (obtenir, réussir, apporter, gagner) correspondait également à notre volonté de mener ce projet à bien.

Le contenu de ce dossier présente l’intégralité de la méthodologie utilisée dans la réalisation du projet « e-Get ».

# Présentation du projet

## Notre commanditaire

Taoufiq DKAKI est notre commanditaire de projet : en tant que directeur adjoint mais également responsable du master 2 I2N, il est au cœur du processus de création des emplois du temps. Il intervient donc très tôt dans la gestion des UE et la répartition des heures de cours, et soulève la problématique des outils informatiques de la fac, qui demandent beaucoup d’échanges et de manipulations avant d’aboutir à un emploi du temps stable.

C’est pour alléger ces procédures de communication avec le secrétariat qu’il souhaite mettre en place le projet, ce qui évitera l’utilisation d’un grand nombre de logiciels et de recopiage de documents Excel. En tant que responsable de la promotion I2N, il souhaite non seulement améliorer le processus des emplois du temps de manière générale, mais également impliquer les enseignants du DDAME dans leur façonnement et la gestion des absences, remplacements et reports de cours.

## Acteurs de l’équipe

Nous sommes quatre étudiant-e-es du master I2N impliqués dans la réalisation de ce projet, avec des compétences variées :

|  |  |
| --- | --- |
| **Tarik BOUHJAR** | Compétences avancées en développement, conception et modélisation |
| **Anaïs CALMON** | Familière de l’interface graphique (développement *frontend*) et de la rédaction |
| **Abigail MIRA** | Compétences en gestion de projet et d’équipe, communication, rédaction, très motivée à apprendre à coder |
| **Waugoh OUEREGA** | Apte à coder, développer le graphisme, l’orientation web |

Jusqu’au début du mois de mars, nous étions 5 étudiant-e-es sur le projet, mais Asmae BENYAHYA, alors en charge de la prise en main d’Angular JS et des fonctionnalités techniques, a interrompu son cursus et quitté le projet.

Dans le cadre de notre projet, nous avons fait appel à plusieurs professionnels, pour des entretiens sur l’existant ou de l’aide sur des points techniques.

|  |  |
| --- | --- |
| **Christine POUILLET** | *Service de Gestion des salles*  Entretien réalisé pour comprendre l’utilisation de CELCAT et le processus de création des emplois du temps au niveau de l’université |
| **Patricia MOULY** | *Gestionnaire de scolarité, salles et emplois du temps*  Rencontrée pour présenter son utilisation de CELCAT et les éventuelles problématiques qui en découlent, ses attentes par rapport à un nouveau logiciel de gestion des plannings |
| **Fanny BACK** | *Secrétariat du DDAME*  Elle nous a montré le fonctionnement de CELCAT, le processus d’intégration des emplois du temps par rapport à l’occupation des salles, les problématiques liées à la gestion actuelle des plannings |
| **Madié DIAKITE** | *Enseignante – Gestion de projet*  Intervenue pour suivre la gestion des projets et la production des livrables |
| **Isabelle CRAMA** | *Enseignante – Web design*  Sollicitée pour la conception d’une palette viable pour la future application |

## Commande du projet

Plusieurs éléments entrent en jeu dans la commande de projet : d’une part la place de l’application à venir au sein du processus de création des emplois du temps, son fonctionnement propre, mais aussi le public auquel elle est destinée.

Dans un premier temps, notre projet doit s’inscrire dans un processus déjà complexe et qui utilise de nombreux logiciels, détaillés dans la deuxième partie du dossier. Notre commanditaire souhaite pouvoir faire usage d’un seul outil pour gérer les emplois du temps et les salles et alléger la chaîne de création, notamment au niveau des échanges avec le secrétariat. Déjà adepte d’un logiciel maison, créé par des étudiants de l’IUT il y a de ça quelques années, il voudrait que l’application puisse s’intégrer sans problèmes dans le processus.

Dans son fonctionnement propre, l’application doit donc pouvoir proposer les mêmes actions que le logiciel utilisé actuellement par M. Dkaki. Sa structure sera donc sensiblement identique : une base de données regroupant les informations nécessaires à la création des séances de cours, et une interface permettant d’une part d’agir sur la base de données (créer, modifier, supprimer) et d’autre part d’utiliser ces données pour construire les emplois du temps.

L’application doit ainsi être capable de gérer les données nécessaires pour la création des emplois du temps : les formations, les groupes et leur arborescence, étudiants, enseignants, UE, modules, également les données sur l’année scolaire, nombre de semaines, jours, créneaux horaires, jours fériés et vacances, etc.

Ensuite, elle doit permettre la gestion des emplois du temps avec une grille interactive : un système de drag and drop permet de modifier les modules ou les mettre en attente sur un espace prévu sur le côté, qui les sauvegarde en attendant de les replacer. De nombreuses possibilités présentes sur l’ancien logiciel, comme la propagation de modules sur plusieurs semaines et les alertes en cas d’incompatibilité, sont envisageables sur la nouvelle application. Les emplois du temps créés sont destinés à être publiés et diffusés, il faut donc prendre en compte la gestion des formats pour l’export (.ics, .ical, etc.).

La plateforme d’emploi du temps est destinée aux responsables pédagogiques et aux professeurs pour la gestion et aux étudiants pour la consultation. L’usager principal est le commanditaire, en tant qu’administrateur, puisqu’il sera amené à l’utiliser pour remplir les données et bâtir les emplois du temps. Mme Mouly, en tant que gestionnaire de scolarité et responsable des emplois du temps, peut également être considérée comme bénéficiaire, afin d’alléger ses échanges avec M. Dkaki. Enfin, il est attendu que l’application puisse permettre d’intégrer les enseignants du DDAME au processus, pour faciliter les démarches de report et replacement de cours avec les interlocuteurs concernés.

Les besoins exprimés dans la demande de projet présentent un logiciel collaboratif, permettant de prendre le relais de la gestion et la publication d’emplois du temps, et orienté technologie du web. L’objectif principal est de faciliter le processus de création et la gestion quotidienne des emplois du temps. Des objectifs secondaires peuvent également être dégagés : rendre l’outil agréable et accessible aux professeurs ; permettre une possible expansion du modèle pour la fac (ne pas limiter son organisation à celle du DDAME).

# Analyse de l’existant

Afin de percevoir les blocages liés au processus de création des emplois du temps, nous avons réalisé plusieurs entretiens avec notre commanditaire et les acteurs concernés pour établir la note de cadrage. Ils nous ont permis d’établir une matrice décrivant le projet selon plusieurs points cruciaux :

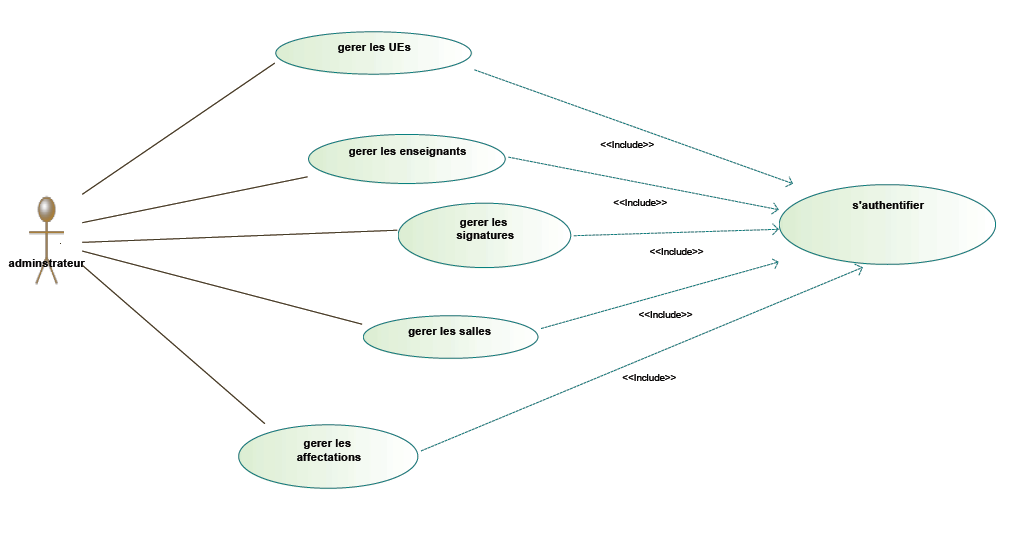
## Matrice 3QOCCP

### Pour qui ?

L’application est destinée aux responsables administratifs et professeurs pour la gestion, et aux étudiants pour la consultation. Les utilisateurs principaux sont Taoufiq Dkaki et Patricia Mouly.

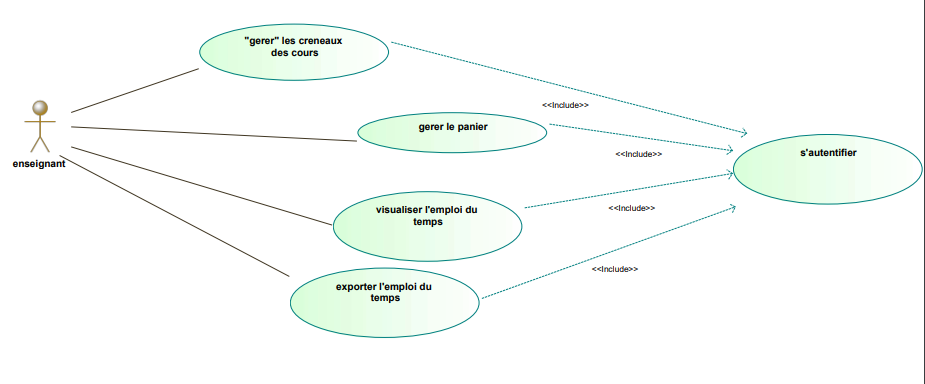
Grâce aux diagrammes des cas d’utilisation, nous avons classé les futurs utilisateurs de notre application en trois catégories :

**L'administrateur :** il a le privilège de gérer toutes les données (salles, enseignants, groupes d’étudiants, etc.), ainsi d’avoir une vision globale sur les emplois des temps, suivant des filtre personnalisés comme l’affichage par enseignant, par groupes, par salles.



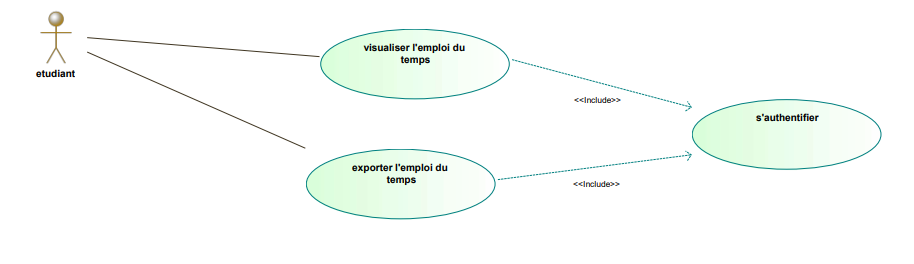
*Cas d’utilisation : administrateur*

**L'enseignant :** il a le privilège d'ajouter ses propres séances à son emploi du temps, et la possibilité de les modifier, les supprimer ou les mettre en attente.

**

*Cas d’utilisation : enseignant*

**L’étudiant :** il peut consulter l’emploi du temps de son groupe.



*Cas d’utilisation : étudiant*

### Quoi ?

Un logiciel permettant la gestion et la publication des emplois du temps, sous forme d’un outil collaboratif orienté technologie du web.

### Où ?

Le périmètre concerne le DDAME dans un premier temps, mais peut être étendu à l’Université par la suite.

### Comment ?

**Attentes :** notre commanditaire attend un fonctionnement proche de son ancienne application, qui permette de gérer à la fois les données nécessaires pour créer une séance et de construire un emploi du temps avec des outils ergonomique.

**Objectifs** **:** un meilleur fonctionnement pour la gestion des plannings pour M. Dkaki, et si possible la conversion de ses collègues responsables et enseignants au nouvel outil. La consultation directe des étudiants serait un bonus non négligeable.

**Niveau d’exigence** : un logiciel utilisable, opérationnel.

**Langage de programmation - outils** : les étudiants doivent favoriser les connaissances acquises au cours de leur formation.

### Combien ?

Il n’y a pas de budget attribué pour le projet.

### Pourquoi ?

Plusieurs éléments interviennent : l’utilisation du logiciel CELCAT n’est pas satisfaisante, le logiciel InfoCOM\_ET vieillit, le processus de création des emplois du temps n’est pas optimisé.

## Logiciels existants au sein de l’Université

Les problématiques inhérentes à la construction et la gestion des emplois du temps dépendent des logiciels existants, ou plutôt de la multitude de logiciels nécessaires au bon déroulement du processus : chaque logiciel a un apport particulier et/ou ponctuel dans le processus de création des emplois du temps.

|  |  |  |
| --- | --- | --- |
| **SAGHE**  (Système automatisé de gestion des heures d’enseignement) | Pour les services centraux de l’administration et des contrôles de gestion, les enseignants ;  Gère et suit les heures réalisées grâce à une large gamme d’outils ; s’occupe aussi des charges d’enseignement et des services. | **Gestion des données des enseignants, le statut, le nombre d’heures, etc.** |
| **HARPEGE** | Logiciel de ressources humaines, en termes de personnels, structures, mais également d’évaluation et de compétences. |
| **APOGEE** (Application pour l’organisation et la gestion des enseignements et des étudiants) | Progiciel de gestion intégrée qui gère les inscriptions (via Apoweb) et les dossiers étudiants avec la saisie des notes. | **Gestion des données sur les effectifs et les formations des élèves inscrits** |
| **IGWEB** | Application en ligne qui permet aux étudiants de s’inscrire dans les groupes de TD en fonction des UE correspondant à leur formation. |
| **CELCAT** | Logiciel de gestion interactive ou automatique des emplois du temps et des salles, utilisé dans plusieurs universités françaises. | **Centralise les données : interactions entre étudiants, enseignants et salles pour créer les emplois du temps** |

Le site de CELCAT promet un « gain de temps lors de la création des emplois du temps et plannings » au moyen des fonctionnalités du logiciel, pourtant à Toulouse, il semble poser problème. En effet, le responsable de CELCAT sur le site de l’université n’exploite pas les nombreuses possibilités du logiciel pour profiter de sa capacité de gestion, aussi n’est-il utilisé que pour les salles et non pour la partie emploi du temps.

Comme nous le confirme Mme Back, il n’est actuellement utilisé que pour avoir un aperçu de l’occupation des salles de cours attribuées au DDAME. Les données tenant des plannings sont rentrées à la main d’après une feuille Excel, créée en amont par le responsable pédagogique. L’utilité de CELCAT est donc remise en question, puisqu’il ne permet la gestion que d’une petite part du processus. En outre, certaines manipulations sont redondantes pour remplir les informations, et le logiciel manque d’ergonomie pour faciliter sa prise en main et son utilisation quotidienne. Destiné au personnel administratif, il empêche l’intégration des enseignants et chargés de cours au processus.

## Le logiciel InfoCOM\_ET

Du fait de l’impossibilité d’utiliser CELCAT pour les emplois du temps, M.Dkaki emploie InfoCOM\_ET, un logiciel conçu par des étudiants de l’IUT de Strasbourg et datant de 2002. Nous nous sommes penchés sur son cas.

|  |  |
| --- | --- |
| **Les plus** | **Les moins** |
| * Tient compte des salles, des groupes, sous-groupes, regroupements de promotions, heures complémentaires, etc. * Permet de supporter tous les cas particuliers que peut représenter une séance : plusieurs professeurs pour un module, plusieurs groupes sur une séance, plusieurs professeurs pour plusieurs groupes… | * Plannings produits sont difficilement publiables car le format utilisé empêche l’échange de données avec le secrétariat. Il lui faut donc les convertir au format Excel, puis ils sont imprimés et retranscris sur CELCAT. * Manque également d’ergonomie, il est divisé en deux parties distinctes qui doivent être lancées de deux fichiers différents + système de navigation entre les promotions n’est également pas viable avec un nombre important de diplômes. * Propose des fonctions qui sont aujourd’hui obsolètes et ne sont plus nécessaires à la gestion des emplois du temps. |

La somme de ces nombreux éléments constitue le « pourquoi » de notre projet : pour améliorer le processus de création des emplois du temps.

## Planning prévisionnel

La réalisation de la matrice a aussi permis de déterminer les délais dont nous disposions, la date de remise de l’application étant fixée, comme celle de la soutenance, au **jeudi 22 mars 2018**. Avec une estimation de temps de cinq mois, nous avons construit un planning prévisionnel comme suit :

**15/11/2017**

Cahier des charges,  
début de la modélen points positifs je mettrais qu'on a appris plein de choses sur la conduite de projet, appris à manier de nouveaux outils et qu'on s'est approprié des savoir vus en coursisation

**15/12/2017**  
Développement

**15/02/2018**  
Livraison V1

**01/03/2018**  
Tests

**15/03/2018**  
Débogage

**22/02/2018**  
Livraison  
application

## Matrice SWOT

Forts de cette prévision et en toute connaissance du contexte et de ce qui était attendu de notre application, nous avons évalué les forces et faiblesses de notre projet grâce à la matrice SWOT :

|  |  |
| --- | --- |
| **Forces** | **Faiblesses** |
| * Le projet est porté par une équipe. * Les membres du groupe ont des compétences variées et complémentaires : gestion de projet, enquête, pratiques documentaires, informatique. * Notre commanditaire est enseignant et informaticien : ses compétences lui permettent de nous accompagner, avec rigueur et bienveillance, tout au long du projet. * La commande qui nous est passée est précise et détaillée. * I2N ouvre l’accès à des ressources variées, matérielles et pratiques : salle informatique, logiciels, conseil, réseau de relations. * I2N forme des ingénieurs, chefs de projet : nous sommes formés aux les processus fondamentaux du développement d’une application web de gestion documentaire : gestion de projet, modélisation, langages structurés et webdesign. * Soutien d’acteurs compétents externes au projet : réseau personnel des membres de l’équipe projet. * Existant référentiel : notre commanditaire a mis à notre disposition la documentation d’un outil logiciel comparable à celui qu’il nous a commandé. | * Les membres du groupe ne se connaissent pas : difficultés pour évaluer les compétences de chacun-e * Hétérogénéité interne du groupe en termes de formation, de compétences : risque de difficultés pour trouver un terrain commun, un vocabulaire commun ; différences au niveau des pratiques de conduite de projet * I2N ne forme pas des développeurs : les domaines de l’architecture et de la programmation doivent être explorés en autonomie par les membres du groupe : risque de sous-évaluation de leur importance ; risque d’absence de transfert de savoir interne au groupe. * Des outils nouveaux doivent être découverts et pris en main en autonomie : risque de mauvais ciblage, risque de difficultés d’acquisition des savoir-faire nécessaires à leur utilisation. * Absence de créneaux de cours dédiés à la réalisation technique (code) du projet. |
| **Opportunités** | **Menaces** |
| * Le travail de mise en application de savoir-faire acquis en cours permettra d’obtenir de meilleures notes aux examens. * Les compétences acquises au cours du projet pourront être valorisées dans le cadre de la recherche de stage puis de la recherche d’emploi. * L’écosystème du logiciel libre offre de nombreuses solutions logiciel qui pourront être exploitées dans le cadre de notre projet. | * Multiplicité des acteurs dans le processus de production et gestion des Emplois du temps universitaires. * Multiplicité des logiciels utilisés au cours du processus, pour des tâches particulières et/ou ponctuelles. * Risque d’être confrontés à des enjeux internes aux services qui pourraient être concernés par l’outil qui nous est commandé : prérogatives professionnelles, accès à et utilisation des ressources. * Contexte universitaire de mobilisation locale liée à des problématiques de gouvernance : risque de perturbation des activités sur le campus, risque sur l’accès aux services, aux personnes. |

# Cahier des charges

## Spécifications fonctionnelles

Le futur logiciel doit répondre aux besoins fonctionnels suivants :

**Accès sécurisé** : l’utilisateur doit saisir un login et un mot de passe pour accéder à son espace.

**Gestion des données** : l’administrateur peut ajouter et modifier des diplômes, des UE, des modules, des enseignants, des groupes d’étudiants et des salles ; il peut faire des liens entre ces données, grâce à l’affectation ou aux signatures.

**Affichage de l’emploi du temps** : pour l’administrateur, l’emploi du temps présente plusieurs options :

* Affichage par jour,
* Affichage par semaine,
* Affichage par mois.

Une option de recherche permet également de filtrer l’affichage :

* Par enseignant,
* Par groupe,
* Par salle.

**Importation et exportation**: le logiciel doit permettre d'exporter l’emploi du temps suivant des formats exploités par d’autres applications de gestion de calendrier, comme Google Calendar. Il doit également permettre d’importer des fichiers extérieurs d’emploi du temps.

## Spécifications techniques

**Application web centralisée**: elle donne aux différents utilisateurs une vision à jour de l’emploi du temps, sans passer par des personnes tierces ; cela permet de gérer le conflit de disponibilité des salles a un moment donné.

**Accessible à distance** : un accès à distance via internet permet d’offrir aux utilisateurs la possibilité de gérer les emplois du temps sans contrainte géographique.

**Persistance des données dans la base** : la base de données permet de sauvegarder l’historique de l’emploi du temps et de lever des exceptions suivants des contraintes liées au modèle de données.

**Interface graphique interactive** : l’interface graphique de l'application doit offrir une interaction à l’utilisateur, comme le *drag and drop* et le *resize*.

# Conception et mise en œuvre

## Développement

### Démarche de gestion de projet

Concernant le développement de notre logiciel nous avons choisi, d’un commun accord au sein du groupe et sur les conseils de notre enseignant de Programmation et production web, Mr Taoufiq Dkaki, de travailler en « mode agile ».

Quelques mots sont nécessaires pour présenter ce mode d’organisation du travail de développement de logiciels, avant que nous ne fassions le récit des différentes étapes du développement de notre produit.

Les formes d’organisation du travail dites « méthodes agiles » émergent dans le dernier quart du XXe siècle, avec l’essor des techniques de l’informatique.

Différents outils convergent autour de pratiques de pilotage et de réalisation de projet formalisées en 2001 dans le Manifeste Agile[[1]](#footnote-1) :

Nous découvrons comment mieux développer des logiciels  
par la pratique et en aidant les autres à le faire.  
Ces expériences nous ont amenés à valoriser :

**Les individus et leurs interactions**plus que les processus et les outils  
**Des logiciels opérationnels**plus qu’une documentation exhaustive  
**La collaboration avec les clients**plus que la négociation contractuelle  
**L’adaptation au changement**plus que le suivi d’un plan

Nous reconnaissons la valeur des seconds éléments,  
mais privilégions les premiers.

L’agilité se caractérise un mode de développement qui est pragmatique ; itératif, incrémental et adaptatif ; et centré sur l'autonomie des ressources humaines impliquées dans la spécification, la production et la validation d'une application intégrée et testée en continu.

Ce mode de conduite de projet nous a semblé mieux adapté à notre projet que les modèles traditionnels en cascade ou en V.

Dans le premier, hérité de l’industrie lourde, les phases sont menées les unes après les autres, le développement succédant à la mise en place d’un cahier des charges détaillé définissant au préalable l’ensemble des livrables et le date précise de livraison du produit. Cette méthode présente l’avantage d’un solide travail de documentation établissant clairement les besoins, les processus à mettre en œuvre, les spécifications logicielles, l’architecture logicielle et les tests à mener. Bien adaptée aux structures complexes regroupant des centaines de personnes sous une autorité centrale, elle l’est moins à un projet logiciel impliquant peu de personnes, sur un temps court, sur un même site et travaillant de manière auto-organisée. Ses risques principaux nous ont semblé considérables : l’impossibilité technique de suivre un cahier des charges qui se révèlerait inadapté, la possibilité de passer trop de temps sur le cahier des charges, au détriment du développement, et l’émergence d’imprévus auxquels le cahier des charges ne pourrait s’adapter.

Dans le second mode, ce modèle en V ou semi-itératif, on retrouve cette même disjonction entre les phases de conception et de réalisation. Ici deux phases classiques se succèdent : d’abord la recherche sur l’expression des besoins et la conception de la solution, ensuite la construction du produit. Ici aussi les risques liés à une division stricte entre la conception et la réalisation semblaientt considérables.

De plus, ces deux modèles nous auraient imposé une division du travail interne au groupe basée sur les différentes compétences détenues par ses membres au départ du projet, plutôt que de permettre la coopération et la collaboration tout au long du projet, avec des efforts d’application visant à l’acquisition des contenus vus en cours.

### Scrum

À la différence de ces modèles classiques, les méthodes agiles nous semblaient particulièrement adaptées au travail à réaliser. Parmi elle nous avons particulièrement puisé dans les pratiques du schéma d’organisation Scrum.

Les grandes valeurs sur lesquelles se fonde le schéma Scrum sont la collaboration, l’autonomie et la pluridisciplinarité, ses « trois piliers » étant la transparence, l’inspection et l’adaptation. Ensemble, ces concepts indiquent la philosophie derrière cette façon d’organiser le travail : les différentes composantes de l’équipe projet (commanditaire et développeurs) doivent travailler de concert, dans le respect des compétences de chacun et le partage des informations, en tenant compte des résultat produits et en faisant preuve de réactivité et de souplesse. Afin de résoudre des problèmes complexes (comme la production d’un logiciel), Scrum préconise une division du problème en tâches individuelles, la priorisation de chacune de ces tâches, et l’attribution de chacune d’entres elles à la personne la plus compétente.

Dans le contexte particulier de notre travail universitaire nous avons choisi d’appliquer ces préconisations avec quelques adaptations : ainsi, la répartition des tâches s’est faite en fonction des centres d’intérêt de chaque membre du groupe, sans forcément se baser sur une maitrise préalable du processus à mettre en œuvre, puisqu’il s’agit d’un travail d’application dans le contexte d’une acquisition de savoir-faire et de connaissances.

Le schéma Scrum offre une méthodologie simple autour de quatre items : les acteurs, les outils, les pratiques et les artefacts.

Nous avons ici aussi adapté la méthodologie à nos besoins : ainsi, nous avons créé un outil de priorisation et de suivi des tâches inspiré du Backlog préconisé, et adopté un rythme de travail assimilable aux Sprints de développement incrémentale du produit. Quand aux rôles, nous avons opté non pas pour la désignation d’un Scrum Master responsable de la bonne marche du projet, mais plutôt une « casquette tournante » de chef de projet, chacun-e d’entre nous ayant ainsi à se familiariser avec ce rôle.

La souplesse caractéristique de ce mode d’organisation nous a été bénéfique lorsque nous avons du faire à des bouleversements en cours de route : le départ d’un membre de l’équipe et l’abandon de l’outil initialement choisi pour le codage (framework Symfony)

### Sprints : Enquête, modélisation, codage

Le développement *per se* a pris la forme de sprints concernant les trois étapes clé de l’enquête, de la modélisation, et du codage.

**L’enquête** évoquée ci-dessus a prit plusieurs formes, dont l’évocation successive ne doit pas laisser oublier que leur conduite a été concomitante, répondant à des itérations et des processus de feed-back en enrichissant chaque phase. D’abord, des interviews *in situ* avec notre commanditaire ainsi qu’avec des personnes concernées par le process de création et de gestion des emplois du temps. Ensuite, de la recherche documentaire, avec notamment le dépouillement de la documentation du logiciel Infocom\_ET. Enfin, des retours réguliers auprès de notre commanditaire, alors place en position de *Product Owner*, afin de faire évaluer et valider nos propositions portant sur l’outil que nous élaborions (voir Annexes).

La **modélisation** a également été un processus de longue durée, itératif et incrémental. Le premier travail visant à établir un modèle conceptuel de données (MCD) a été suivi de retours réguliers en lien avec la mise en place d’une base de données basée sur le modèle logique de données en étant issu.

Plus concrètement, une fois établit le MCD et le MLD à l’aide du logiciel JMerise, nous avons utilisé le logiciel MySQL Workbench pour construire un modèle EER (*Enhanced entity-relationship* model ou modèle entité-association étendu), puis créer la base de données grâce à la fonctionnalité de *forward engineering* de cet outil. Ceci nous a fait gagner un temps précieux puisque une fois que dans le schéma étaient définies toutes les associations, établis et typés tous les attributs, et indiquées toutes les clés primaires et étrangères, le logiciel à non seulement généré le script sql pour créer la base et ses tables mais également exécuté ce script sans qu’il faille procéder à aucun changement de logiciel.

MySQL Workbench s’est d’ailleurs révélé un outil précieux pour ce qui est du travail sur la base de données (peuplement, modifications de tables ou de colonnes), puisque c’est un logiciel qui se connecte à la base de données et permet d’agir sur le modèle et les données (éditeur de requêtes) au sein d’un même espace de travail (voir captures d’écran en Annexe), et de les synchroniser.

Parallèlement, puis en cohérence avec les résultats de la modélisation, nous avons établi les maquettes fonctionnelles de notre application : interface utilisateur et arborescence des pages (statique et dynamique). Les itérations et ajustements caractéristique d’un travail en mode agile on ici concerné les champs des l’interface utilisateur et les attributs de différentes tables de la base de données. Par exemple : l’attribut Code\_UE de la table UE était initialement typé INT (entier), ce n’est qu’au moment d’envisager la saisie de ce code par l’utilisateur, et l’importance de ce code dans la manipulation, par l’utilisateur, des données relatives à un emploi du temps, que nous avons réalisé que cet attribut devait être typé VARCHAR (chaine de caractères), de sorte à pouvoir utiliser les codes déjà en usage dans notre université, de type alphanumériques.

En ce qui concerne le **codage**, enfin, nous avons pu apprécier tout le bénéfice offert par la souplesse de la philosophie agile.

Un premier sprint a permis l’établissement de maquettes de l’interface, codées en HTML+CSS avec l’utilisation de Bootstrap.

Concernant les traitements, nous avons rencontré de réelles difficultés. Dans un premier temps, il a été décidé d’utiliser les frameworks Angular et Symfony, censés nous faire gagner du temps à travers l’automatisation et la modularité. Toutefois, ces outils ont été abandonnés tardivement, et nous nous sommes tourné-e-s vers un travail plus simple de codage en PHP.

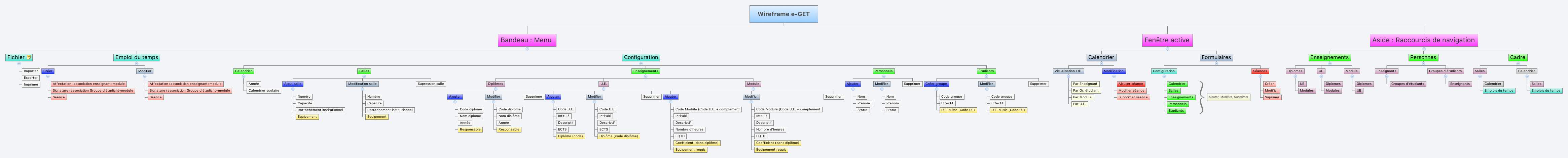
En l’état, nous avons utilisé Bootstrap et Ajax pour le frontend, PHP pour le backend. Finalement, ces nouveaux choix technologiques nous mettent en cohérence avec la méthodologie de conduite de projet adoptée : dans l’esprit de Scrum et de l’*Extreme programming* nous avons redéfini nos objectifs autour de l’impératif KISS (*Keep It Simple Stupid*) ; privilégié la création de fonctions permettant de réduire la quantité de code à écrire (*refactoring*), travaillé en binôme et pratiqué l’intégration continue, le tout au service de la production d’un Produit Minimum Viable (MVP) répondant au niveau d’exigence exprimé par notre commanditaire.

## Arborescence

Nous avons choisi une organisation en trois parties offrant différentes entrées et voies de navigation au sein du l’interface.

Les plans suivants permettent de visualiser tant l’organisation (architecture) que le fonctionnement des différents éléments de notre application.

### Plan du logiciel: statique



Les couleurs représentent des informations quant à la nature des éléments représentés. Par exemple : pages de même niveau ou de même nature partageant une même couleur bien que situées en différents point de l’arborescence ; champs représentant des attributs en position de clé étrangère. Ce document sert ainsi notamment de soutien au travail sur les requêtes sql permettant de rendre l’application dynamique.

(voir version en Annexe)

### Plan du logiciel: dynamique



Différentes formes donnent ici des indications sur les dynamiques relevant du fonctionnement de l’application : échanges avec la base de données, entrées et sorties de données. Ceci permet d’avoir une représentation précise de la navigation que nous souhaitons implémenter.

## Graphique

Notre ambition au niveau du graphisme du projet était de présenter une application avec une qualité graphique professionnelle, un design léger et adapté à sa cible, et indépendante de la charte graphique du DDAME.

### La recherche d’une qualité graphique professionnelle

Le graphisme est réputé pour être la première chose à laquelle les utilisateurs prêtent attention en découvrant une application, un site web avant de s'intéresser au contenu. Nous avons donc tenu compte du fait que la qualité graphique d’une application est une de ses composantes les plus importantes quant à sa crédibilité.

Bien entendu, les goûts et couleurs restent très subjectifs et dépendent du contexte, que nous aborderons en suivant. Toutefois, des éléments tels que le contraste, la gestion équilibrée de l’espace ou la qualité des compressions peuvent être objectivés.

Nous avons donc fait l’effort de soigner notre finition, d’opter pour des qualités respectables pour les images telles que les logos ainsi qu’une palette de couleur cohérente, que nous justifierons par la suite.

### Un design léger et adapté à sa cible

Un design léger est celui qui permet ou n’empêche pas la page de charger rapidement. Nous avons par conséquent fondé autant que possible notre design sur du HTML avec du Bootstrap, et minimisé l’utilisation d’images.

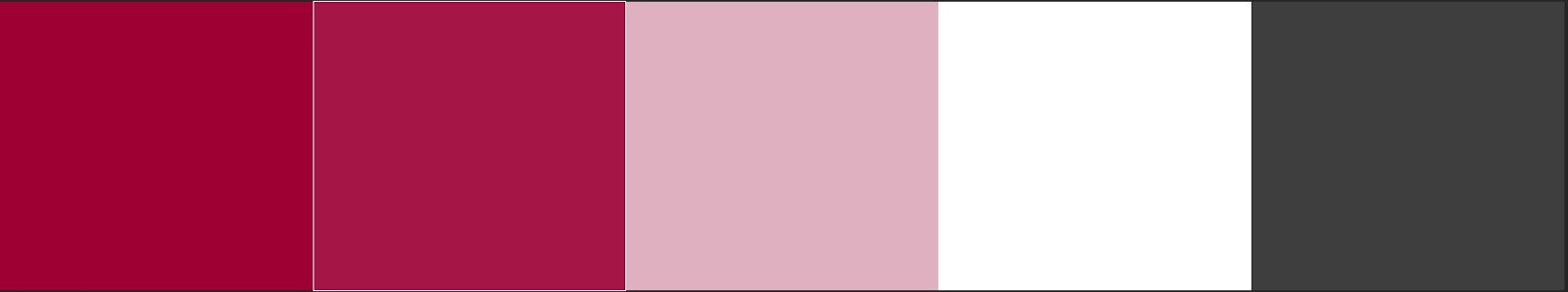
La cible de notre projet dans un premier temps était notre commanditaire Monsieur Taoufiq Dkaki, dans un second temps le secrétariat du DDAME et de façon plus large l’administration de l’Université Toulouse Jean Jaurès.

Le design de la première version de notre application a essentiellement tenu compte des “goûts” de Monsieur Dkaki. De fait, bien que sans préférences particulières quant au choix des couleurs, il a montré une nette préférence pour des couleurs dans les tons pastels et neutres. C’est donc en tenant compte de cette tendance que nous avons amélioré nos propositions (tous les visuels sont disponibles en annexes).

#### Échéanciers de couleurs des premières propositions

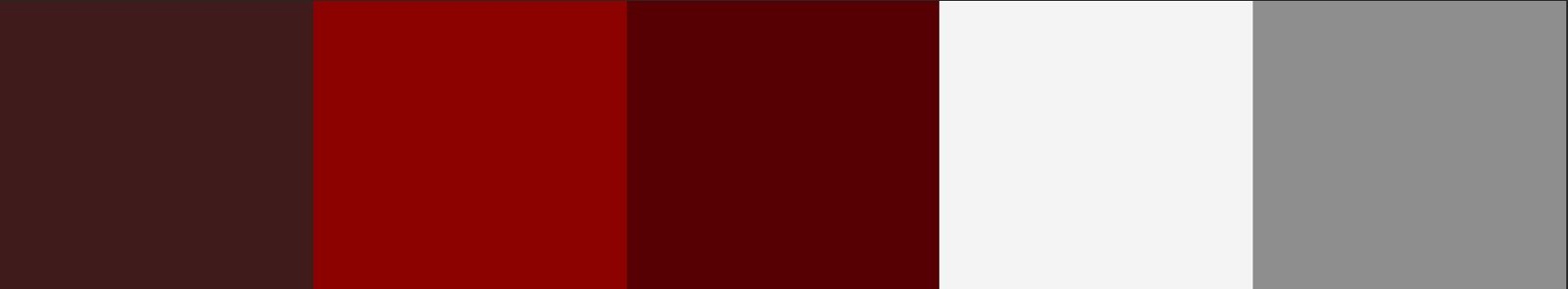




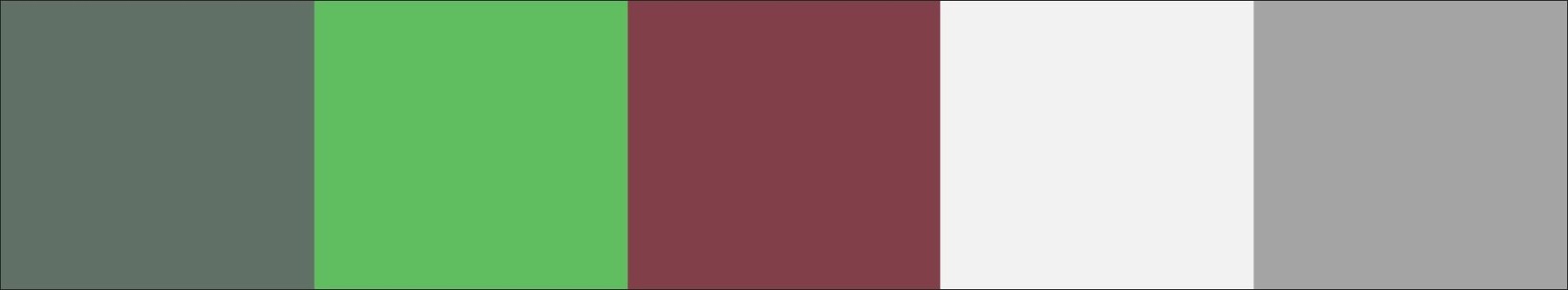








#### Proposition retenue



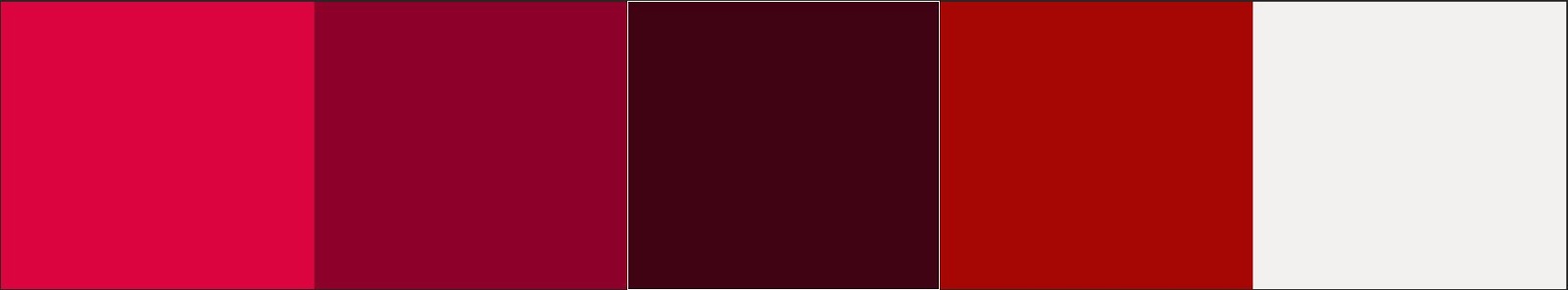
*Echéancier de couleurs de notre application e-Get*

Nous avons essayé de conserver une touche de rouge tout au long de nos propositions afin de rappeler quelque peu le site du DDAME. En ce qui concerne notre proposition retenue, elle a justement été élaborée sur la base de ce rouge grâce au site www.paletton.com. Le vert et le gris autour répondaient à la préférence de notre commanditaire, en plus d’être la couleur complémentaire du rouge et une teinte neutre avec le blanc.

### Une indépendance par rapport la charte graphique du DDAME

Le commanditaire, Monsieur Dkaki, lors d’une première réunion à propos du graphisme du DDAME nous avait laissé carte blanche. Un nouveau design rentrait ainsi en conformité avec la logique de l’expansion de l’application à toute l’université.

De plus, les couleurs de l’ancienne charte graphique, en fonction de leur utilisation pouvaient facilement paraître agressives. En effet, on y observe une grande dominante du rouge.

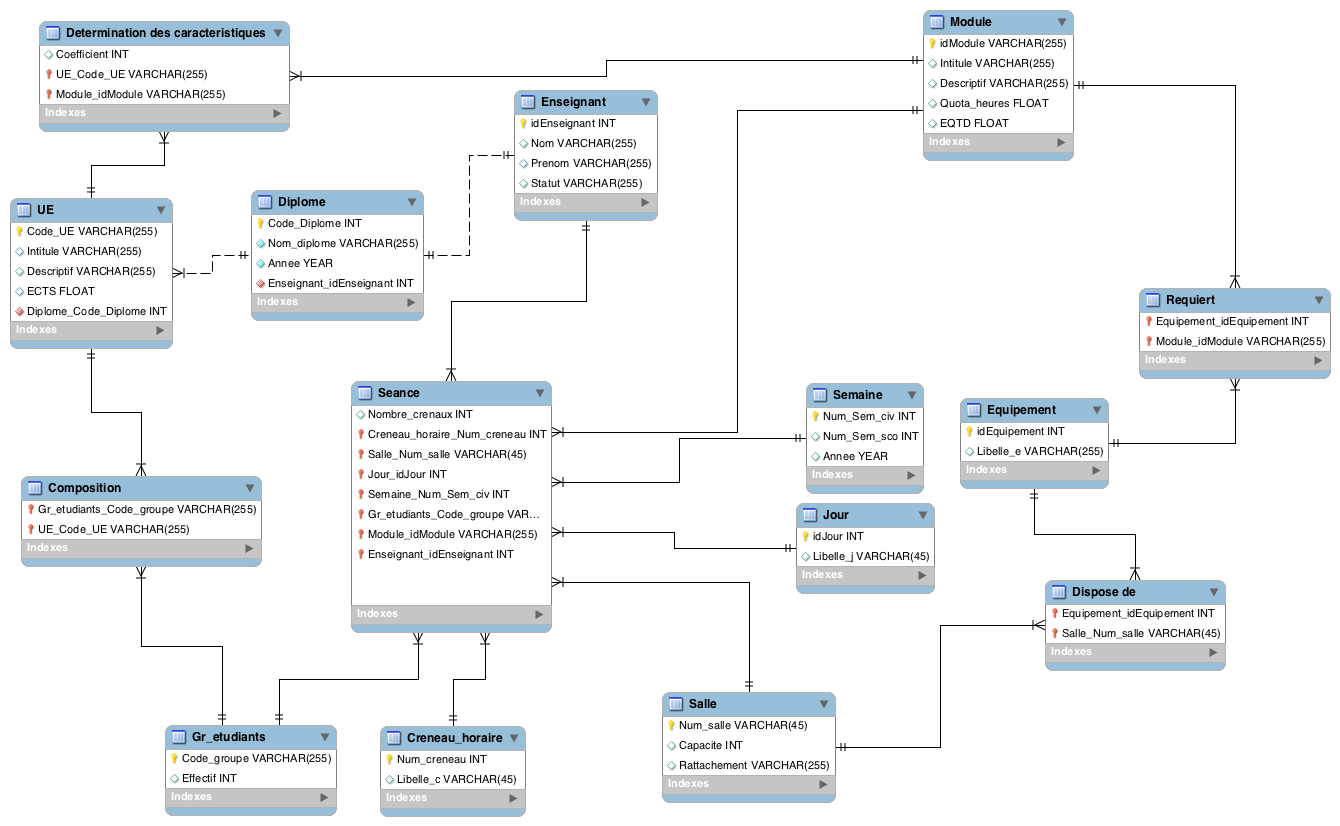


**

*Echéanciers de couleurs du site du DDAME*

Bien que le rouge soit une couleur chaude et saisissante, elle est souvent assimilée au danger en communication visuelle, ainsi qu’aux avertissements et interdictions.

## Modèle de données



Le schéma comporte 15 tables :

* 10 correspondant à des entités
* 5 correspondant à des associations

La table « Séance » est au cœur du dispositif de conception des emplois du temps. Elle correspond à l’association de 7 des 10 entités.

## Langages et outils

### Langages utilisés

**Frontend** : HTML, CSS, Boostrap, JavaScript, Ajax, librairie FullCalendar (jQuery) ;

**Backend** : PHP.

**Base de données** : SQL, mySQL.

La plupart de ces langages nous ont été enseignés au cours de notre formation, leur utilisation semblait par conséquent évidente pour ancrer nos acquis dans la réalité du projet.

### Outils de gestion de projet

**SCOUT** : espace collaboratif mis à disposition grâce à l’ENT, permettant la mise en commun de la documentation sur le projet (CRR, fiche projet, etc.). En outre, il est accessible à tous les étudiants et enseignants avec l’adresse mail de l’université, permet de réunir nos fichiers mais également l’édition en ligne, l’organisation en dossiers et sous-dossiers.

**Adresses mails personnelles** : utilisées pour la communication avec le commanditaire et la transmission des CRR et ODJ, média de transfert officiel et rapide des informations sur le projet.

**Slack**: application de communication instantanée (identique au fonctionnement de Messenger), Slack intervient pour la communication régulière du groupe, notamment grâce à sa disponibilité sur mobile pour la réactivité des réponses, et le côté moins formel par rapport aux mails.

**Trello**: application permettant d’organiser et positionner des cartes, utilisée pour mettre en place notre backlog et les tâches à réaliser ; la possibilité d’ajouter des étiquettes de couleur, des deadlines, de notifier les participants, attribuer des responsabilités, etc. constitue un atout majeur dans son utilisation.

### Outils de développement

**Modélisation** : JMerise ; un logiciel libre et pratique, déjà utilisé lors de la formation de Master 1 au DDAME, et recommandé par nos enseignants de modélisation. MySQL Workbench, offrant davantage de fonctionnalités (voir *supra*). XMind et Pencil Project pour la réalisation des plans de l’architecture.

**Editeurs de texte** : Sublime Text, Notepad++, Atom ; chacun a travaillé avec l’outil dont il est le plus familier.

**Serveur**: WampServer ; c’est également un outil que nous avons largement utilisé pour nos cours de CMS ou de programmation, son choix était par conséquent aisé. Mamp, pour le travail sur MacOS X.

**Travail collaboratif** : pour la mise en commun des différentes parties sur lesquels nous travaillions, nous avons opté pour Git et le dépôt de fichiers GitHub, nous permettant d’avoir des sauvegardes de toutes les modifications apportées aux versions de code et de travailler à plusieurs sur les fichiers.

# Présentation du prototype

## Point de vue graphisme

Notre application est responsive, notamment grâce à l’utilisation de Bootstrap et des pourcentages comme unités de valeurs dans le CSS.

Perspectives pour la V2 :

* Utilisation de fenêtres pop-up (Bootstrap et Javascript) plutôt qu’une redirection sur une autre page,
* Graphisme susceptible d’être amélioré afin de répondre à des préférences plus générales, voire personnalisable ?
* Possibilité d’importer son emploi du temps dans son agenda personnel (iCalendar, Outlook, Google Calendar, …)
* Une version mobile de consultation pour les étudiants et professeurs

# Conclusion

Le travail sur le projet transversal du Master 2 a été la source de nombreux enseignements pour notre groupe, et l’occasion de réaliser une analyse critique de nos propres compétences dans la gestion de sa conduite.

Points positifs : nous avons beaucoup appris en matière de conduite de projet ; nous avons appris à manier de nouveaux outils ; nous nous sommes approprié des savoirs et savoir-faire étudiés dans le contexte de nos cours.

Difficultés: nous avons rencontré des problèmes avec le suivi régulier de la méthode Scrum et de la méthodologie de conduite de projet. Il nous a manqué la production de certains livrables nécessaires, tels la note de cadrage, et nous avons eu des difficultés à mettre en place et nous tenir à une planification revue et contrôlée au fur et à mesure de l’avancée des tâches.

Nous avons rencontré des problèmes de communication après des périodes sans réunion, nous avons dans une certaine mesure manqué d’une reprise de contact rapide entre les membres de l’équipe et d’un suivi du travail personnel sur le projet.

Enfin, la principale erreur au niveau de la conduite du projet fut la décision initiale d’utiliser des frameworks pour réaliser le travail du back et frontend : une trop faible part du groupe maîtrisait ces technologies et elles ne pouvaient finalement pas servir à l’équipe, excluant la majorité de leur utilisation. Leur abandon total tardif nous a de surcroit obligé-e-s à revoir à la baisse nos ambitions quant au produit final.

# Annexes

[Fiche de projet validée II](#_Toc509360219)

[Modèle du logiciel InfoCOM\_ET IIV](#_Toc509360220)

[Fonctionnalités extraites de la documentation d’InfoCOM\_ET IIV](#_Toc509360221)

[Premières recherches et propositions graphiques V](#_Toc509360222)

[Charte graphique X](#_Toc509360223)

[Évolutions du modèle XIX](#_Toc509360232)

[Travail sur la Base de données (MySQL Workbench) XXI](#_Toc509360233)

[Plan de l’application XXIII](#_Toc509360234)

|  |  |  |
| --- | --- | --- |
| Résultat de recherche d'images pour "université toulouse jean jaures" M2 I2N ‑ 2017-2018 | **FICHE PROJET**  **« e-Get »** | **Date du document : 05/12/2017** |

**Descriptif du projet**

M. Dkaki a commandé un système de **conception**, de **gestion** et de **publication des emplois du temps** des équipes enseignantes du DDAME. Il souhaite un outil collaboratif orienté technologie du web.

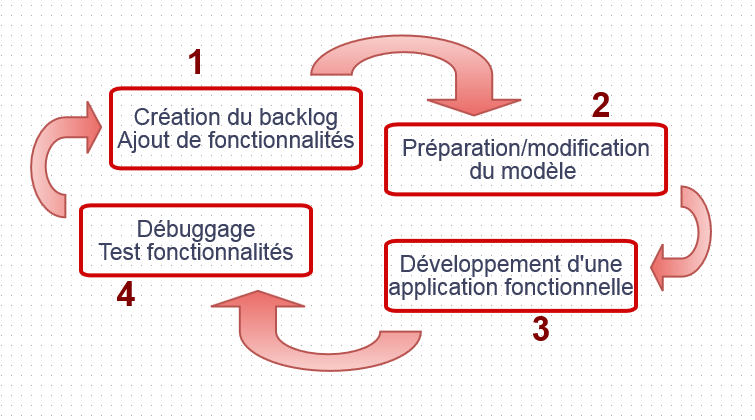
Le logiciel doit être **simple**, **efficient** et **ergonomique** pour surmonter les réticences à son utilisation. Son modèle doit être idéalement applicable à d’autres départements.

**Phasage du projet**

La présentation de l’application finie est prévue pour le **22 mars 2018**. Voici un rétro-planning prévisionnel :

15 nov. 2017 : cahier des charges, modélisation  15 décembre : développement  15 février 2018 : livraison V1  1 à 2 semaines de test  2 semaines de débogage.

Après l’analyse de l’existant et la validation de la fiche projet, travail de l’équipe en mode agile :



Une fois les principales fonctionnalités demandées testées, le produit est livré.

**Logistique – Organisation**

Prise en compte applis existantes / serveur en cours de développement, déploiement sur le SI ? Infrastructure = université.

Outils envisagés : **Symfony** (php), **Angular** (JS, HTML, CSS)

Méthodologie de conduite projet : mode **Agile**.

**Contraintes**

CONTRAINTES TECHNIQUES : création d’une application web, accessible via internet.

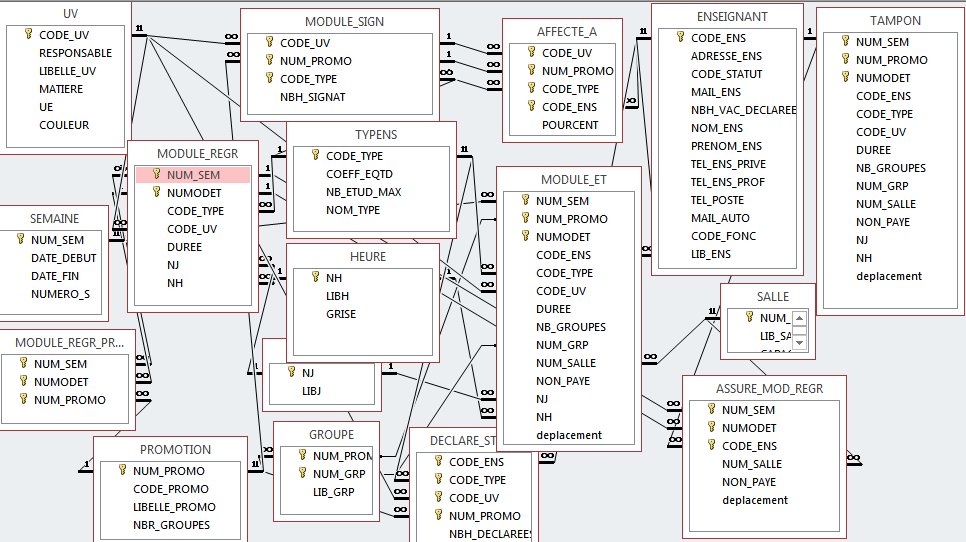
**Livrables**

* Fiche projet
* Backlog - fonctionnalités
* Application minimum viable
* Application finale
* Présentation, guide utilisateur ?

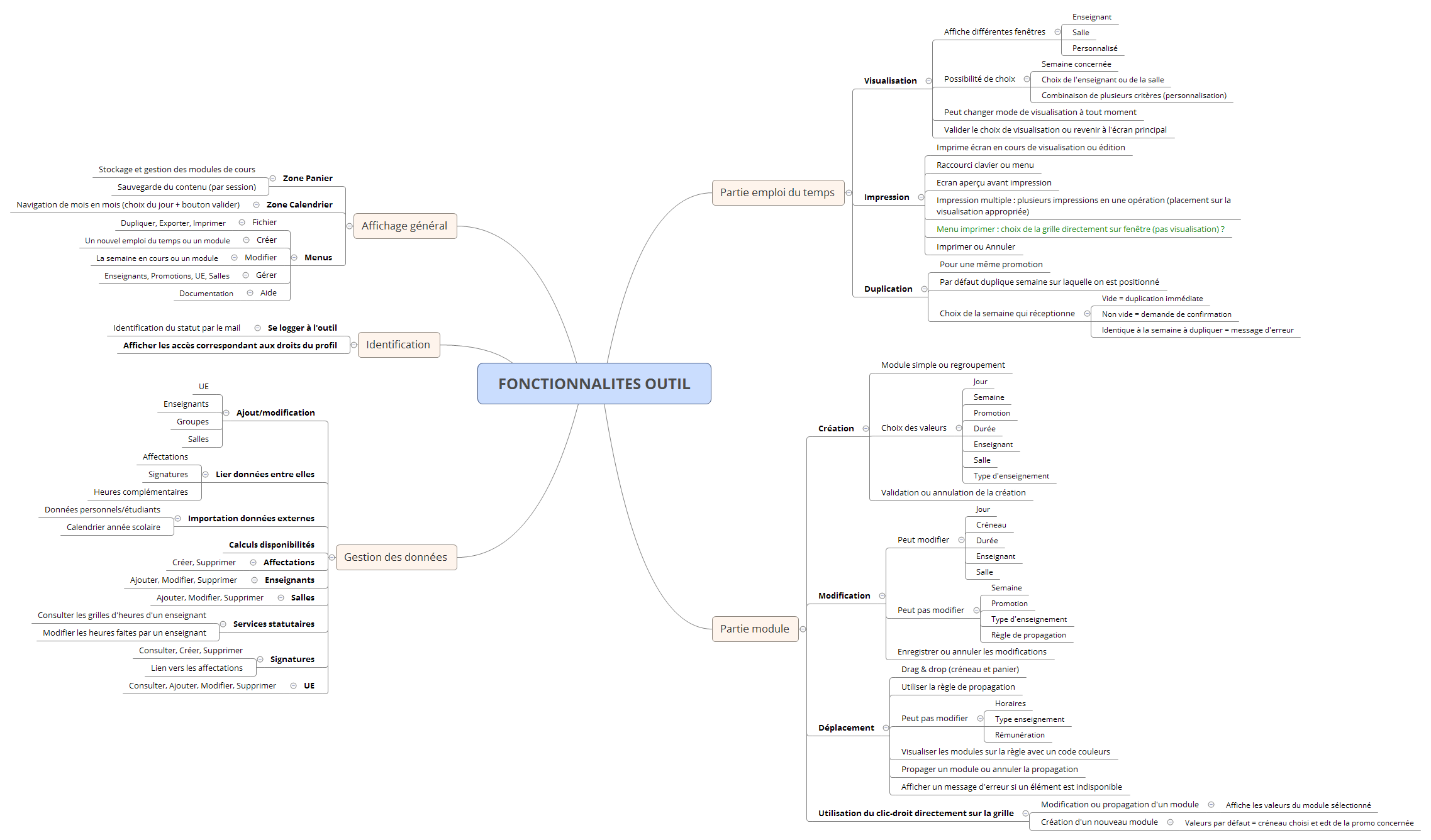
**Contacts**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NOM** | **PRENOM** | **QLT** | **FONCTION** | **TEL** | **MEL** |
| **MOA – Commanditaire** | | | | | |
| DKAKI | Taoufiq | M. | MCF, dir.adjoint DDAME, UT2J | 06 68 70 78 44 | Taoufiq.Dkaki@irit.fr |
| **MOE – Réalisation** | | | | | |
| BENYAHYA | Asmae | Mlle | Etd Master 2 I2N, UT2J | 07 68 55 95 16 | asmae.benyahya001@gmail.com |
| BOUHJAR | Tarik | M. | Etd Master 2 I2N, UT2J | 06 47 53 63 76 | tarik.bouhjar@gmail.com |
| CALMON | Anaïs | Mlle | Etd Master 2 I2N, UT2J | 06 26 81 06 64 | anaiscalmon31@gmail.com |
| MIRA | Abigail | Mme | Etd Master 2 I2N, UT2J | 06 61 37 10 18 | MIRACRICK@GMAIL.COM |
| OUEREGA | Waugoh | Mme | Etd Master 2 I2N, UT2J | 06 15 18 04 73 | reneelista225@gmail.comp |
| **Usagers Finaux** | | | | | |
| DKAKI | Taoufiq | M. | MCF, dir.adjoint DDAME, UT2J | 06 68 70 78 44 | Taoufiq.Dkaki@irit.fr |
| MOULY | Patricia | Mme | Resp. administrative DDAME, UT2J | 05 61 50 41 90 | dam@univ-tlse2.fr |
| BACK | Fanny | Mme | Gestionnaire scolarité, UT2J | 05 61 50 42 11 | gestionddame@univ-tlse2.fr |
| Responsables de formation | | | | | |
| Enseignants du DDAME | | | | | |
| Etudiants du DDAME | | | | | |

# Modèle du logiciel InfoCOM\_ET

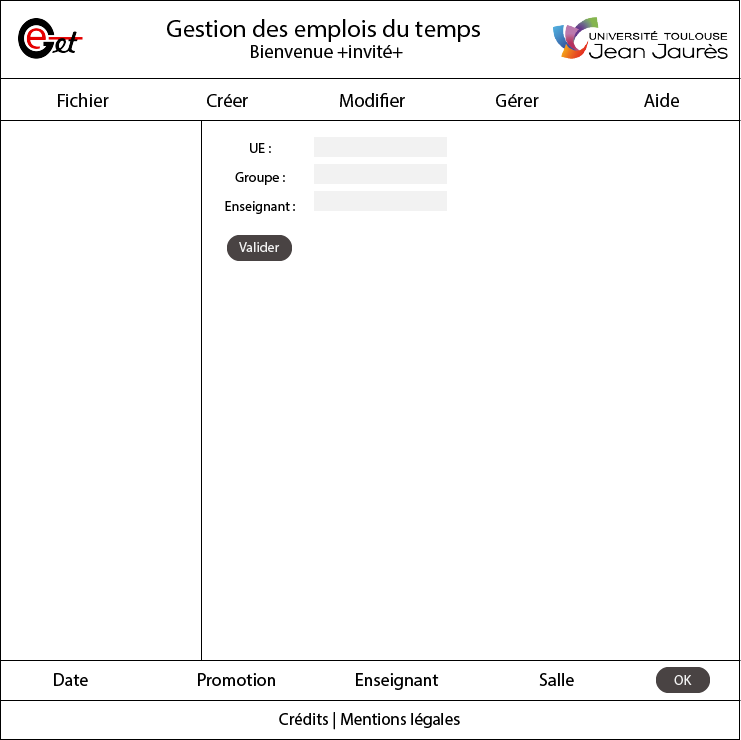


# Fonctionnalités extraites de la documentation d’InfoCOM\_ET

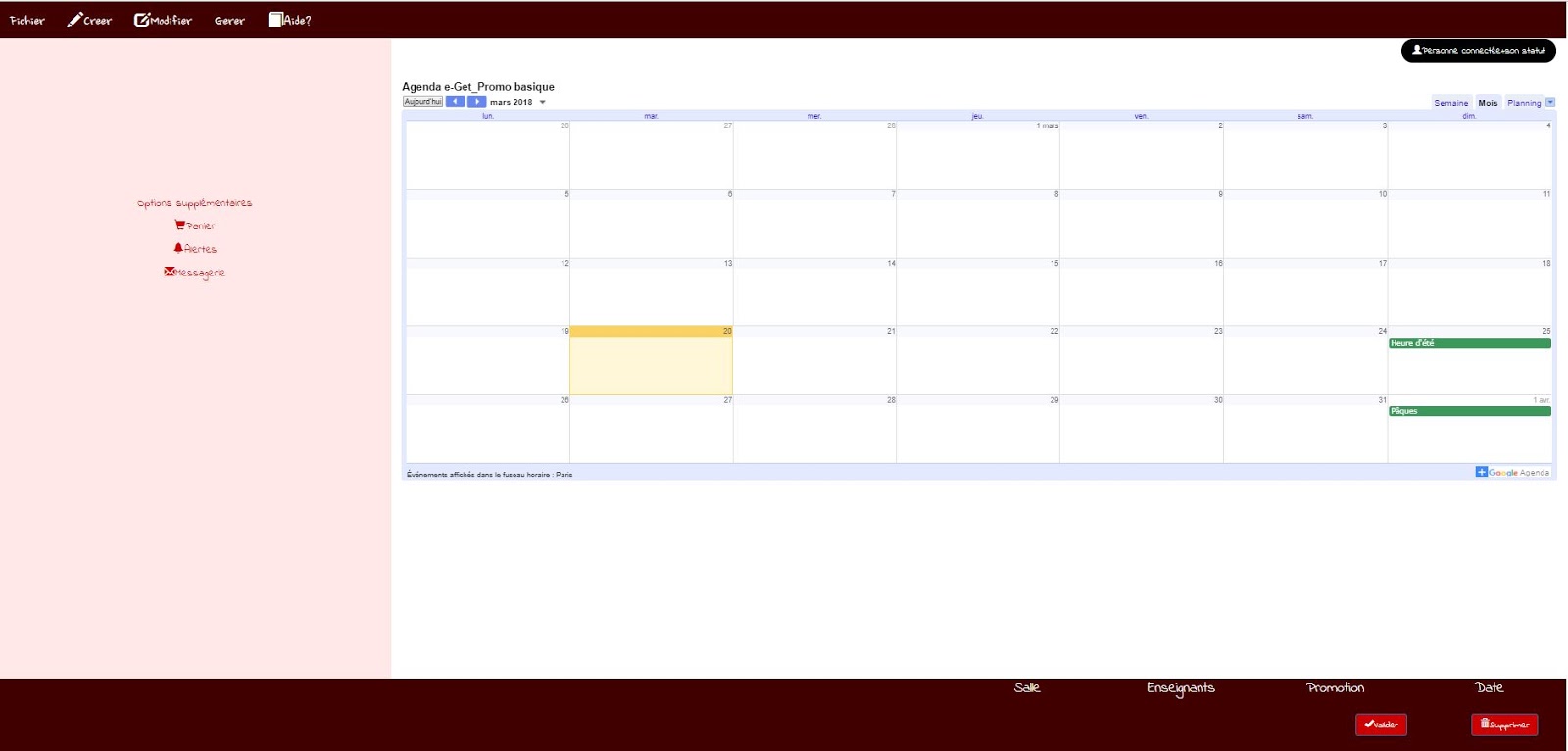


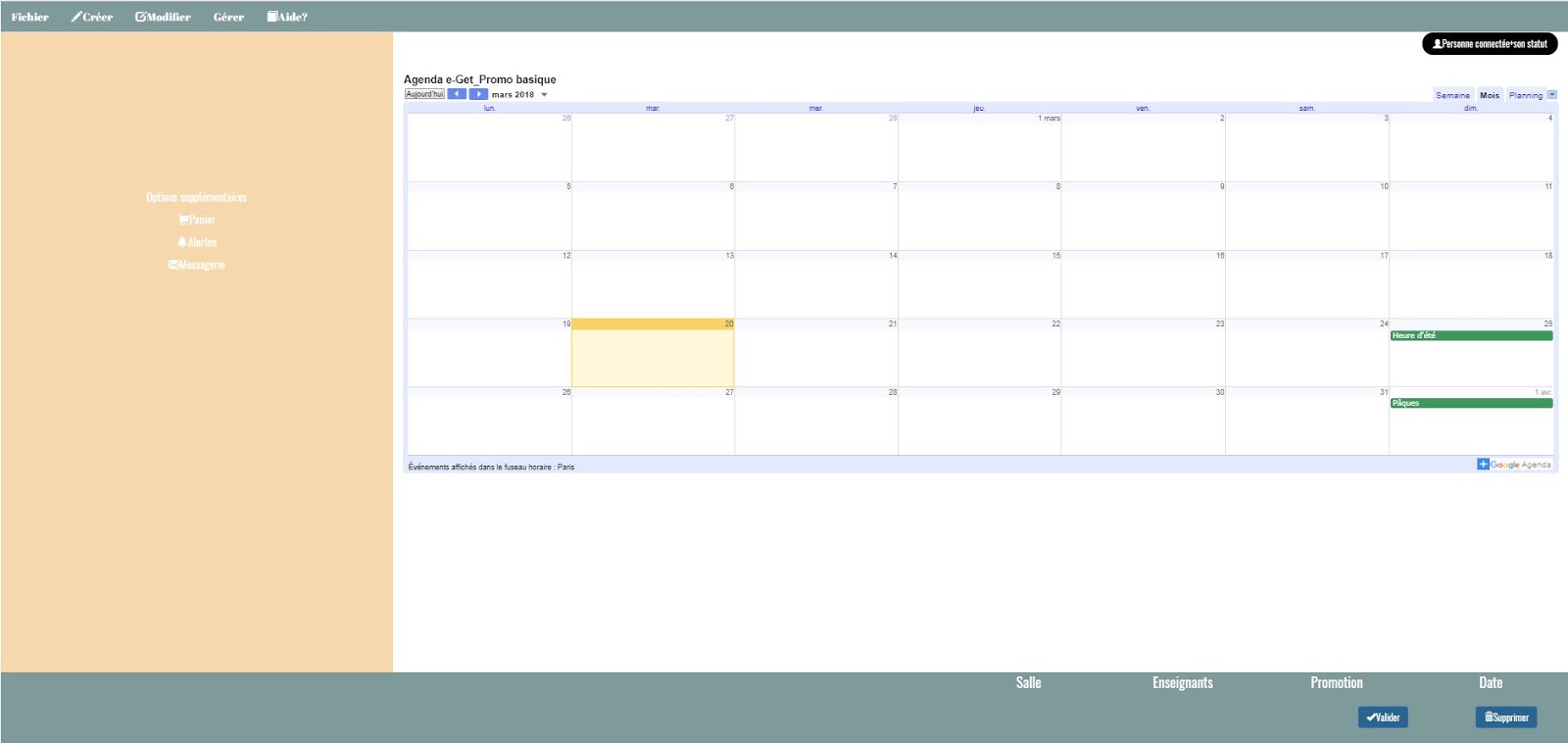
# Premières recherches et propositions graphiques

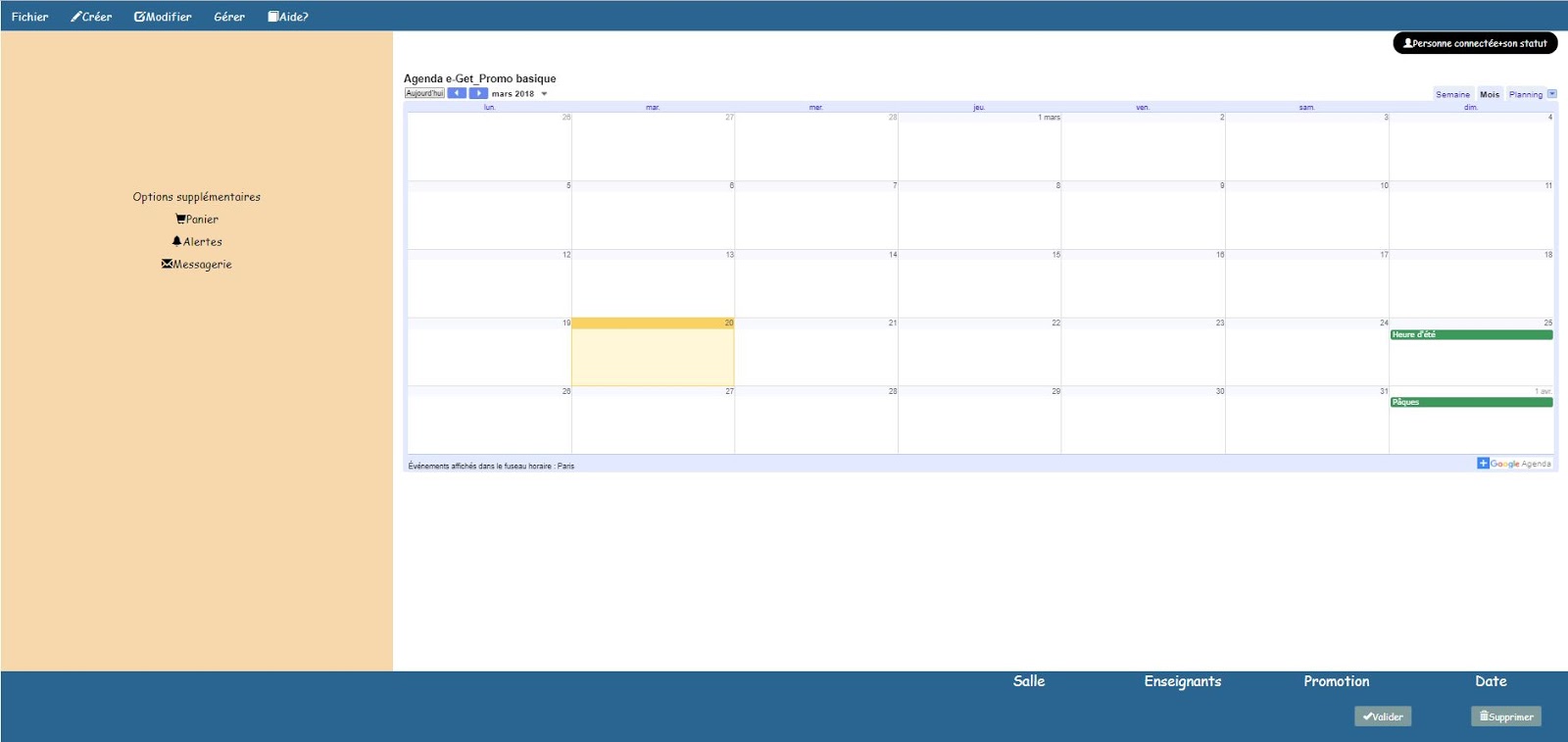
### Squelette de page :

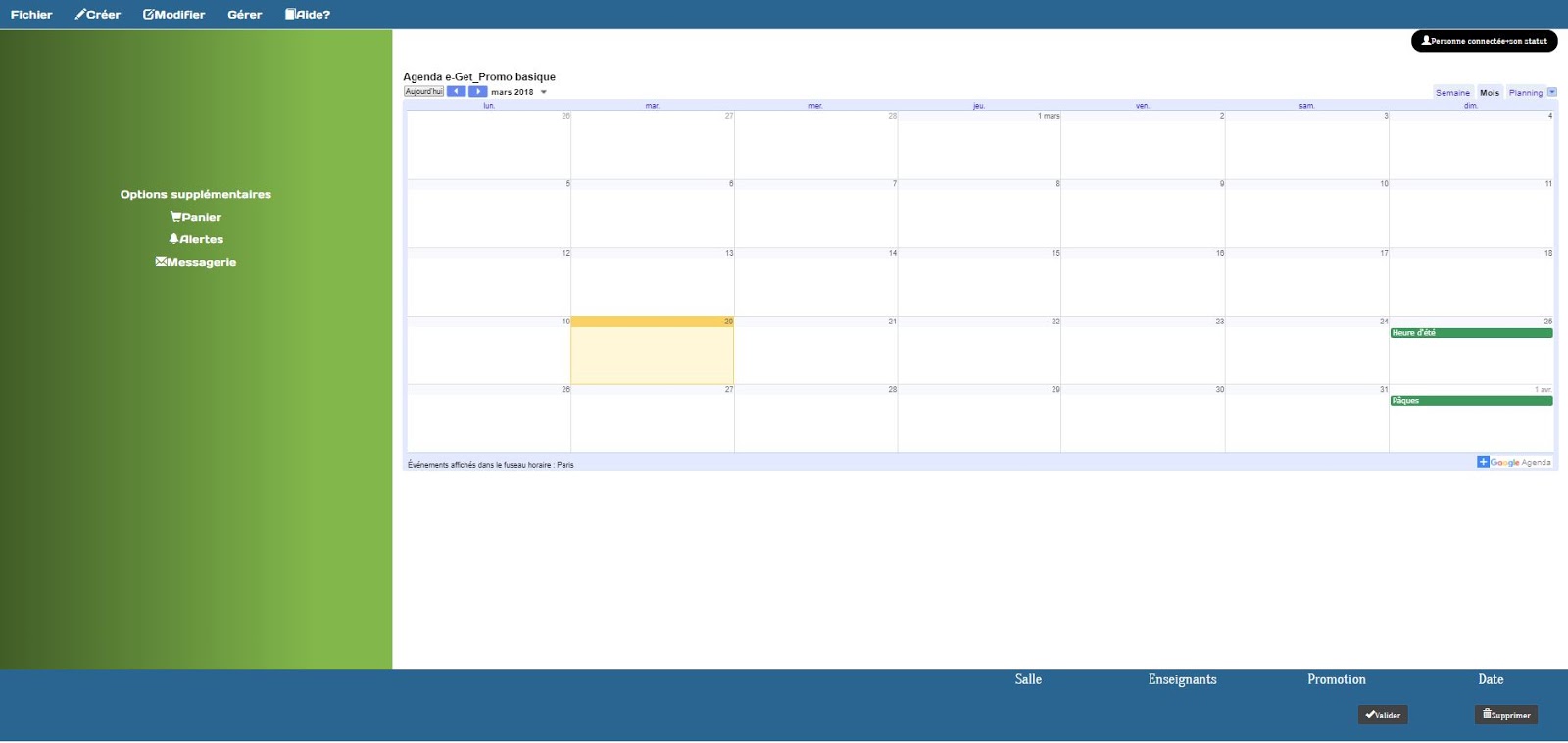


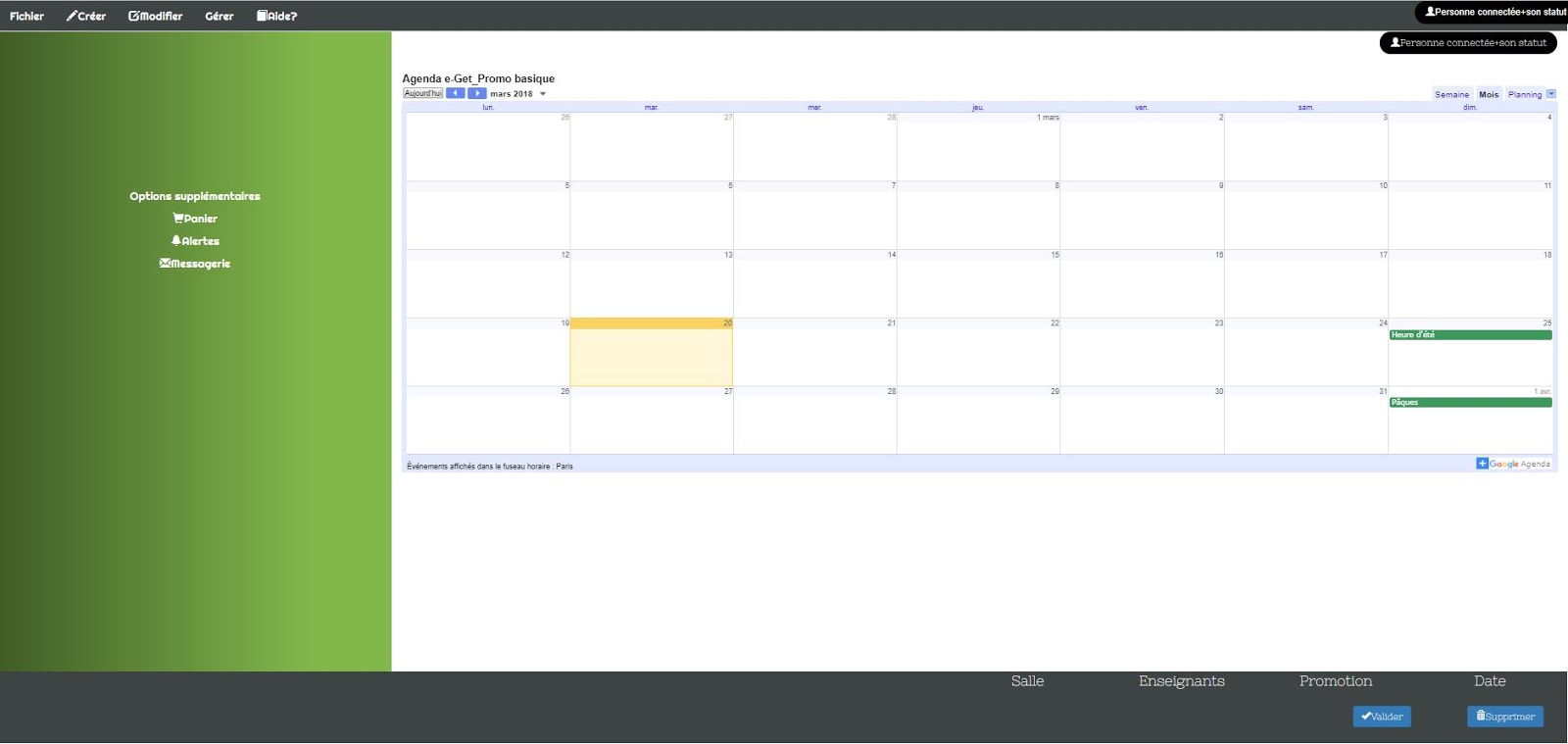
### Propositions page d’accueil :



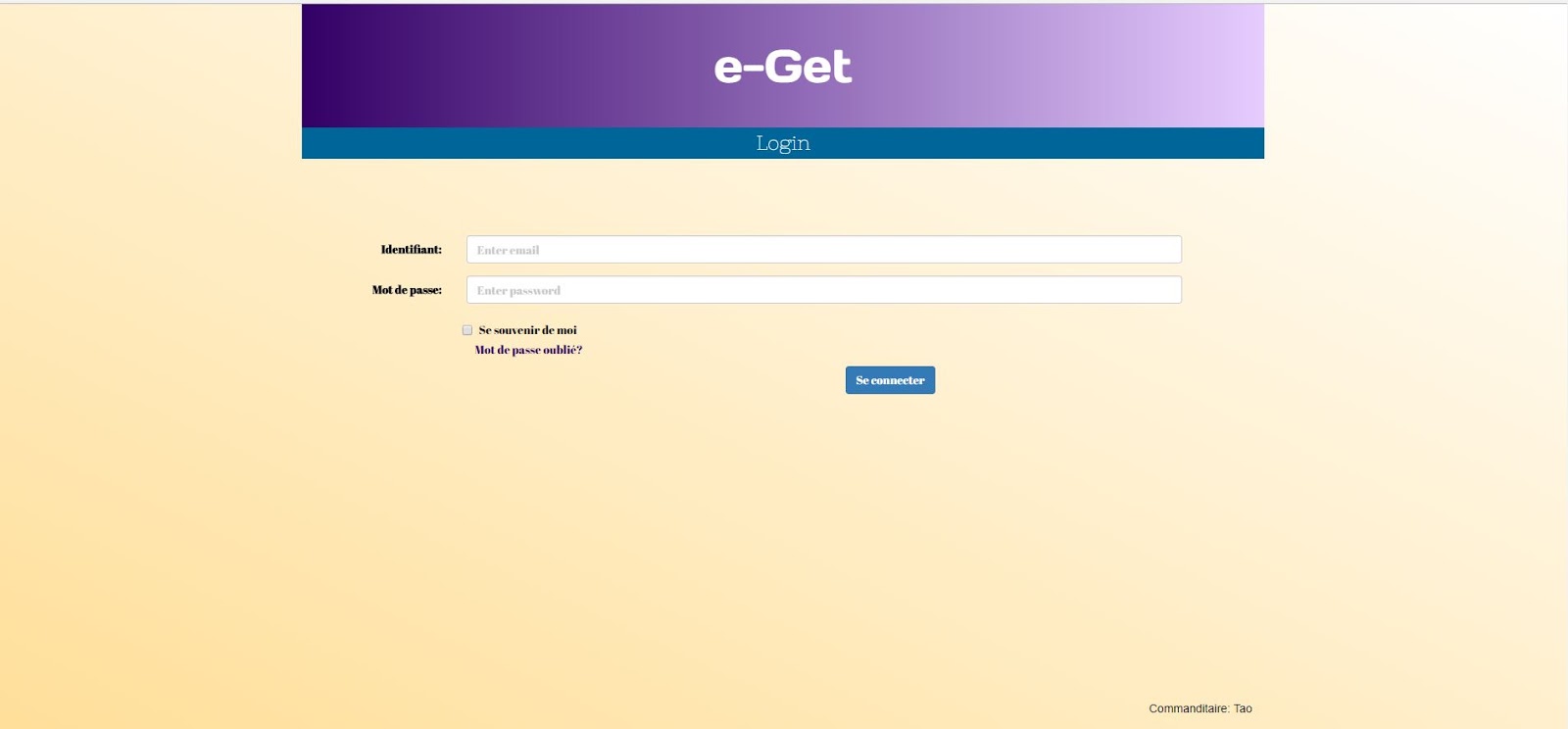


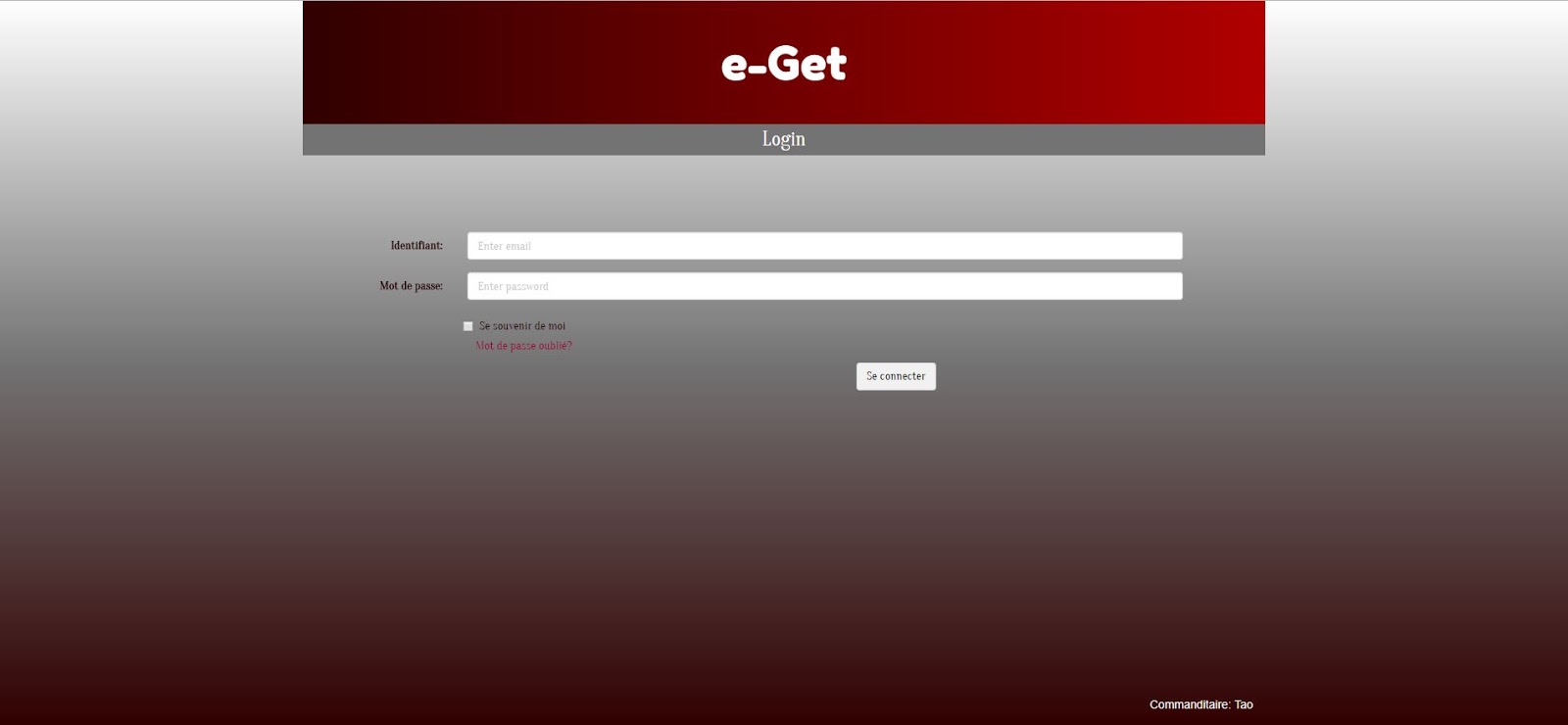


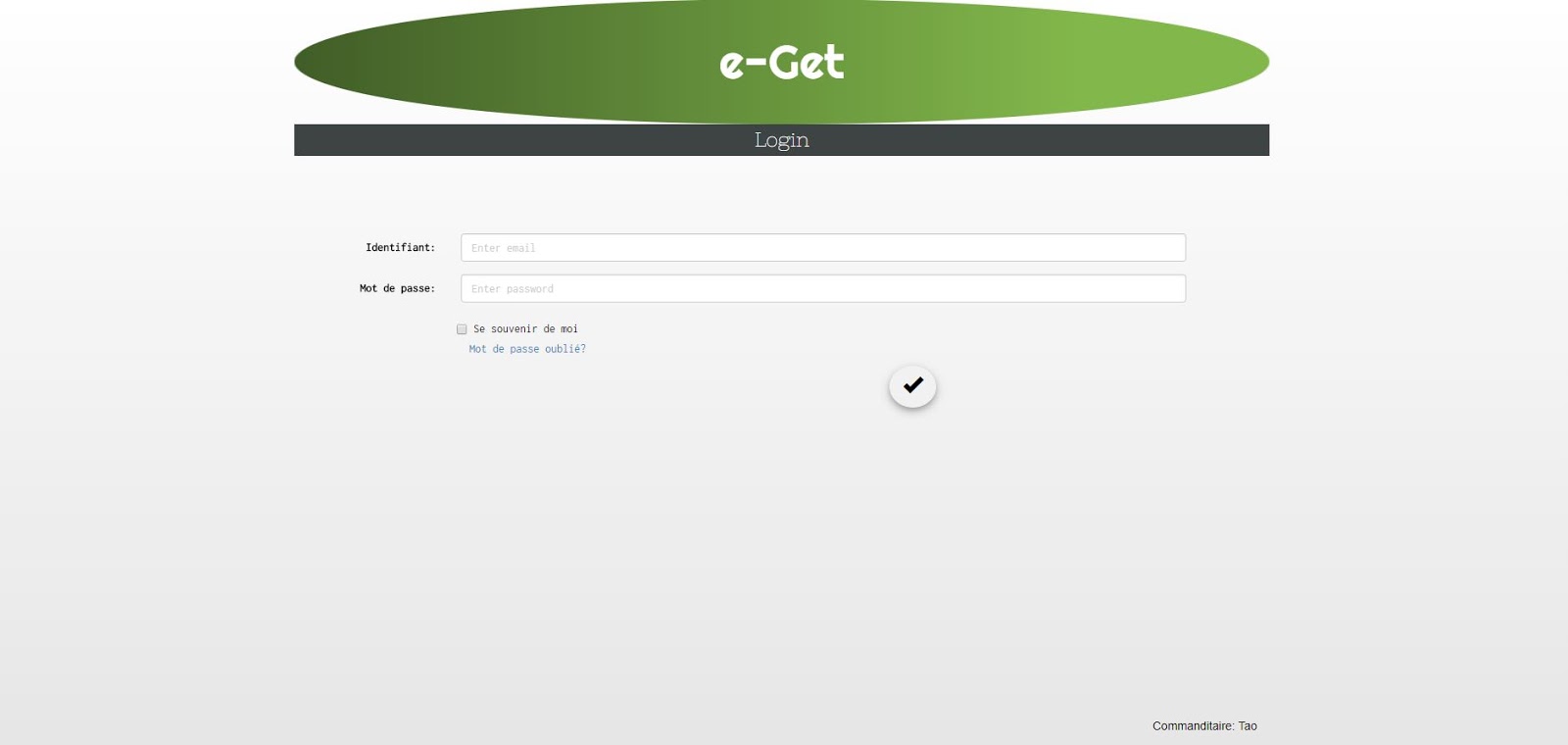


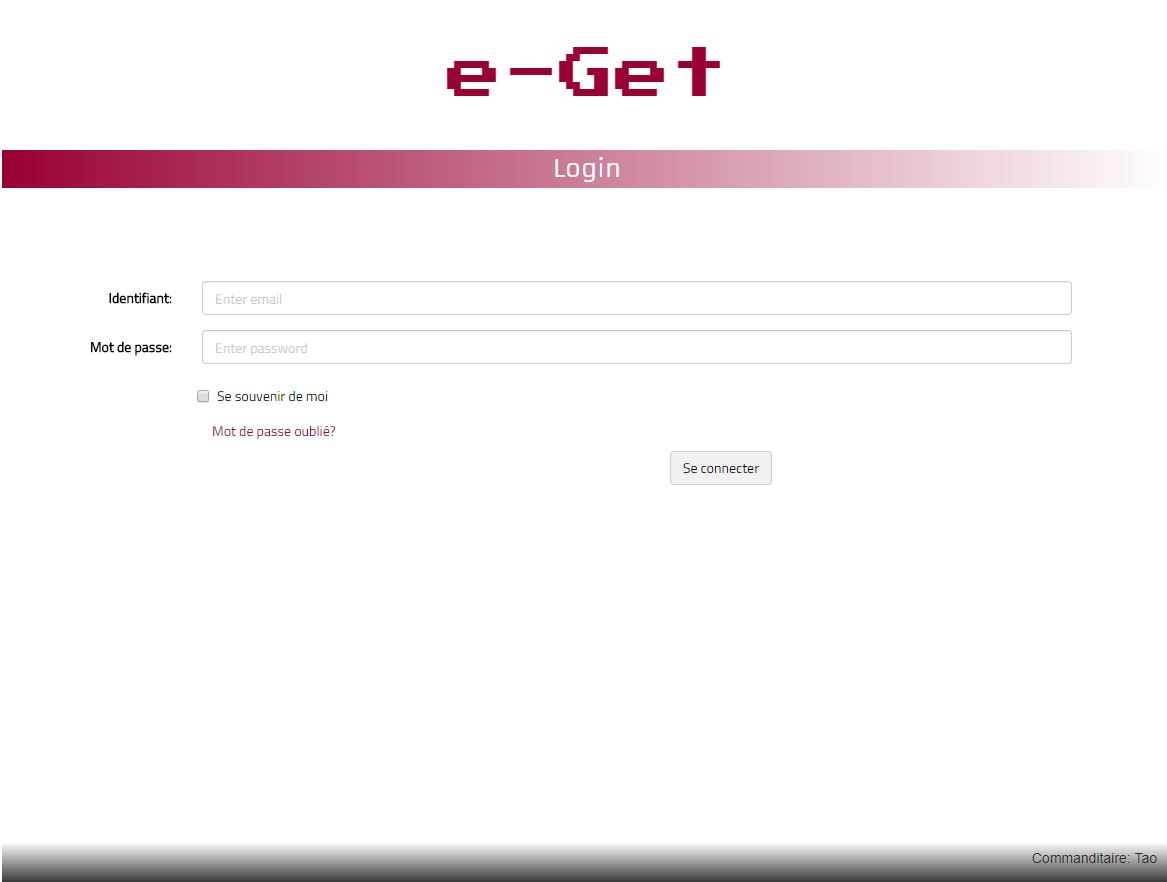


### Propositions pages login :









Projet e-Get – Charte graphique

# Langages utilisés

La structure de l’application est réalisée grâce au HTML, le positionnement et la personnalisation des éléments se font avec le CSS. Le langage *bootstrap* permet d’organiser et dimensionner les blocs de la page, et il intervient avec les *media queries* pour gérer la responsivité de l’application.

# Polices, tailles, couleurs des éléments

## Entête

Couleur de fond : #A4A4A4



Bordure inférieure de 4px, couleur : #54705F



Police : Mukta Mahee (importée de Google Fonts), police sans serif pour le remplacement ;

Taille : h1 ;

Couleur : #983535



Modifications : utilisation des petites majuscules et d’un espacement de 0.2em entre les lettres.





## Barre de menu et sous-menu

Police : Mukta Mahee (importée de Google Fonts), police sans serif pour le remplacement ;

Taille : 120%.



### Normal

Couleur de la cellule : #D8D8D8



Couleur de la police : #54705F

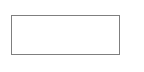


### Survol

Couleur de la cellule : #64AE82

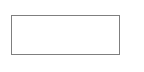


Couleur de la police : #FFF



### Sélection

Couleur de fond : #FFF



Couleur de la police : #54705F



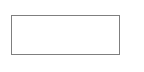
## Sous-Menus

### Normal

Couleur du fond : #54705F



Couleur de la police : #FFF

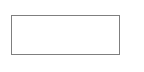


### Survol

Couleur du fond : #64AE82



Couleur de la police : #54705F



## Bienvenue

Police : Montserrat (importée de Google Fonts), police sans serif pour le remplacement ;

Taille : 1em.

Couleur : rgba(46, 46, 46, 0.5)





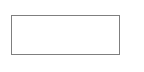
## Colonne de gauche

Liste non ordonnée, titre en h4.

Police : Montserrat (importée de Google Fonts), police sans serif pour le remplacement ;

Hauteur de ligne : 25px.

Couleur du fond : #FFF



Couleur de la police : #A75355



## Section principale

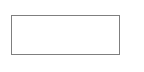
Police : Montserrat (importée de Google Fonts), police sans serif pour le remplacement ;

Titre en h3, sous-titre en h4.

Style des boutons grâce à *bootstrap*, taille small : btn-success, btn-primary et btn-default.



Couleur du fond : #FFF



Couleur de la police : #000





### Lignes <hr>

Taille : 50%

## Pied de page

Couleur du fond : #D8D8D8



Bordure supérieure de 3px, couleur : #54705F



Police : Mukta Mahee (importée de Google Fonts), police sans serif pour le remplacement ;



Couleur de la police : #54705F

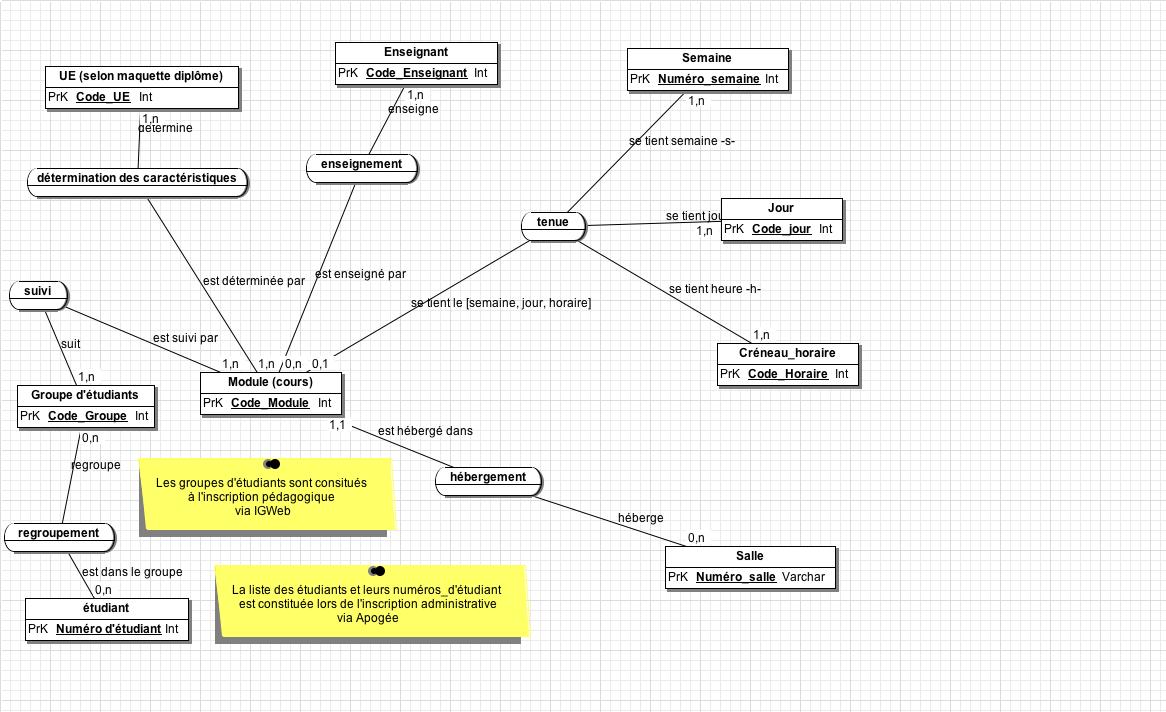




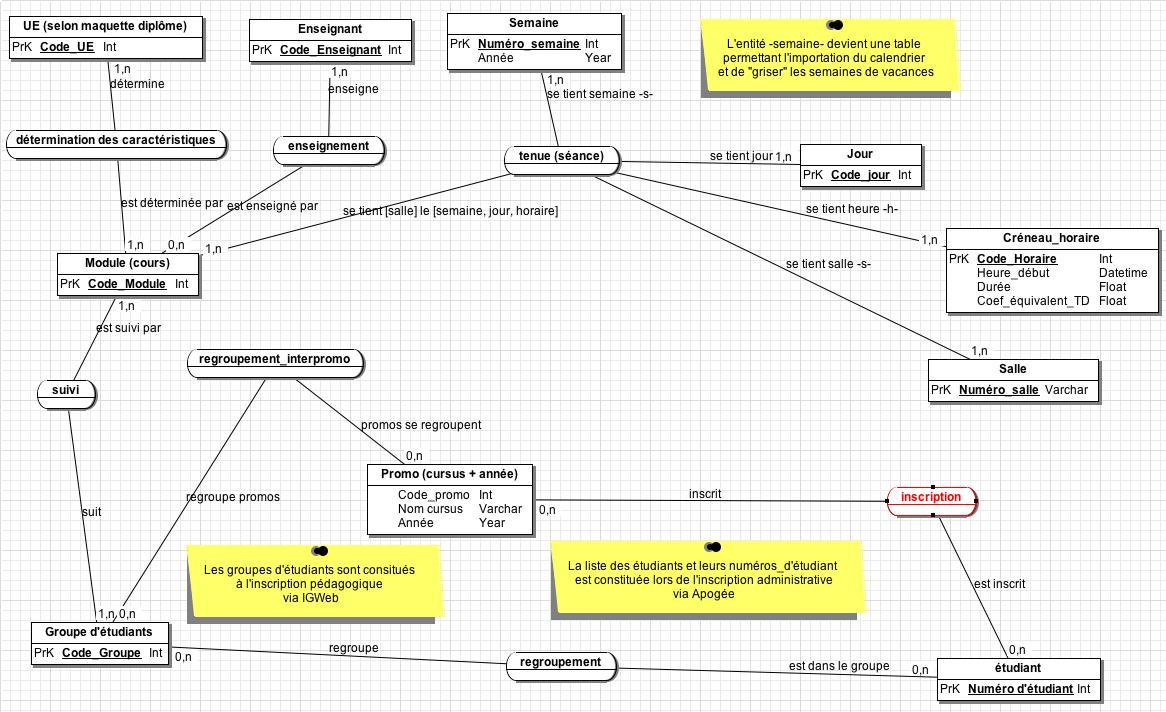
Capture d’écran d’une page de l’application

# Évolutions du modèle

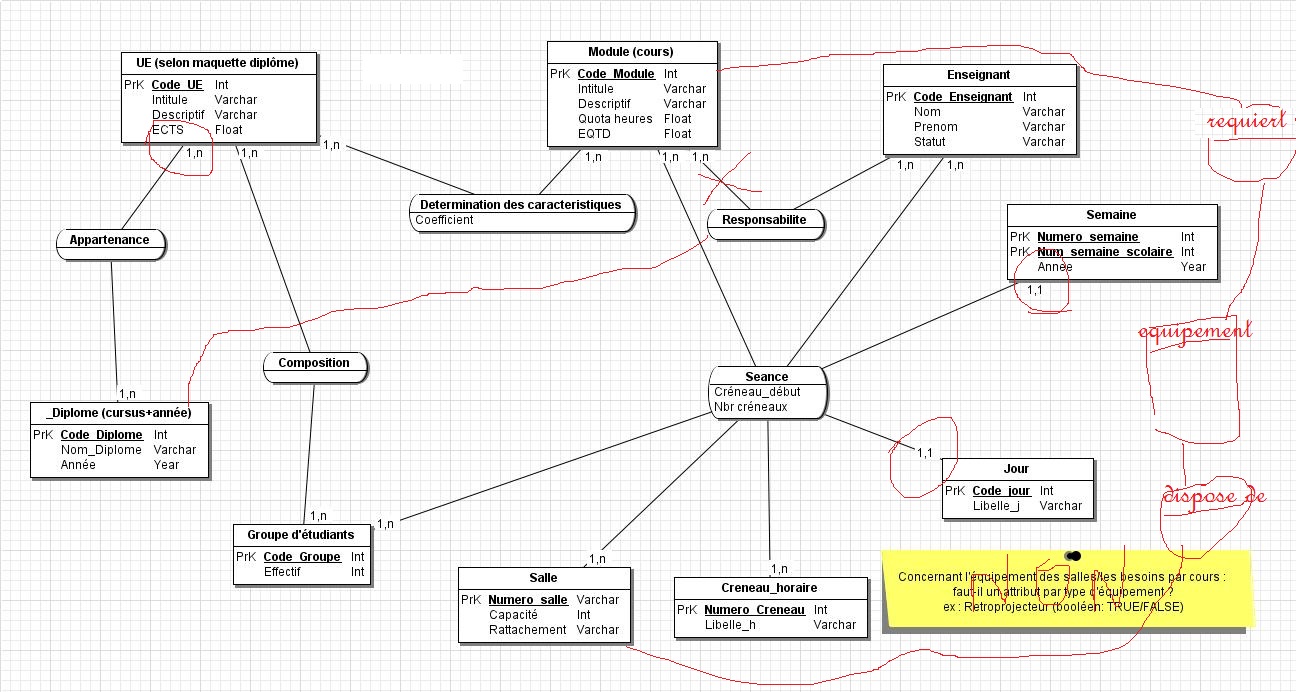
## Version 1



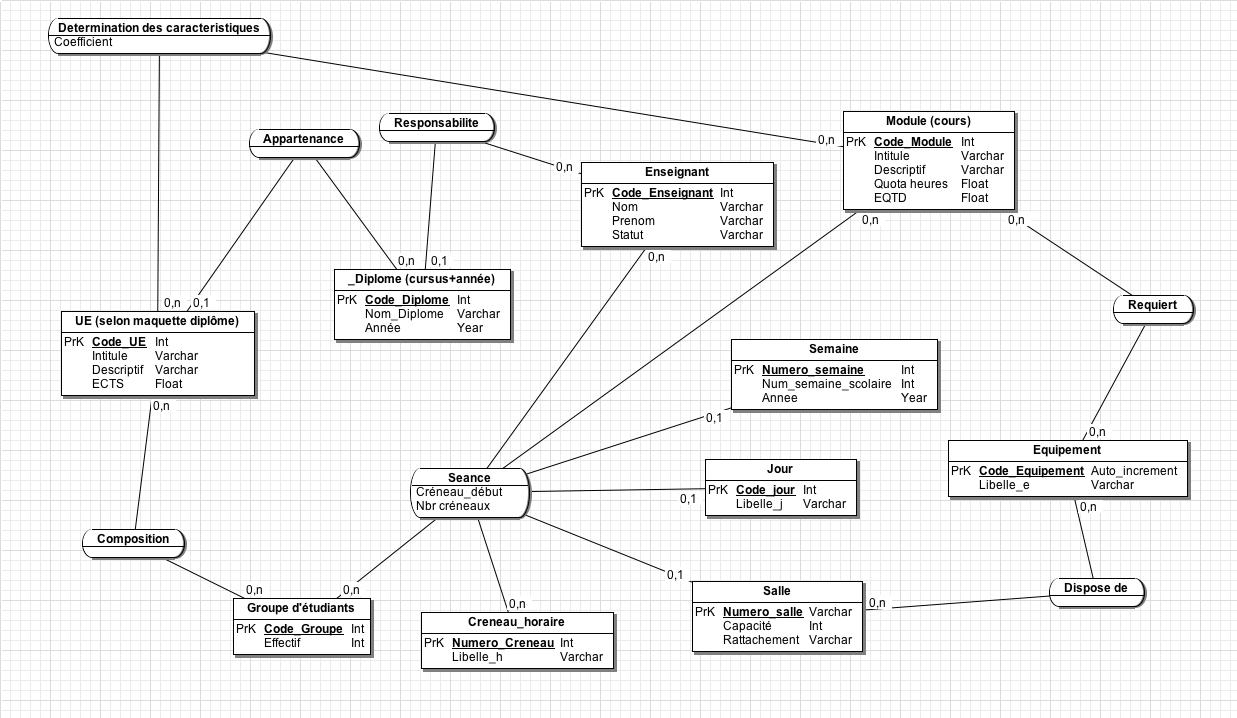
## Version 3



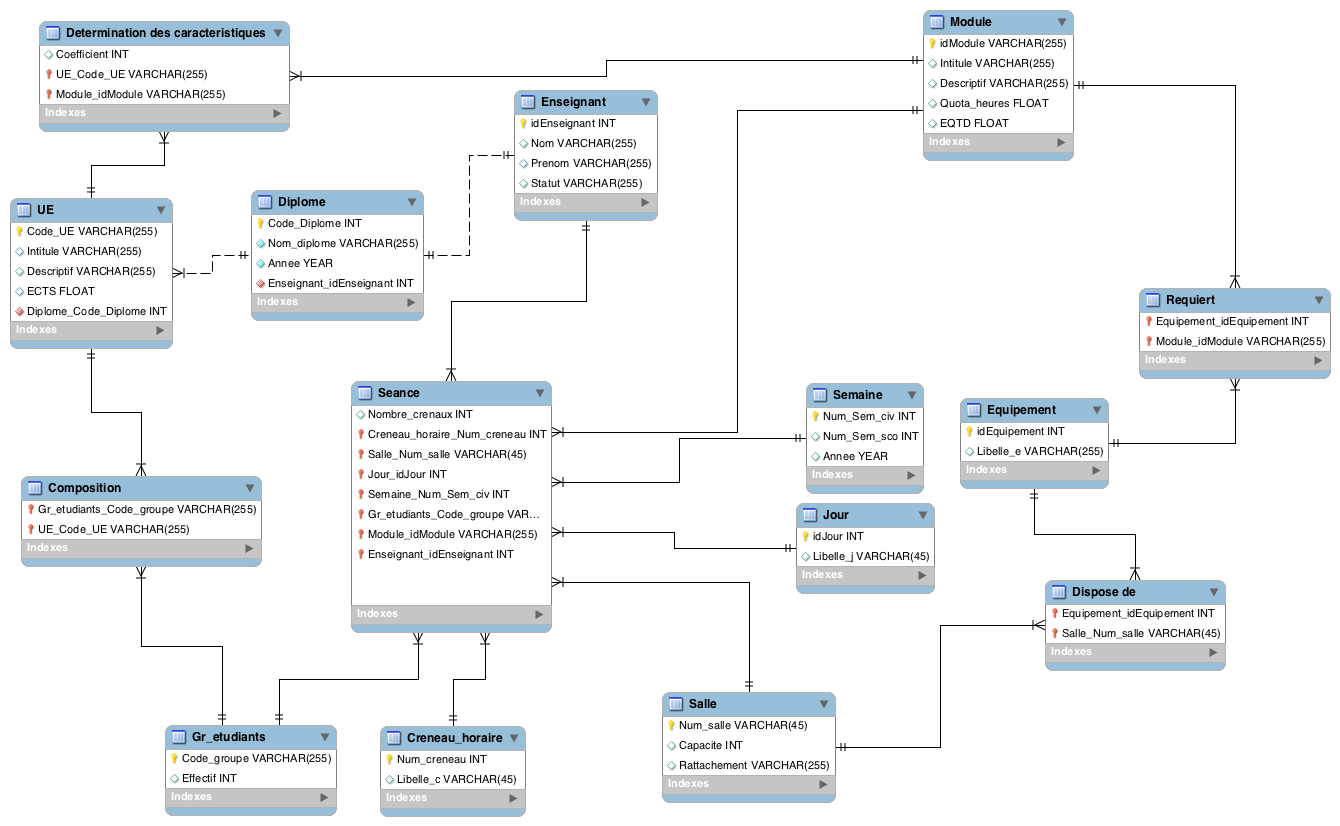
## Version 5 (commentée par Mr Taoufiq Dkaki)



## Version finale (JMerise)

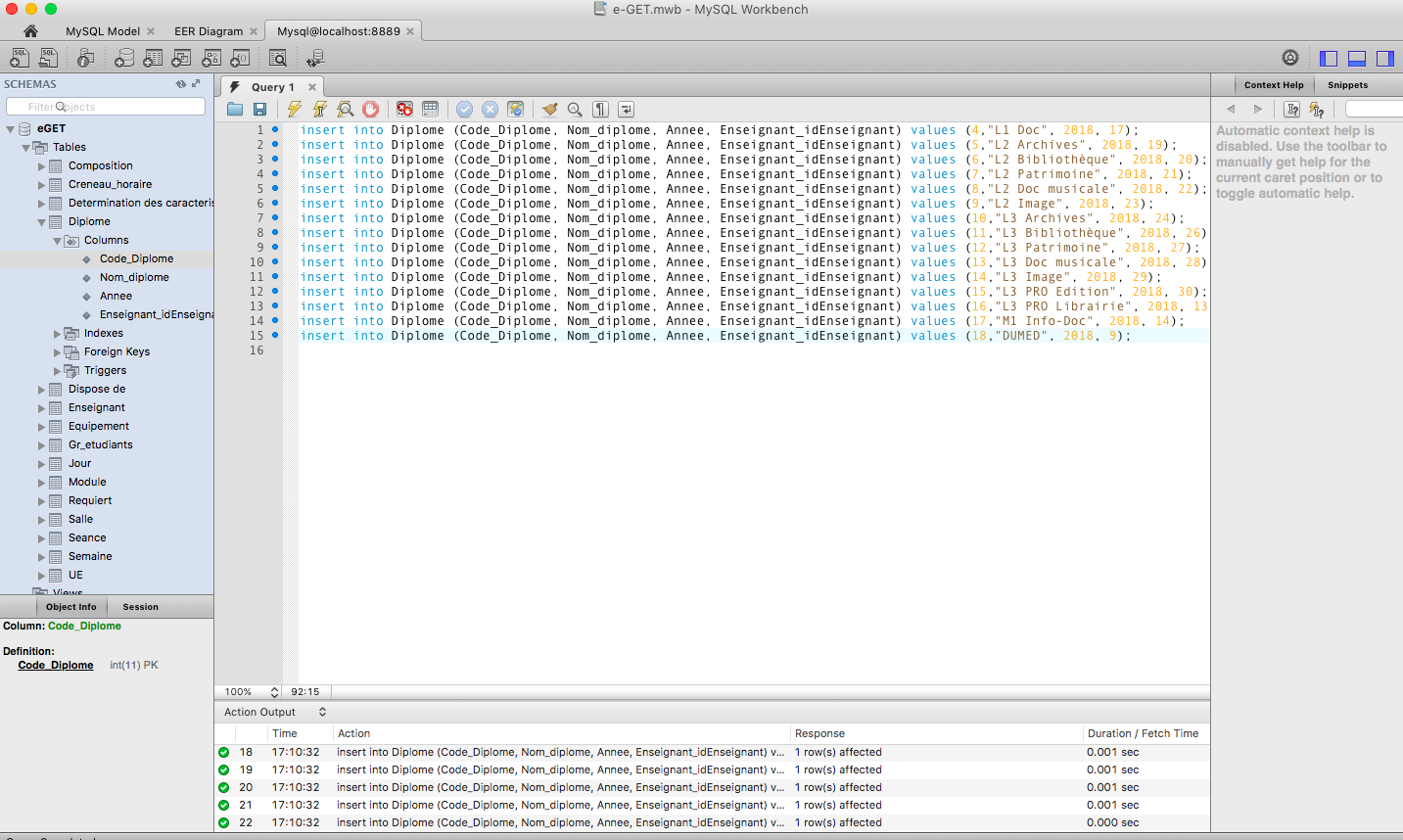


## Version complète (MySQL Workbench)

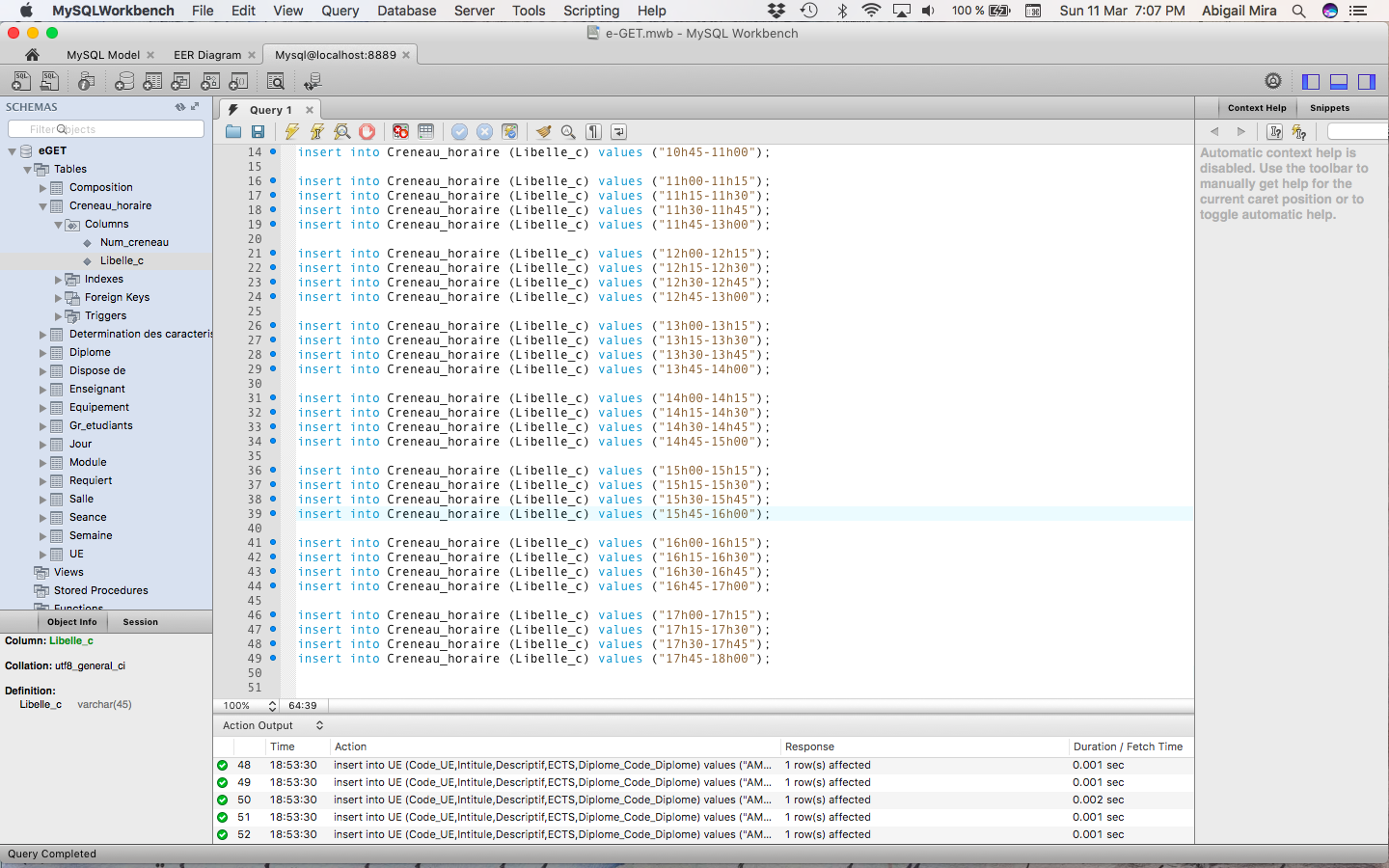


# Travail sur la Base de données (MySQL Workbench)

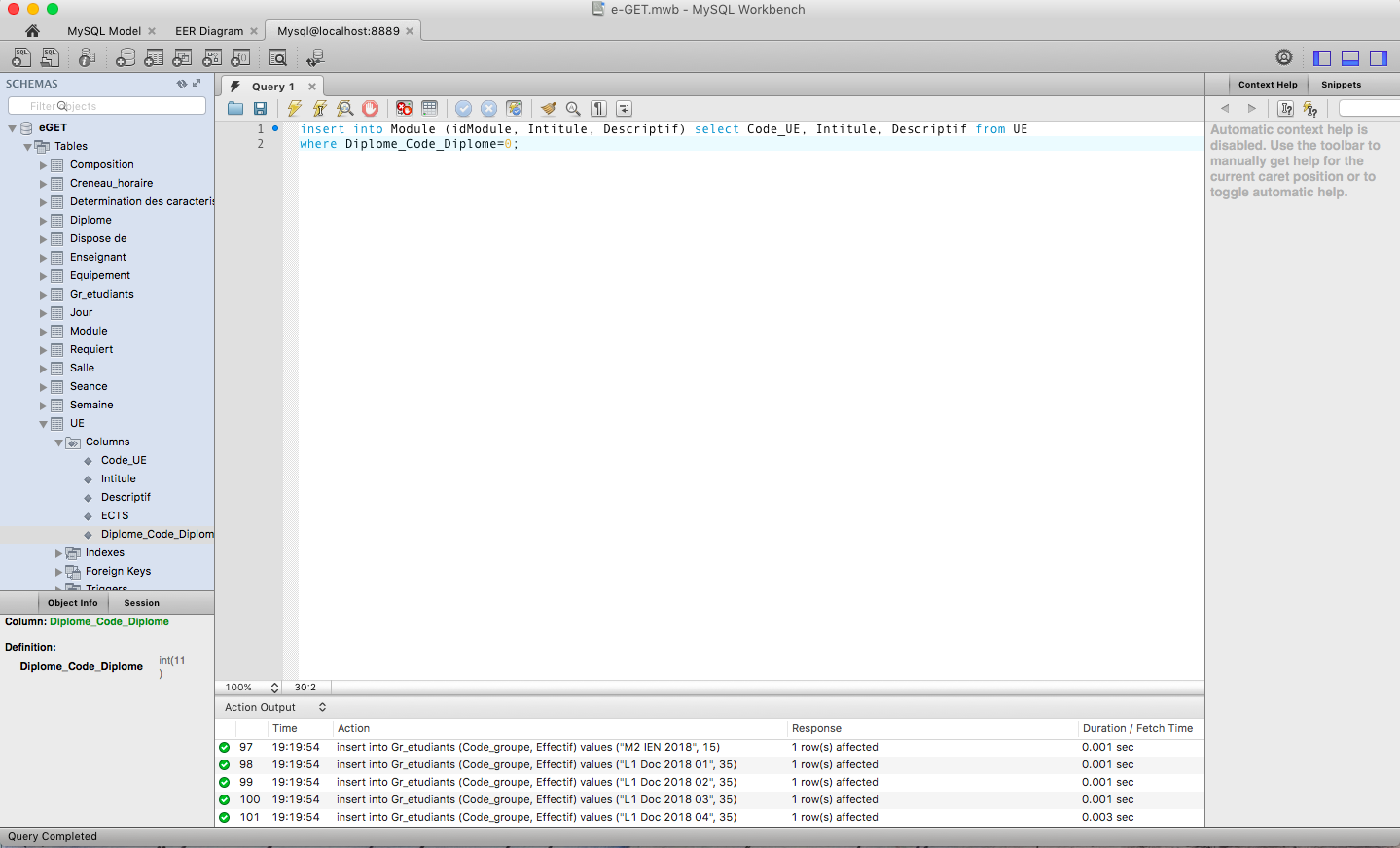
## Image 1



## Image 2

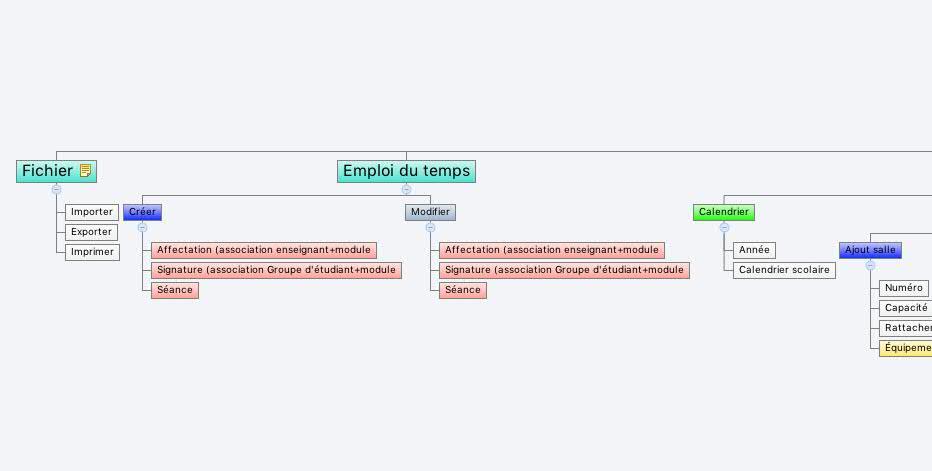


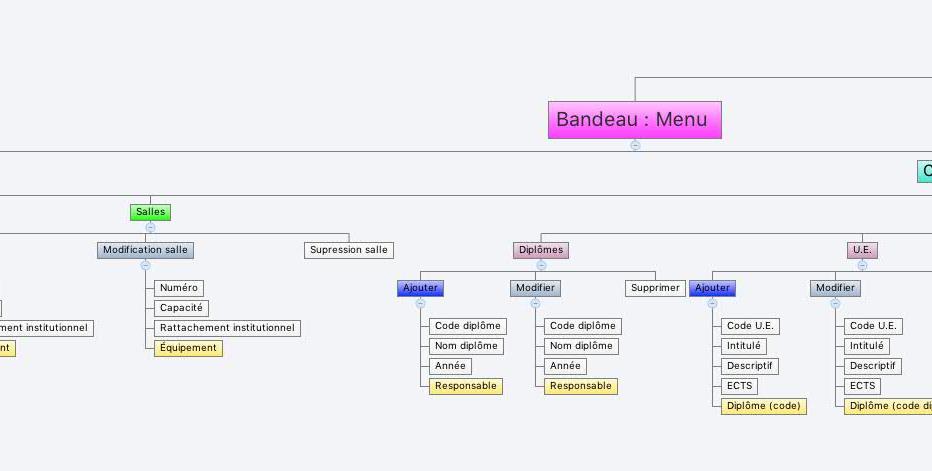
## Image 3

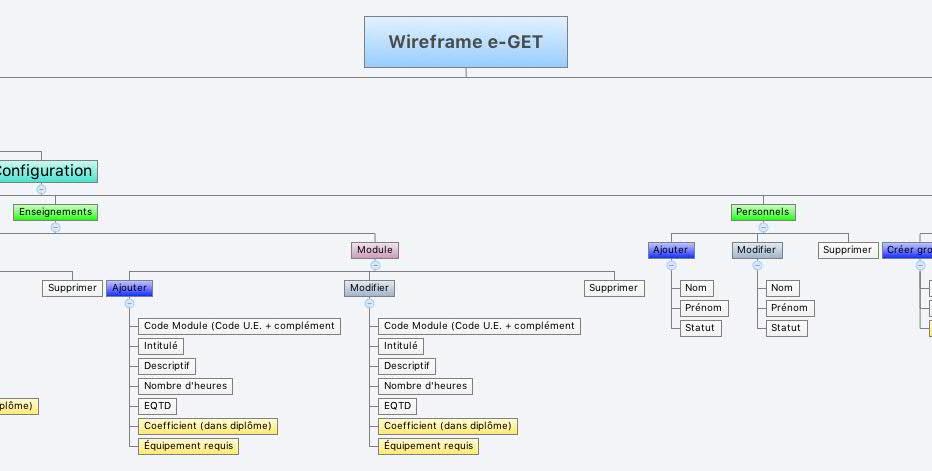


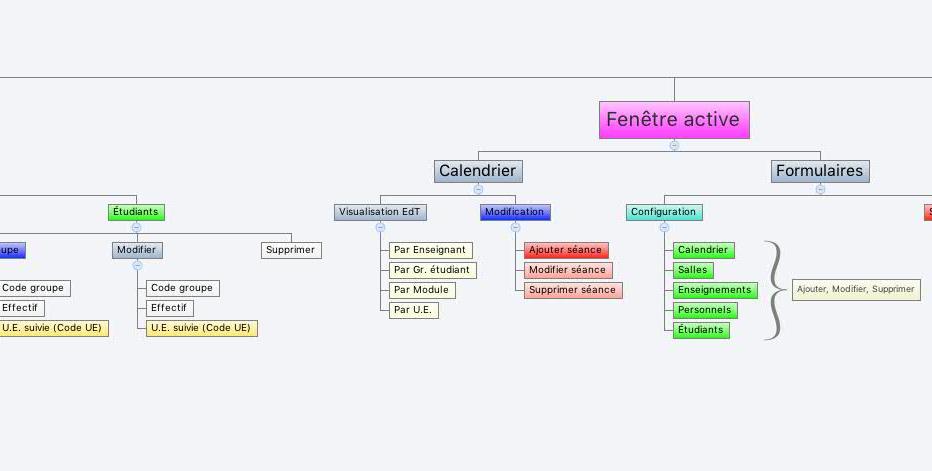
# Plan de l’application

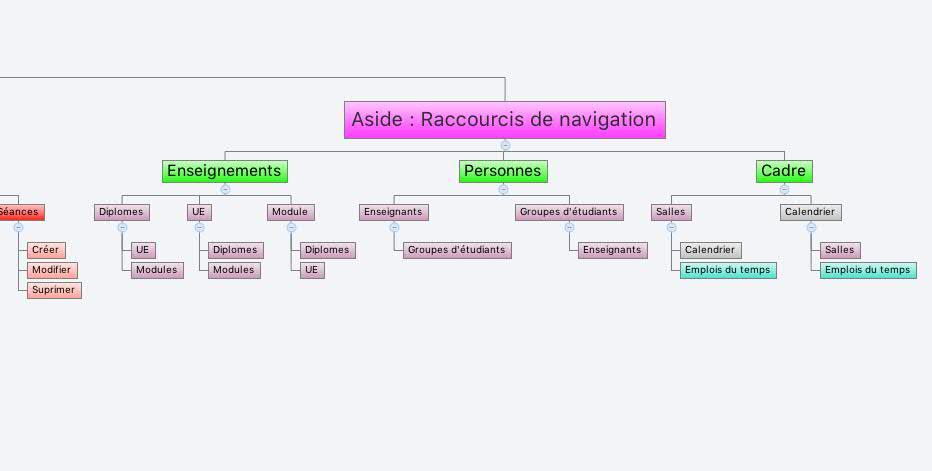
## Statique











1. <http://agilemanifesto.org/iso/fr/manifesto.html>. Site consulté le 20 mars 2018. [↑](#footnote-ref-1)