****

Institut Supérieur Polytechnique de

Madagascar

ETech

Mémoire de fin d’études pour l’obtention du diplôme de Master en informatique

Option : Informatique de Gestion Génie Logiciel et Intelligence Artificielle



**Présenté par:**

Monsieur RAJAONARISON Maminiaina

**Membres du jury :**

* *Président du jury* : Professeur RABOANARY Julien Amédée
* *Encadreur professionnel* : Madame RABETOKOTANY Rota
* *Encadreur pédagogique* : Monsieur RASANDIMANANA Tanjona Tsioharana

*Année Universitaire 2014-2015*

« Le cœur de l’Homme peut méditer sa voie, mais c’est l’Eternel qui dirige ses pas. »

*Proverbes 16 :9*

REMERCIEMENTS

Nous tenons à remercier ici tous ceux qui ont œuvré pour la réalisation de ce mémoire d’étude en vue d’obtenir notre diplôme de licence en informatique de gestion génie logiciel intelligence artificielle, entre autres :

* Rendons grâce à Dieu Tout Puissant pour la force, la santé et le savoir qu’il nous a offerts tout au long de ce stage ;
* Le Professeur RABOANARY Julien Amédée, Recteur de l’Institut Supérieur Polytechnique de Madagascar, qui nous a permis d’entreprendre les études à l’ISPM ;
* Monsieur Paul Pierre ARDILE, le Président Directeur Général de la société eTech, grâce à qui nous avons pu nous imprégner dans une vie d’entreprise ;
* Monsieur Andry RAMANAMBAHOAKA, Directeur de projet Web-PHP ;
* Monsieur Clarel RAKOTONDRAHAJA, Directeur de projet et aussi enseignant à l'Institut Supérieur Polytechnique de Madagascar ;
* Monsieur Rija RAOBIVELO RABETOKOTANY, Directeur de projet (.Net – Java - Mobile) ;
* Madame Eva RALALAHARIMANANA, Responsable Administratif et Financier ;
* Madame RABETOKOTANY Rota, chef de projet Web-PHP et aussi notre encadreur professionnel, pour les conseils professionnels ;
* Monsieur RASANDIMANANA Tanjona Tsioharana, notre encadreur, qui a fait de son mieux pour nous enseigner la vie d’entreprise et ce que pourrait subvenir à l’avenir ;
* Le corps administratif et professoral que nous devons notre savoir sur l’informatique ;
* Tout le personnel de la société eTech qui nous ont accueilli à bras ouvert et font leur possible pour nous aider ;
* Toute la famille pour leur soutien moral et financier ;
* Tous nos amis et proches qui ont participé de près ou de loin à la réalisation de ce mémoire.

A tous, un grand merci !

LISTE DES ABREVIATIONS

LISTE DES FIGURES

LISTE DES TABLEAUX

SOMMAIRE

Bibliographie

INTRODUCTION

Les premiers réseaux de télécommunication ont conduit à la création d'un réseau informatique. Ce dernier permet aux utilisateurs de différents ordinateurs de communiquer, puis se développa par de nombreuses étapes successives. La somme de tous ces développements conduit au "réseau des réseaux" que nous connaissons aujourd'hui en tant qu'***Internet***.

Vers les années 1989, Tim Berners-Lee, un physicien britannique, a inventé le web au CERN. Au début, le projet « World Wide Web », connu aussi sous l’appellation WWW, a été conçu et développé pour que des scientifiques travaillant dans les universités et les instituts du monde entier puissent s'échanger des informations instantanément. Mais le domaine du web s’épanouit au fil du temps, d’où la mise en ligne du premier site web le 20 décembre 1990. De nos jours, le monde du web prend une très grande place [a] [B] [C].

De ce fait, notre thème de mémoire se portera justement sur la création d’un site web qui s’intitulera *« Groupe SANTE ».* Ce site web aura pour objectif principal de regrouper en un seul site web tous les établissements de santé de Madagascar. Chaque établissement est relié à un projet. Ce dernier peut aussi appartenir à plusieurs établissements. Les utilisateurs inscrits peuvent participer ou bénéficier des projets existants.

Nous nous permettons de prime abord de vous présenter l’*ISPM* notre Institut. En second lieu, la société *eTech* qui nous a pris en charge pendant ces quelques mois avant d’exposer point par point le projet que nous avons opté pour ce stage.

# PRESENTATION DU CADRE D’ETUDE ET DE STAGE

### PRESENTATION DE L’ISPM

#### Historique et présentation de l’ISPM

##### Historique de l’ISPM

Connu auparavant par E.S.S.T.I.M (Ecole Supérieur des Sciences et Techniques de l’Information à Madagascar), cet institut naît le 23 Janvier 1993 selon l’arrêté N°3725/94 du Ministère de l’Enseignement Supérieur et accueille quelque dizaines d’étudiants. A cette époque, son siège se trouvait à Ankadindramamy, il était connu sous le nom d’ESSTIM ou École Supérieure des Sciences et de la Technologie Informatique à Madagascar et ne comportait que le département Informatique: la filière Informatique de Gestion génie Logiciel et Intelligence Artificiel ou I.G.G.L.I.A et la filière Maintenance des Systèmes Informatiques ou M.S.I C’était le seul institut privé polytechnique à Madagascar. Son siège se trouvait à Ankadindramamy.

Figure 1: E.S.S.T.I.M (source ISPM)

Au fil du temps, E.S.S.T.I.M devient I.S.P.M (Institut Supérieur Polytechnique de Madagascar) et compte des milliers d’étudiants actuellement. L’ISPM et son bureau administratif se trouve à Ambatomaro Antsobolo.



Figure 2: ISPM Ambatomaro Antsobolo (source ISPM)

I.S.P.M a été fondé par RABOANARY Julien Amédée et est aujourd’hui l’un des plus grands Instituts privés de Madagascar. L’école est bâtie sur plus de 5 500 m² de surface sur un terrain de trois hectares. Avec ses soixantaines de salles de classes, ses terrains et ses cafétérias, I.S.P.M fournit à ses étudiants un environnement adéquat pour la préparation de leur avenir.

Au cours du temps, plusieurs départements et filière sont nés:

* 1994: Création du Département Informatique avec les deux filières: I.G.G.L.I.A et E.S.I.I.A;
* 1995: Création du Département Biotechnologique comportant deux filières: I.A.A et P.I.P;
* 1996: Création du Département tertiaire avec la filière C.A.A;
* 1998: Création du Département Génie Civil et Génie Industriel et aussi de la filière E.M.I.I;
* 1999: Mise en place de la filière G.C.A;
* 2000: Ouverture de la formation en troisième cycle;
* 2004: Création et ajout de la filière F.I.C dans le Département tertiaire;
* 2004: Intégration des deux filières: I.M.T.I.C.I.A dans le Département Informatique et A.E.E dans le Département Biotechnologie;
* 2009: Ajout des filières D.T.J.A, E.M.P et I.C.M.P;
* 2009: Tourisme et hôtellerie dans le Département Technique Tourisme;
* 2010: Création de la nouvelle filière I.S.A.I.A.

##### Présentation de l’ISPM

###### Les différentes mentions

Les filières de l’ISPM sont réparties à travers six mentions, à savoir la mention informatique et télécommunication, la mention droit et techniques des affaires, la mention biotechnologie et agronomie, la mention génie industriel, la mention génie civil et enfin la mention techniques du tourisme composés par des filières réparties comme suit :

###### Etudes et Formations

L’entrée à l’ISPM se fait par voie de concours. L’inscription à ce dernier exige aux candidats le Diplôme de Baccalauréat de l’Enseignement Secondaire.

Le cursus comprend deux cycles consécutifs dont :

* Le premier, d’une durée de trois ans, est sanctionné par une Licence ou BACC+3. Afin d’obtenir sa licence, après ces trois années d’étude passées en premier cycle, l’étudiant doit effectuer un stage d’une durée de trois mois au sein d’une entreprise et soutenir un mémoire sur ce stage ;
* Le second, d’une durée de deux ans, qui aboutit à un diplôme d’Ingéniorat ou BACC+5. La formation en second cycle est accessible aux étudiants ayant une licence. A la fin des deux années passées en second cycle, l’étudiant doit effectuer un stage d’une durée de six mois au sein d’une entreprise à l’issue duquel il soutient un mémoire de fin d’études.

Notons qu’après chaque cycle, l’étudiant est opérationnel et a le choix entre la poursuite de ses études ou l’accès immédiat dans le monde du travail.

Voici le cursus de l’ISPM :

Figure 3: Cursus à l'ISPM

#### Objectif de l’I.S.P.M

Dans le cadre de l’harmonisation du cursus de l’enseignement supérieur au niveau international, l’Etat Malagasy utilise actuellement le système L.M.D (Licence Master Doctorat) au niveau national. Pour cela, l’I.S.P.M a adopté cette méthode.

Par définition, le système LMD est un système d'enseignement supérieur anglo-saxon utilisant une formation à trois paliers : Licence, Master et Doctorat. Il offre une plus grande flexibilité dans la formation aussi bien aux apprenants qu'aux formateurs. Deux modes de formation sont offerts : une formation académique orientée sur la recherche et une formation appliquée orientée sur la professionnalisation.

L’utilisation du système LMD donne de nombreux avantages au niveau de notre établissement et aussi pour ses étudiants. Ces avantages sont : l’harmonisation des niveaux de formation universitaire, la mobilité accrue pour les étudiants, l’orientation progressive et le suivi du projet personnel de l’étudiant grâce à un système de passerelles, la professionnalisation facilitée par les unités d’Enseignements de stages, projets, etc.

#### Les particularités de l’I.S.P.M

Actuellement, les enseignants sont impliqués profondément dans la recherche et le système LMD (Licence- Masters- Doctorat) est déjà adopté.

##### Les portes ouvertes

L’année académique se divise en deux semestres. A chaque fin de semestres, un examen s’effectue. Le deuxième semestre se démarque par l’organisation des « Portes Ouvertes ». Cet évènement est une occasion pour les étudiants de montrer, au grand public, les « Projets » effectués dans le cadre des recherches personnelles.

##### Les supports pédagogiques

L’ISPM met à la disposition de ses étudiants :

* Deux laboratoires d’informatiques pour un total de cent vingt (120) ordinateurs ;
* Un laboratoire expérimental pour le département Biotechnologie ;
* Un laboratoire de travaux pratiques en électricité et électronique ;
* Un atelier de travaux pratique pour les étudiants en Mécanique ;
* Divers appareils topographiques.

##### L’examen clinique

L’examen « Clinique » est aussi une des plus grandes particularités de l’ISPM. C’est l’épreuve à passer avant la préparation de l’ingéniorat. Il consiste à évaluer de l’étudiant sur toutes ses connaissances, depuis la première année jusqu’à la cinquième année. La réussite à cet examen donne accès au stage ainsi qu’au mémoire de fin du Second Cycle.

##### Les supports éducatifs

L’I.S.P.M possède différentes infrastructures composées de :

* Plus de Soixante-dix salles de cours ;
* Une grande salle pouvant contenir jusqu’à 1500 étudiants ;
* Un amphithéâtre en plein air ;
* Des bureaux pour l’administration ;
* Des laboratoires d’informatique ;
* Un laboratoire de langes ;
* Un laboratoire de chimie, de biologie et de pharmacologie ;
* Un atelier de mécanique ;
* Deux cafétérias ;
* Deux salles de soutenance de mémoire ;
* Un terrain de Basket-ball ;
* Un terrain de football (Mini-Foot) ;
* Un terrain de Volley-ball.

##### La devise de l’I.S.P.M

L’ISPM a une devise particulière : **FAHAIZANA – FAMPANDROSOANA – FIHAVANANA**

##### Le logo de l’I.S.P.M



Figure 4: Logo de l'ISPM (source: ISPM)

* La toque noir signifie que l’I.S.P.M forge des cadre supérieurs d’où le «FAHAIZANA».
* Le globe terrestre montre que l’I.S.P.M s’ouvre au monde entier d’où le «FAMPANDROSOANA».
* Les Mains qui se serrent nous indiquent que l’I.S.P.M est ouvert à toutes les nationalités d’où le «FIHAVANANA».

##### L’hymne de l’I.S.P.M

**1.** Ilay I.S.P.M tena maminay

Tsy mba foinay tokoa hatrizay

Toera-nanabeazana ny tenanay

Mba ho tena olom-banona mahay

***Ref:*** *Ho mendrika tokoa,*

*Ho mafy orina avokoa*

*Ny fahaizana, fampandrosoana, fihavanana*

*Ho andry sy tokin’ny Tanindrazanay*

**2.**  Ny fahaizana no apinga ho enti-miady

Nampitaina,nozaraina sy nomena

Tsy andrarak’ilo fa ho tena kiady

Hoenti-mampandroso ny firenena

**3.** Ka ny fihavanana, firaisankina

No asandratray hatrany hatrany

Manoloana fifaliana, fijaliana

Mandra-piala anay ety an-tany

#### Adresse et contact

Le siège de l’I.S.P.M est actuellement à Ambatomaro – Antsobolo – Antananarivo 101.

|  |  |
| --- | --- |
| **Ses contacts** | : 034 17 583 69 / 033 12 171 60 / 032 84 509 39 / 020 26 384 08 |
| **Son adresse e-mail** | : [*ispm@ispm-edu.com*](mailto:ispm@ispm-edu.com) |
| **Son site web** | : [*www.ispm-edu.com*](http://www.ispm-edu.com) |



Figure 5: Site web de l'ISPM (source: ISPM)

#### Prix obtenus par l’I.S.P.M

Le tableau ci-dessous montre les prix obtenus par l’ISPM lors des différents concours.

Tableau 1: Prix obtenus par l'I.S.P.M depuis son existence (source: ISPM)

|  |  |  |  |
| --- | --- | --- | --- |
| Rang | PROJET | ANNEE | ORGANISATEUR |
| 1er | *Logiciel de Fanorona* | 1997 | Ambassade de France |
| 3ème | *SRI (Repiqueuse)* | 2003 | FIERMADA |
| 3ème | *Tracteur et Repiqueuse* | 2003 | Forum de la Recherche MENRS |
| 1er | *Analyse de la structure biotechnologique du riz* | 2006 | Salon de la Créativité et de l’invention |
| 2ème | *Etuve phytotronique assistée par ordinateur* | 2006 | Salon de la Créativité et de l’invention |
| 1er | *Réchaud à alcool* | 2007 | Fondation Tany Meva |
| 1er | *Valorisation de la jacinthe d’eau pour la fertilisation en aquaculture* | 2008 | Fondation Tany Meva |
| 1er | *Production d’un plastique biodégradable* | 2009 | Fondation Tany Meva |
| 1er | *Smart planning managing* | 2012 | Forum de la Recherche MESUPRES |
| 2ème | *Intelligent House* | 2012 | Forum de la Recherche MESUPRES |
| 4ème | *Energies renouvelable* | 2012 | Forum de la recherche MESUPRES |
| 1er | *Olympiade Informatique de Madagascar* | 2012 | Madagascar au temps TIC |
| 2è | *Concours de développement d'application malagasy : « Flash Credit »* | 2014 | Orange Madagascar |
| 2è | *Concours de développement d'application malagasy : « Safe sms »* | 2014 | Orange Madagascar |
| 1er | *WEBCUP* | 2015 | GOTICOM & Office du Tourisme |
| 1er | *WEBCUP VIRALITE* | 2015 | WEBCUP Association |

#### Organigramme de l’I.S.P.M

Figure 6: Organigramme de l'ISPM (source: ISPM)

### PRESENTATION DE LA SOCIETE

#### Historique de l’entreprise eTech

eTech est une Société de Services en Ingénierie Informatique (SSII) implantée à Madagascar depuis 2006. Son local se trouve dans l’immeuble Master Trade à Anosivavaka Ambohimanarina, route du Pape. Elle est née d’une collaboration entre Patrick Trebalag, Dominique et Fred Félix en tant que société spécialisée dans le développement de solutions informatiques.

En juin 2008, suite à des problèmes commerciaux, les actionnaires d’eTech se sont rapprochés de Monsieur Pierre-Paul Ardile, Président Directeur Général d’un cabinet de conseil expert des solutions professionnelles mobiles appelé : « Mov’age», pour un transfert d’actions.

En Septembre 2008, sous la direction de Monsieur Pierre-Paul Ardile, de nouveaux collaborateurs, associant compétences malagasy et française, reprennent les activités de la société. Depuis, eTech travaille en étroite collaboration avec la société de services informatiques française « ArkeUp » qui la représentent en France.

eTech, spécialiste des solutions digitales web, mobile et e-business en offshore fait partie du Groupe ArkeUp avec une représentation en France et au Maroc.

#### Siège de l’entreprise

L’entreprise eTech est située à Antananarivo I, route du Pape Jean Paul II, Anosivavaka Ambohimanarina, dans la capitale de Madagascar.

#### Logo de l’eTech



Figure 7: Logo de l'entreprise eTech (source: eTech)

#### Caractéristiques de l’entreprise

##### Adresse et contact

|  |  |
| --- | --- |
| **Adresse** | : Immeuble Master Trade, route du Pape Jean Paul II, Anosivavaka Ambohimanarina Antananarivo |
| **Téléphone** | : +261 20 22 536 04 |
| **Boîte postale** | : 8721 |
| **Code postal** | : 101 |
| **Courriel** | : etech@etechconsulting-mg.com |
| **Site web** | : <http://www.etechconsulting-mg.com> |

##### Activité principale

La société eTech se spécialise dans le domaine du développement d'applications informatiques (sites web, applications mobiles) et utilise un certain nombre de technologies afin de répondre à tous les besoins et exigences de ses clients et collaborateurs.

Entre autre, l’entreprise effectue :

* L’audit et l’analyse des besoins ;
* Les spécifications et la conception de base de données ;
* La conception graphique : de la création à l’intégration ;
* Le développement ;
* Le déploiement ;
* La maintenance et le support des applications.

Le tableau ci-après résume les différentes technologies utilisées par eTech.

Tableau 2: Tableau des différentes technologies utilisées par eTech (source eTech)

|  |  |
| --- | --- |
| Catégorie | Technologies |
| Langages de programmation | PHP, C#, Java, Objective-C, Python, |
| Mobilité | iOS, Android, Windows mobile, PhoneGap |
| CMS | Drupal, WordPress |
| ERP | Odoo/OpenERP |
| E-commerce | Magento, Prestashop |
| Serveurs web et serveurs d’applications | Apache, Tomcat, Microsoft IIS, JBoss |
| Autres technologies (Frameworks, autres plateformes…) | Zend, Symfony, AngularJS, AJAX, R.I.A, Flex3/Flex4, GWT, Adobe AIR, Silverlight, .Net, Talend |

##### Ressources humaines

Son centre de compétences et de développement situé à Madagascar regroupe plus de 160 ingénieurs et experts dans différentes technologies. Dans le cadre de la forte croissance de son activité, la société eTech a recruté plus de 30 personnes sur le début d’année 2015 et souhaite recruter encore 70 personnes avant la fin de l’année 2016.

##### Organigramme de l’entreprise

Figure 8: Organigramme de l'eTech (source: eTech)

* PDG : Monsieur Pierre Paul ARDILE;
* Responsable Administratif et Financier : Madame Eva RALALAHARIMANANA ;
* Responsable Commercial : Madame Tiana NOHAVY ;
* Chargé de Marketing : Karen Radrigan ;
* Directeur de projet (.Net - Java - Mobile) : Monsieur Rija RAOBIVELO RABETOKOTANY;
* Directeur de projet (Transversale et Qualité): Monsieur Clarel RAKOTONDRAHAJA;
* Directeur de projet (Web-PHP) : Monsieur Andry RAMANAMBAHOAKA.

### Présentation du thème

#### Analyse de la situation

Durant notre période de stage chez eTech, on était affecté dans plusieurs projets parmi lesquels des projets en cours et d’autres en création. Dans chaque projet, on rencontre toujours quelques blocages qui finissent toujours par être résolus. Parmi ces nombreux projets, **« Groupe Santé »**, qui est un des projets en début de création.

« Groupe Santé » est une association réunissant les établissements ou centres de santé à Madagascar, comme son nom l’indique. Elle a donc un site web personnel qui a pour but principal de regrouper tous les établissements de santé à Madagascar. Chaque établissement étant lié à des projets, les utilisateurs du site peuvent participer ou bénéficier aux projets. Cela peut se faire en un clic après avoir remplir un formulaire via web. Il faut savoir qu’un projet peut appartenir à plusieurs centres de santé.

Nous avons décidé d’effectuer les analyses approfondies concernant le système d’information du client selon leur cahier de charge et de créer leur nouveau site web.

N’oublions pas de rappeler que « Groupe Santé » avait déjà un autre site web mais sous un autre nom. Auparavant, l’association était connue sous un autre nom. Aujourd’hui, elle a décidé de changer de nom et en parallèle créer un nouveau site web avec une ergonomie plus agréable. Ce nouveau site web offrira une facilité d’utilisation pour les visiteurs du site et aussi une nouvelle interface graphique. L’association a donc fait appel à la société eTech pour refondre son site web au niveau ergonomique ainsi qu’au niveau fonctionnel et design du site.

Le projet consiste à analyser les nouveaux besoins selon le cahier des charges du client et trouver ainsi la solution la mieux adaptée pour réaliser et mettre en place le nouveau site web.

Cependant, quelques problématiques se posent. La première est comment donner le maximum d’informations aux utilisateurs. La deuxième problématique est la facilité de manipulation et maintenance du site web. La troisième et dernière problématique est sur la facilité de communication avec la clientèle à travers le site web.

Pour notre cas, on a choisi d’utiliser un CMS. Plusieurs types de CMS s’offrent à nous. Pour répondre aux besoins du client et résoudre les problématiques ci-dessus, le CMS Drupal est la solution. On a donc développé le site web sous PHP avec le CMS Drupal 7.

#### Généralité du projet

En tant que développeur, il faut savoir surtout travailler en équipe. Pour la réalisation de ce projet, on a compté trois (3) développeur, un (1) intégrateur, un (1) ingénieur qualité et le tous dirigé par un chef de projet au sein de l’eTech. En somme, le projet a duré quatre (4) mois.

Nous avons aussi utilisé Git comme logiciel de gestion de versions décentralisé.

La réalisation du projet sera composée de diverses étapes dont :

* La récolte des informations selon le cahier de charge ;
* L’analyse des informations ;
* La conception du site web.

##### Logo du site web



Figure 9: Logo du site web G-Santé

#### Objectif et opportunité du projet

##### Objectif du projet

L’objectif principal est tout d’abord de satisfaire et de répondre aux demandes du client selon les spécifications dans le cahier de charge. En d’autre terme, nous avons pour but de concevoir un site web.

Il a aussi pour but d’offrir une meilleur condition d’utilisation aux utilisateurs du site web. Et de leurs donner le maximum d’informations et cela dans un seul site web, *« Groupe Santé »,* avec une facilité d’accessibilité.

Du point de vue personnel, la conception de ce site web évoluera nos connaissances et expériences, particulièrement dans le travail en équipe.

##### Opportunité du projet

Ce projet offre surtout plusieurs opportunités à l’association « Groupe Santé ». Grâce à ce site web, l’association pourra publier en temps réel les dernières nouvelles concernant chaque établissement de santé. Avec les atouts de Drupal, l’administrateur peut facilement administrer le site et y intégrer des vidéos. Quand le site web sera mise en ligne, plus besoin de dépenser le maximum de budget pour publier des annonces ou faire des publicités.

Pour les utilisateurs, ils pourront être à la page des dernières nouvelles, car le site sera accessible à tout moment et n’importe où. Pour participer ou bénéficier aux projets dans les centres de santé, plus besoin de se déplacer, car l’inscription se fait via web, en remplissant quelques formulaires.

Depuis l’arrivée des grands moteurs de recherche comme Google, il est de plus en plus facile d’être visible sur le Web. Des gens de partout peuvent trouver ce site Internet et consulter leurs produits et services. Leur clientèle ne se limite donc plus aux gens de leur ville ou de leur région pour récolter des informations.

# PRESENTATION DES OUTILS ET LANGAGES DE DEVELOPPEMENT

### Langage de modélisation [D] [E]

#### Utilité d’un langage de modélisation dans un projet [E]

##### Définition d’un modèle

Un **modèle** est une *représentation abstraite* et simplifiée, d'une entité (phénomène, processus, système) du monde réel en vue de le décrire, de l'expliquer ou de le prévoir. Sous une autre façon de voir, un modèle, c'est une théorie orientée vers l'action qu'elle doit servir.

Concrètement, un modèle permet de réduire la complexité d'un phénomène en éliminant les détails qui n'influencent pas son comportement de manière significative. Il reflète ce que le concepteur croit important pour la compréhension du système d’information.

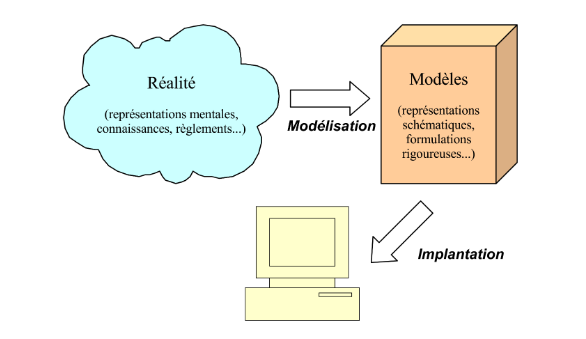


Figure 10: Schéma d'une modélisation

##### Nécessité de la modélisation

Le but de la modélisation d’un SI, est de spécifier les besoins et les exigences des acteurs, le système et l’architecture globale.

Modéliser un système avant sa réalisation permet de mieux comprendre le fonctionnement du système. C'est également un bon moyen de maîtriser sa complexité et d'assurer sa cohérence. Un modèle est un langage commun, précis, qui est connu par tous les membres de l'équipe. Cette communication est essentielle pour aboutir à une compréhension précise d'un problème donné.

Dans le domaine de l'ingénierie du logiciel, le modèle permet de mieux répartir les tâches. Le choix du modèle a donc une influence capitale sur les solutions obtenues.

#### U.M.L [2] [8] [D]

##### Introduction

La description de la programmation par objets a fait ressortir l'étendue du travail conceptuel nécessaire : définition des classes, de leurs relations, des attributs et méthodes, des interfaces, etc.

Pour programmer une application, il ne convient pas de se lancer tête baissée dans l'écriture du code : il faut d'abord organiser ses idées, les documenter, puis organiser la réalisation en définissant les modules et étapes de la réalisation. C’est cette démarche à l’écriture que nous appelons *modélisation*. La modélisation demande aux maîtrises d'ouvrage une compétence et un professionnalisme qui ne sont pas aujourd'hui répandus.

##### Historique

Les méthodes utilisées dans les années 1980 pour organiser la programmation impérative (notamment Merise) étaient fondées sur la modélisation séparée des données et des traitements. Lorsque la programmation par objets prend de l'importance au début des années 1990, la nécessité d'une méthode qui lui soit adaptée devient évidente. Plus de cinquante méthodes apparaissent entre 1990 et 1995 (Booch, Classe-Relation, Fusion, HOOD, OMT, OOA, OOD, OOM, OOSE, etc.), mais aucune ne parvient à s'imposer. En 1994, le consensus se fait autour de trois méthodes :

* *OMT* de James Rumbaugh (General Electric) fournit une représentation graphique des aspects statique, dynamique et fonctionnel d'un système ;
* *OOD* de Grady Booch, définie pour le Department of Defense, introduit le concept de paquetage (package) ;
* *OOSE* d'Ivar Jacobson (Ericsson) fonde l'analyse sur la description des besoins des utilisateurs (cas d'utilisation, ou use cases).

Chaque méthode avait ses avantages et ses partisans. Le nombre de méthodes en compétition s'était réduit. Événement considérable et presque miraculeux, les trois gourous qui régnaient chacun sur l'une des trois méthodes se mirent d'accord pour définir une méthode commune. **UML** est né de cet effort de convergence. L'adjectif « *unified »* est là pour marquer qu'UML unifie, et donc remplace.

UML est donc non seulement un outil intéressant, mais une norme qui s'impose en technologie à objets et à laquelle se sont rangés tous les grands acteurs du domaine, acteurs qui ont d'ailleurs contribué à son élaboration.

##### Définition

**UML** est un ***langage unifié de modélisation*** objets à base de représentation graphique, conçu pour fournir une méthode normalisée pour visualiser la conception d'un système.

##### Pourquoi utiliser UML ?

De nos jours, les outils de modélisation de processus métier s’étoffent chaque année et les suites logicielles sont de plus en plus nombreuses. L’usage et les fonctionnalités d’UML diffèrent d’un périmètre à un autre, selon les besoins des clients et des fournisseurs d’applications.

Il permet dans un premier temps de bien définir les besoins clients, et ainsi d’éviter des surcoûts liés à la livraison d’un logicielle qui ne satisfait pas le client.

Aussi, elle apporte une compréhension rapide du programme à d’autres développeurs externes en cas de reprise du logiciel et facilite sa maintenance.

Toutefois, il existe aussi des inconvénients quant à l’utilisation d’UML. Pour la plupart, essentiellement liés au dépassement du délai de livraison du logiciel. Pour pallier à ce problème, le recours à un cycle de projet en spirale est recommandé car il apporte plus d’agilité et une meilleure gestion des risques.

##### Les avantages et inconvénients de l’U.M.L [F]

###### Les points forts d’UML

**UML est un langage formel et normalisé :**

* gain de précision ;
* gage de stabilité ;
* encourage l'utilisation d'outils : UML est un support de communication performant.

**UML est un support de communication performant :**

* Il cadre l'analyse ;
* Il facilite la compréhension de représentations abstraites complexes ;
* Son caractère polyvalent et sa souplesse en font un langage universel.

###### Les points faibles d’UML

* La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation. UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.
* Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue.

##### La notion de classe

Une classe représente donc une catégorie d'objets. Elle apparaît aussi comme un *moule* ou une *usine* à partir de laquelle il est possible de créer des objets. (C’est en quelque sorte une « boîte à outils » qui permet de fabriquer un objet). On parle alors d'un objet en tant qu'[**instance**](https://fr.wikipedia.org/wiki/Instance_%28programmation%29) d'une classe.

En d’autre terme, une classe est un type de données abstrait qui précise des caractéristiques (*attributs et méthodes*) communes à toute une famille d'objets et qui permet de créer (instancier) des objets possédant ces caractéristiques. Les autres concepts importants qu'il nous faut maintenant introduire sont l'encapsulation, l'héritage et l'agrégation.

Exemple :

VOITURE

**marque**

**couleur**

**immatriculation**

Classe

Attribut

demarrer()

conduire()

Méthode

###### Les attributs d’une classe

Une classe de type entité est composée d’un ensemble d’informations élémentaires, appelée ***attribut de la classe****.*

La syntaxe générale d’un attribut : *nom\_attribut[multiplicite][type\_attribut][=valeur initiale][{propriete}]*

* *Nom\_attribut :* un attribut doit toujours être nommé
* *Multiplicité :* indication d’un nombre minimum et maximum de valeur possible de l’attribut pour une instance de la classe.

Exemple :

PERSONNE

**Nombre\_enfant [0,5]**

On pourra indiquer entre 0 et 5 le nombre d’enfants.

###### Les méthodes d’une classe

En POO, une méthode est une [**fonction**](https://fr.wikipedia.org/wiki/Fonction_%28informatique%29) membre d'une [classe](https://fr.wikipedia.org/wiki/Classe_%28informatique%29).

Une méthode peut être:

* une *méthode d'*[*instance*](https://fr.wikipedia.org/wiki/Instance_%28programmation%29), n'agissant que sur un seul objet (instance de la classe) à la fois;
* une *méthode statique* (ou méthode de classe), indépendant de toute instance de la classe (objet).

##### Notion d’objet

* Un objet est une entité aux frontières précises qui possède une identité (un nom).
* Un ensemble d'attributs caractérise l'état de l'objet.
* Un ensemble d'opérations (méthodes) en définissent le comportement.
* Un objet est une instance de classe (une occurrence d'un type abstrait).
* Une classe est un type de données abstrait, caractérisé par des propriétés (attributs et méthodes) communes à des objets et permettant de créer des objets possédant ces propriétés.

Exemple : Instance de la classe « Voiture ».

Twingo : VOITURE

**marque = Renault**

**couleur = grise**

**immatriculation = 995 LKZ 75**

##### UML en œuvre [E]

**UML n'est pas une méthode**, car ses auteurs ont en effet estimé qu'il n'était pas opportun de définir une méthode en raison de la diversité des cas particuliers. Ils ont préféré se borner à définir un langage graphique qui permet de représenter et de communiquer les divers aspects d'un système d'information. Aux graphiques sont bien sûr associés des textes qui expliquent leur contenu. UML est donc un métalangage, car il fournit les éléments permettant de construire le modèle qui, lui, sera le langage du projet.

UML 2.0 comporte ainsi **treize (13) types de diagrammes** représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ils se répartissent en deux grands groupes :

* **Les diagrammes utilisés pour l’étape d’analyse (7) :**
  + Le diagramme de classe ;
  + Le diagramme d’objets ;
  + Le diagramme d’états-transitions ;
  + Le diagramme d’activité ;
  + Le diagramme de collaboration ;
  + Le diagramme de séquence et de communication;
  + Le diagramme de cas d’utilisation.
* **Les diagrammes utilisés dans l’étape de conception :**
  + Le diagramme des composants ;
  + Le diagramme de déploiement.

###### Diagramme de cas d’utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre.

**Objectif :** Son objectif principal est de rendre service à ses utilisateurs. Par conséquent, il faut comprendre les désirs et les besoins des futurs utilisateurs. Le processus de développement sera donc centré sur l’utilisateur.

Les éléments du diagramme de cas d’utilisation

* **Acteur :** c’est l’idéalisation d’un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme (figure 10) avec son nom inscrit dessous.



Figure 11: Représentation d'un auteur

* **Cas d’utilisation :** c’est l’usage que des acteurs font du système, généralement composé de plusieurs scénarios. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin. Un cas d’utilisation se représente par une ellipse (figure 2.3) contenant le nom du cas.



Figure 12: Représentation d'un cas d'utilisation

Représentation d’un diagramme de cas d’utilisation

Exemple simplifié de diagramme de cas d’utilisation modélisant une borne d’accès à une banque.

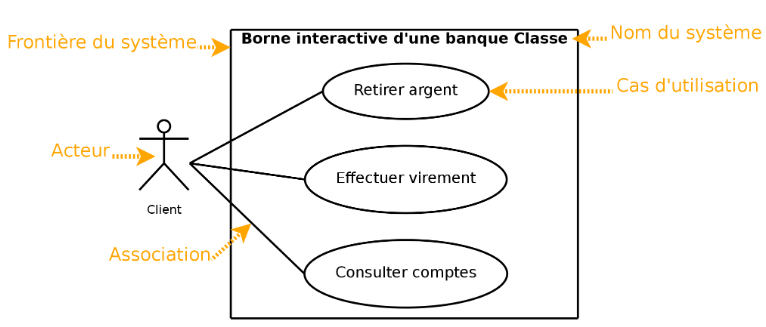


Figure 13: Représentation d'un diagramme de cas d'utilisation

###### Diagramme de classes

Le diagramme de classes est généralement considéré comme le plus important dans un développement orienté objet. Il représente l'*architecture conceptuelle du système* : il décrit les classes que le système utilise, ainsi que leurs liens.

**Objectif :** modéliser les entités du système d’information.

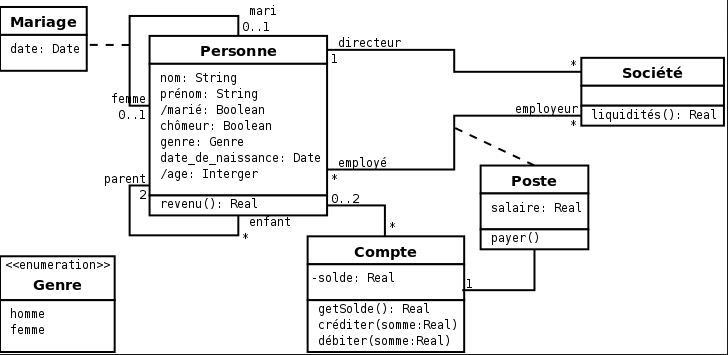


Figure 14: Exemple de diagramme de classes

###### Diagramme d’objets

Le diagramme d'objets permet *d'éclairer un diagramme de classes en l'illustrant par des exemples*. Il est, par exemple, utilisé pour vérifier l'adéquation d'un diagramme de classes à différents cas possibles.

**Objectif :** Mettre en évidence les liens entre les objets. Ils sont essentiellement utilisés pour comprendre ou pour illustrer des parties complexes de diagramme de classes.

###### Diagramme d’états-transitions

Le diagramme d'états-transitions représente la façon dont évoluent les objets appartenant à une même classe. La modélisation du cycle de vie est essentielle pour représenter et mettre en forme la dynamique du système.

**Objectif :** Représenter des traitements permettant de gérer le domaine étudié.

###### Diagramme d’activités

Le diagramme d'activités n'est autre que la transcription dans UML de la représentation du processus telle qu'elle a été élaborée lors du travail qui a préparé la modélisation : il montre l'enchaînement des activités qui concourent au processus.

###### Diagramme de séquence et de communication

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur. Il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre. On peut représenter les mêmes opérations par un diagramme de communication, graphe dont les nœuds sont des objets et les arcs (numérotés selon la chronologie) les échanges entre objets. En fait, diagramme de séquence et diagramme de communication sont deux vues différentes, mais logiquement équivalentes (on peut construire l'une à partir de l'autre) d'une même chronologie.

**Objectif :** Montrer les interactions entre objets selon un point de vue temporel.

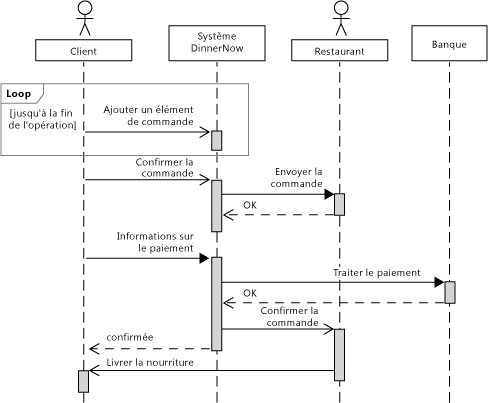


Figure 15: Exemple de diagramme de séquence

#### Analyse Orientée Objet [3]

L'analyse orientée objet (OOA en anglais) est une réflexion faite en amont de la création d'une application ou d'un logiciel. Elle a pour but **de modéliser l'ensemble des besoins de l'utilisateur** pour décrire ce qu'il faut faire. Durant l’analyse, *on oublie tout ce qui concerne l’implémentation concrète de l’application*. On reste totalement hors de tout langage spécifique.

L’objectif de l’A.O.O sera donc de construire le modèle objet du domaine ainsi que le diagramme de classe du système. C’est ce dernier qui servira de base pour la conception orienté objet.

Les étapes à suivre seront :

* L’identification des acteurs et des cas d’utilisation ;
* La construction du modèle de cas d’utilisation ;
* L’inventaire et construction du dictionnaire des objets métiers ;
* L’élaboration du diagramme d’objet du modèle ;
* La mise en place du diagramme de classe du modèle.

##### Identification des acteurs et des cas d’utilisation

Il s’agit de la solution U.M.L pour représenter le modèle conceptuel. Les cas d’utilisation permettant de structurer les besoins des utilisateurs et les objectifs correspondants du système. Ils servent de base à la traçabilité des exigences du système dans un processus de développement intégrant U.M.L.

**Exemple :**

*Nous allons utiliser comme système à modéliser le guichet automatique bancaire.*

Le client peut consulter son compte ou retirer de l’argent. Le technicien peut ravitailler en argent liquide le GAB et éteindre ou allumer le GAB surtout lors du ravitaillement. Comme ce système est simple, nous pouvons identifier clairement les acteurs : le client et le technicien ainsi que les cas d’utilisation : consulter son compte, retirer de l’argent, éteindre ou allumer GAB, ravitailler le GAB.

##### Construction du modèle de cas d’utilisation

Après avoir identifié les acteurs et les cas d’utilisations, nous pouvons dresser le diagramme de cas d’utilisation du système.

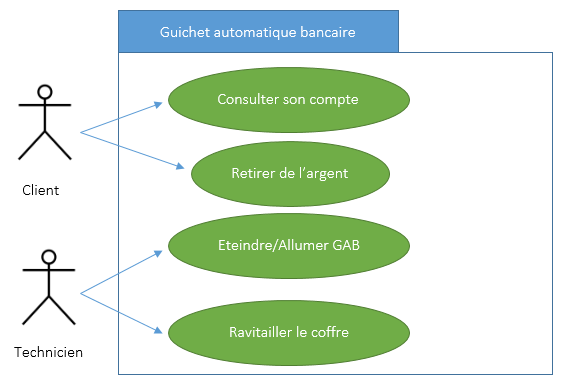


Figure 16: Diagramme de cas d'utilisation

##### Inventaire et construction du dictionnaire des objets métiers

###### Processus

Pour pouvoir identifier et dresser le dictionnaire des objets métiers, nous devons prendre chaque cas d’utilisation et d’écrire dans l’ordre le scénario du processus.

**Exemple :**

Pour le cas *« retirer de l’argent »,* on a :

* Le client insert sa carte ;
* Le client entre son code secret ;
* Le système valide le code ;
* Le client entre le montant ;
* Le système vérifie le solde, fournit de l’argent, débite le compte et rend la carte.

On cherche les noms dans la description de notre cas d’utilisation. Chaque nom peut être un objet. On filtre pour ne garder que des vrais objets qui seront les objets métiers. On pourra dresser ensuite le dictionnaire des objets métiers.

###### Dictionnaire des objets métiers

Inventaire des objets métiers

La méthode standard pour faire les inventaires des objets métiers est de lire les spécifications de l’application a développé, de considérer si chaque nom peut être un objet métier ou non et enfin, de déterminer la responsabilité de chaque verbe pour l’objet métier.

La responsabilité d’un objet est soit la connaissance d’une information, soit l’action qu’il fait.

Pour l’exemple ci-dessus, on a l’inventaire suivant :

Tableau 3: Tableau de l'inventaire des objets métiers

|  |  |  |
| --- | --- | --- |
| NOM | TYPE | RESPONSABILITE |
| *Client* | *Acteur* | Insérer carte, entrer le code secret, saisir le montant. |
| *Carte* | *Objet* | Connaître le code secret, connaître le numéro de compte. |
| *Système* | *Acteur* | Valider, vérifier le code, fournir de l’argent. |
| *Compte* | *Objet* | Connaître le solde, débite le compte. |

Nous trouvons donc 2 objets métiers :

* La carte ;
* Le compte.

Les CRC-Card

Les CRC-Card sont des outils qui ont été proposées par *Ward Curringhan* et *Kert Beck.* Elles sont utilisées pour passer de l’inventaire des objets métiers au diagramme de classe. Une CRC-Card est une petite carte de type carte d’indexation divisée en 3 parties :

* *le nom de la classe de l’objet recensé ;*
* *la responsabilité :* tous ce que la classe est censée connaître et faire ;
* *les collaborateurs :* les autres classes qui interagissent avec la classe en question pour parfaire ces responsabilités.

Représentation d’une CRC-Card

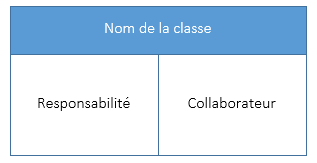


Figure 17: Représentation d'une CRC-Card

Pour l’exemple ci-dessus, on a les CRC-Card suivantes :

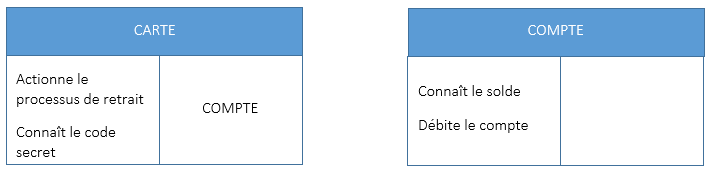


Figure 18: Exemple de CRC-Card

Diagramme de classe métier

Le diagramme de classe permet de définir quels seront les composants du système final. Il permet aussi de séparer les composantes de manière à pouvoir repartir le travail de développement entre les membres du groupe.

A partir des CRC-Card ci-dessus, nous pouvons établir le diagramme de classe de l’application.

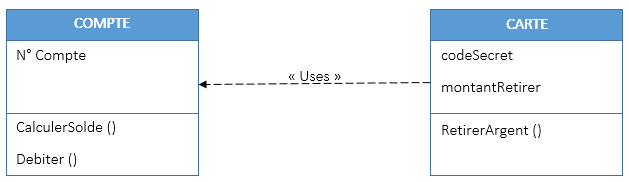


Figure 19: Diagramme de classe de l'exemple ci-dessus

###### Analyse dynamique :

Le diagramme de classe nous a permis de générer le prototype de l’application, mais il reste au développeur de coder les méthodes de chaque classe, on doit donc modéliser les actions et l’état des objets concernés à un instant donné par la méthode concernée.

Pour ce faire, nous pouvons utiliser selon le cas et selon les besoins 4 diagrammes de vu dynamique :

* Diagramme de collaboration ;
* Diagramme de séquence ;
* Diagramme d’état-transitions ;
* Diagramme d’activité.

### Langages de programmation

#### Introduction [G]

En [informatique](https://fr.wikipedia.org/wiki/Informatique), un **langage de** [**programmation**](https://fr.wikipedia.org/wiki/Programmation_informatique) est une notation conventionnelle destinée à *formuler des* [*algorithmes*](https://fr.wikipedia.org/wiki/Algorithme) et *produire des* [*programmes informatiques*](https://fr.wikipedia.org/wiki/Programme_informatique) *qui les appliquent*. D'une manière similaire à une langue naturelle, un langage de programmation est composé d'un [alphabet](https://fr.wikipedia.org/wiki/Alphabet), d'un [vocabulaire](https://fr.wikipedia.org/wiki/Vocabulaire), de règles de [grammaire](https://fr.wikipedia.org/wiki/Grammaire) et de [significations](https://fr.wikipedia.org/wiki/Sens_%28linguistique%29).

Les premiers langages de programmation ont été créés dans les années 1950. De nombreux concepts de l'informatique ont été lancés par un langage, avant d'être améliorés et étendus dans les langages suivants. La plupart du temps la conception d'un langage de programmation a été fortement influencée par l'expérience acquise avec les langages précédents.

#### Critères de choix d’un langage [H]

Il est impossible de dire de but en blanc que tel ou tel langage soit le meilleur: Le meilleur choix sera très dépendant de vos besoins et contraintes. Les questions ci-dessous devraient vous guider.

Notez que chaque question peut ou non être importante ou pertinente pour votre projet. Tout ceci est très dépendant de votre projet.

##### Nature

* Est-ce un langage à typage fort, faible ? à typage statique/dynamique ?
* Le langage permet-il une séparation nette données/code ?
* Permet-il une séparation nette logique du programme/présentation ?
* Est-ce important pour votre projet ?
* Ce langage (et des librairies) peut-il se conformer à des standards ? (XML, SOAP, HTTP, Rosetta.Net, ITIL, RFC et protocoles divers...)

##### Portabilité

* Ce langage est-t-il dépendant d'un système d'exploitation particulier ?
* Sur combien de systèmes d'exploitation des compilateurs pour ce langage sont disponibles ? De combien de vendeurs différents ?
* Est-ce que les API et librairies que vous utilisez seront disponibles dans d'autres systèmes d'exploitation ?

##### Stabilité

* Depuis combien de temps existe ce langage ?
* Est-il encore en beta ?
* Des entreprises l'utilisent-elles ? Si oui, pour des tâches critiques ?
* Avez-vous pu trouver sur internet des discussions critiques sur ce langage ? Que disent-elles ?

##### Métier et maintenance

* Trouve- t ‘on facilement des gens compétents sur ce langage ?
* Ce langage est-t-il bien documenté ? La documentation est-elle à jour ?
* Des exemples sont-ils disponible en nombre ?
* Ce langage est-t-il déjà utilisé dans votre entreprise ?

##### Ouverture

* Ce langage permet-il d'interagir avec d'autres langages ? (C, C++, Java...)
* A quelles API ce langage permet-il d'accéder ? (Win32, SOAP, .Net...)
* A quels types de bases de données ce langage permet-il de se connecter ? (Oracle, SQL Server, MySQL, ou génériques: ODBC, JDBC... ?)

##### Sécurité

* Ce langage et ses librairies possède-t-il des fonctions intrinsèques de sécurité ?
* Est-il vulnérable aux risques de buffer overflow ? Est-ce important pour votre projet ?
* Aux risques de SQL injection ?
* Aux risques de cross-site scripting ?

#### CMS Drupal 7 [I]

##### Un C.M.S, qu’est-ce que c’est ?

L’acronyme CMS signifie *« Content Management System »*, pour système de gestion de contenu. C’est un type d’application web permettant de séparer le contenu de la mise en forme pour des sites web, et fournissant des moyens simplifiés d’éditer ce contenu. De cette manière, il est possible de maintenir et faire vivre un site web sans la moindre connaissance dans les métiers du Web. Tout de même, pour répondre aux demandes du client, de nombreuses connaissances dans le métier sont nécessaires.

Dans l'absolu, un CMS ne sert que pour les sites web ayant pour vocation à être mis à jour assez régulièrement.

##### Définition

Drupal est un [**système de gestion de contenu**](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_contenu) (CMS) [libre et open-source](https://fr.wikipedia.org/wiki/Free/Libre_Open_Source_Software) publié sous la [licence publique générale GNU](https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU), et écrit en [PHP](https://fr.wikipedia.org/wiki/PHP:_Hypertext_Preprocessor). Par ailleurs Drupal peut être utilisé pour gérer des sites collaboratifs regroupant forum d'échanges, annonces, agendas d'événements... et permet la gestion de groupes de travail.



Figure 20: Logo de Drupal (source: www.drupal.org)

##### Historique

Drupal est créé en *1999* par Dries Buytart à l’Université d’Anvers. En 2009, il est classé 1er du concours organisé par la société britannique [Packt](https://fr.wikipedia.org/w/index.php?title=Packt&action=edit&redlink=1) [(en)](https://en.wikipedia.org/wiki/Packt) dans la catégorie *Best Open Source PHP CMS* suivi par [WordPress](https://fr.wikipedia.org/wiki/WordPress) et. Il fait également son entrée en première position dans le *Hall of Fame Award* nouvellement créé.

En 2010, *Angela Byron* annonce que Drupal équipe à peu près 1 % des sites Internet mondiaux.

Aujourd'hui la communauté de Drupal compte plus de 630 000 membres ce qui lui assure une compatibilité avec les dernières technologies.

##### Versions

###### Drupal 6

Il existait dans Drupal 6 plusieurs types de modules, les plus simples étant les suivants :

* les *modules de bloc*, associés à des informations de petite taille (exemple : qui est en ligne ou combien de personnes, qui sont les derniers inscrits, quels sont les derniers posts, les plus populaires...) ;
* les *modules de nœud*, qui engendrent ce qui n'est pas dans les marges : blog, forum, pages, formulaires, etc.

###### Drupal 7

Dans Drupal 7, il n'existe plus que des zones : l'administrateur du site peut les déplacer à tout moment comme il le veut; le concepteur leur attribue des tailles, des polices et des teintes. Le contenu d'un nœud peut donc parfaitement s'afficher dans les marges latérales si on le désire.

###### Drupal 8

Drupal 8, sorti le 19 novembre 2015, s'articule sur le [framework](https://fr.wikipedia.org/wiki/Framework) français [Symfony 2](https://fr.wikipedia.org/wiki/Symfony) dans un souci de normalisation, de s'ouvrir à un grand nombre de développeurs et de tirer pleinement profit de l'approche objet de PHP, en particulier les [espaces de noms](https://fr.wikipedia.org/wiki/Espaces_de_noms) qui faciliteront l'isolation des modules. Il permettra l'affichage des mêmes contenus - présentés de façon appropriée dans chaque cas - sur les PC, les tablettes et les [téléphones mobiles](https://fr.wikipedia.org/wiki/Smartphone).

##### Principes

Drupal peut être utilisé à quatre niveaux différents :

* Tel quel : une fois celui-ci installé et paramétré, il est utilisable pour créer du contenu structuré et annotable par des utilisateurs qui peuvent s'enregistrer sur le site. Les menus du site ont alors un aspect standard ;
* Personnalisation simple : il est ensuite possible de personnaliser l'emplacement d'affichage, ou l'affichage lui-même, de composants visuels standards (date et heure, derniers posts, nombre de connectés, etc.), ainsi que le thème d'affichage du site ;
* Extension par ajouts externes : ajout, paramétrage et personnalisation de modules optionnels n'appartenant pas au noyau. À ce stade et au suivant, il n'est pas rare que le développeur du site écrive aussi un thème de présentation qui lui soit propre ;
* Extension par développement interne : écriture de nouveaux modules.

##### Les caractéristiques de Drupal

###### Moteur de Template

Drupal permet d'utiliser comme moteur de [Template](https://fr.wikipedia.org/wiki/Template_%28programmation%29) *XTemplate*, *PHPTemplate* qui est le moteur de Template officiel depuis la version 4.7, mais aussi [*Smarty*](https://fr.wikipedia.org/wiki/Smarty).

###### Permissions

De façon à permettre une souplesse maximale, les permissions sont gérées par des libellés que les administrateurs et rédacteurs de modules choisissent librement. Il faut juste prendre garde à ce que le même libellé ne soit pas utilisé par deux modules différents pour désigner des permissions différentes.

###### Modules

Modules les plus utilisés

Tableau 4: Modules les plus utilisés (source: www.drupal.org)

|  |  |
| --- | --- |
| View | présentation de listes de contenu sur des critères de son choix. |
| Content Construction Kit | création de nouveaux types de contenu par ajout de champs. Intégré au core dans Drupal 7. |
| Token | gestion de formats normalisés (par exemple les URL, etc.) |
| Pathauto | insertion automatique de chemin en fonction du contenu. |
| Update-status | prévient des mises à jour de modules. |
| Image | remontée de contenu image par les utilisateurs: Permet de créer des galeries d'images. Ce module nécessite le module *upload*. Il permet également d'effectuer des traitements sur les images transférées (création de vignettes, etc.). Intégré au core dans Drupal 7. |

Modules officiels

Tableau 5: Modules officiels (source : www.drupal.org)

|  |  |
| --- | --- |
| Aggregator | utilisé pour agréger du contenu syndiqué. |
| archive | affiche un calendrier pour accéder aux archives. |
| Block | contrôle l'affichage des boîtes apparaissant en marge du contenu principal. |
| Blog | fournit un [moteur de blog](https://fr.wikipedia.org/wiki/Moteur_de_blog) pour mettre à jour facilement et régulièrement une page web ou un [blog](https://fr.wikipedia.org/wiki/Blog) |
| Blogapi | permet aux utilisateurs de poster en utilisant les applications qui supportent les API blogue XML-RPC. |
| Book | permet aux utilisateurs de collaborer à l'écriture d'un livre. |
| comment | permet aux utilisateurs de commenter et de discuter le contenu publié. |
| Drupal | permet aux membres d'ouvrir une session en utilisant un code d'identification Drupal, et peut signaler à un serveur central l'existence de votre site. |
| Filter | contrôle le filtrage du contenu en préparation pour l'affichage. |
| Forum | rend possible les fils de discussion à propos de sujets généraux. |
| Help | contrôle l'affichage de l'aide en ligne. |
| Legacy | fournit un système de mise à jour de Drupal. |
| Locale | permet la traduction de l'interface dans d'autres langues que l'anglais. |
| Menu | permettre aux administrateurs de personnaliser le menu de navigation du site. |
| Node | permet de soumettre le contenu et de l'afficher sur des pages. |
| Page | active la création de pages pouvant être ajoutées au système de navigation. |
| Path | autorise les utilisateurs à renommer les [URL](https://fr.wikipedia.org/wiki/Uniform_Resource_Locator). |
| Ping | prévient les autres sites quand votre site a été mis à jour. |
| Poll | permet à votre site de collecter des votes sur différents sujets sous la forme de questions à choix multiples. |
| Profile | support des profils utilisateurs configurables. |
| Queue | permettait à la communauté de modérer le contenu. Ce module a été retiré des modules officiels depuis la version 4.7. |
| Search | permet la recherche par mots-clés dans l'ensemble du site. |
| Statistics | prend note des statistiques d'accès pour votre site. |
| Story | permet aux utilisateurs de soumettre des récits, articles ou du contenu similaire. |
| System | manipule la configuration générale du site pour les administrateurs. |
| Taxonomy | permet l'organisation du contenu en différentes catégories. |
| throttle | se charge du mécanisme d'autorégulation, pour contrôler la congestion du site. |
| Tracker | active le suivi des messages récents pour les utilisateurs. |
| Upload | permet aux utilisateurs d'envoyer et d'attacher des fichiers au contenu. |
| User | gère l'inscription des utilisateurs et le système d'authentification. |
| Watchdog | tient un journal des événements du système. |
| Wiki | permet d'utiliser une syntaxe [wiki](https://fr.wikipedia.org/wiki/Wiki). |

#### Langage de programmation PHP [5] [9]

##### A propos

PHP : Hypertext Preprocessor, plus connu sous son sigle PHP est un [langage de programmation](http://fr.wikipedia.org/wiki/Langage_de_programmation)  principalement utilisé pour produire des [pages Web dynamiques](http://fr.wikipedia.org/wiki/Page_Web_dynamique), mais pouvant également fonctionner comme n'importe quel [langage interprété](http://fr.wikipedia.org/wiki/Langage_interpr%C3%A9t%C3%A9_(informatique)) de façon locale. PHP est un langage indispensable disposant depuis la version 5 de fonctionnalités de [modèle objet](http://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet) complètes.

##### Historique

Le langage PHP fut créé en [1994](http://fr.wikipedia.org/wiki/1994) par [Rasmus Lerdorf](http://fr.wikipedia.org/wiki/Rasmus_Lerdorf" \o "Rasmus Lerdorf) pour son [site web](http://fr.wikipedia.org/wiki/Site_web). C'était à l'origine une [bibliothèque logicielle](http://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle) en [Perl](http://fr.wikipedia.org/wiki/Perl_(langage)) (langage de programmation reprenant certaines fonctionnalités du [langage C](http://fr.wikipedia.org/wiki/Langage_C))  dont il se servait pour conserver une trace des visiteurs qui venaient consulter son [CV](http://fr.wikipedia.org/wiki/Curriculum_vit%C3%A6). Au fur et à mesure qu'il ajoutait de nouvelles fonctionnalités, Rasmus a transformé la bibliothèque en une implémentation en [langage C](http://fr.wikipedia.org/wiki/C_(langage)), capable de communiquer avec des bases de données et de créer des applications dynamiques et simples pour le [Web](http://fr.wikipedia.org/wiki/World_Wide_Web). Rasmus décida alors en [1995](http://fr.wikipedia.org/wiki/1995) de publier son code, pour que tout le monde puisse l'utiliser et en profiter. PHP s'appelait alors PHP/FI (pour ***P****ersonal****H****ome****P****age Tools/****F****orm****I****nterpreter*). En [1997](http://fr.wikipedia.org/wiki/1997), deux étudiants, [Andi Gutmans](http://fr.wikipedia.org/wiki/Andi_Gutmans) et [Zeev Suraski](http://fr.wikipedia.org/wiki/Zeev_Suraski" \o "Zeev Suraski), redéveloppèrent le cœur de PHP/FI. Ce travail aboutit un an plus tard à la version 3 de PHP, devenu alors PHP: Hypertext Preprocessor. En [2002](http://fr.wikipedia.org/wiki/2002), PHP est utilisé par plus de 8 millions de sites Web à travers le monde, en [2007](http://fr.wikipedia.org/wiki/2007) par plus de 20 millions et en 2013 par plus de 244 millions.

##### Logo de PHP



Figure 21: Logo de PHP

##### Caractéristiques et Fonctionnement

Le langage PHP est utilisé principalement en tant que [langage de script côté serveur](http://fr.wikipedia.org/wiki/Langage_serveur), ce qui veut dire que c'est le [serveur](http://fr.wikipedia.org/wiki/Serveur_Web) (la machine qui héberge la page Web en question) qui va interpréter le code PHP et générer du code (constitué généralement d'[XHTML](http://fr.wikipedia.org/wiki/XHTML) ou d'[HTML](http://fr.wikipedia.org/wiki/Hypertext_Markup_Language), de [CSS](http://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade), et parfois de [JavaScript](http://fr.wikipedia.org/wiki/JavaScript)) qui pourra être interprété par un [navigateur](http://fr.wikipedia.org/wiki/Navigateur_Web). Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour le Web.

PHP appartient à la grande famille des descendants du [C](http://fr.wikipedia.org/wiki/C_(langage)), dont la [syntaxe](http://fr.wikipedia.org/wiki/Syntaxe) est très proche. Dans une utilisation Web, l'exécution du code PHP se déroule ainsi : lorsqu'un visiteur demande à consulter une [page Web](http://fr.wikipedia.org/wiki/Page_Web), son [navigateur](http://fr.wikipedia.org/wiki/Navigateur_Web) envoie une requête au [serveur HTTP](http://fr.wikipedia.org/wiki/Serveur_HTTP) correspondant. Si la page est identifiée comme un script PHP (généralement grâce à l'extension .php), le serveur appelle l'[interprète](http://fr.wikipedia.org/wiki/Interpr%C3%A8te_(informatique)) PHP qui va traiter et générer le code final de la page (constitué généralement d'[HTML](http://fr.wikipedia.org/wiki/Hypertext_Markup_Language) ou de [XHTML](http://fr.wikipedia.org/wiki/XHTML), mais aussi souvent de [CSS](http://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade) et de [JS](http://fr.wikipedia.org/wiki/JavaScript)). Ce contenu est renvoyé au serveur HTTP, qui l'envoie finalement au client. Le schéma ci-dessous explique ce fonctionnement :



Figure 22:Fonctionnement du langage PHP

#### Langage HTML [5] [J]

##### A propos

L’[anglais](http://fr.wikipedia.org/wiki/Anglais) *Hypertext Markup Language* se traduit littéralement en langage de balisage d’hypertexte []. On utilise généralement l’abréviation HTML, parfois même en répétant le mot « langage » comme dans « langage HTML ». *Hypertext* est parfois écrit *HyperText* pour marquer le *T* de l’abréviation HTML.

Le public non averti parle parfois de HTM au lieu de HTML, HTM étant l’[extension de nom de fichier](http://fr.wikipedia.org/wiki/Extension_de_nom_de_fichier) tronquée à trois lettres, une limitation qu’on trouve sur d’anciens [systèmes d’exploitation](http://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation) de [Microsoft](http://fr.wikipedia.org/wiki/Microsoft).

##### Historique

Durant la première moitié des [années 1990](http://fr.wikipedia.org/wiki/Ann%C3%A9es_1990), avant l’apparition des technologies web comme [JavaScript](http://fr.wikipedia.org/wiki/JavaScript), les [feuilles de style en cascade](http://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade) et le [*Document Object Model*](http://fr.wikipedia.org/wiki/Document_Object_Model), l’évolution de HTML a dicté l’évolution du [*World Wide Web*](http://fr.wikipedia.org/wiki/World_Wide_Web). Depuis [1997](http://fr.wikipedia.org/wiki/1997_en_informatique) et HTML 4, l’évolution de HTML a fortement ralenti ; 10 ans plus tard, HTML 4 reste utilisé dans les pages web. En [2008](http://fr.wikipedia.org/wiki/2008_en_informatique), la spécification du [HTML 5](http://fr.wikipedia.org/wiki/HTML_5) est à l’étude [].

**1989-1992 : Origine :**

HTML est une des trois inventions à la base du [*World Wide Web*](http://fr.wikipedia.org/wiki/World_Wide_Web), avec le [*Hypertext Transfer Protocol*](http://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol) (HTTP) et les [adresses web](http://fr.wikipedia.org/wiki/Adresse_web). HTML a été inventé pour pouvoir écrire des documents [hypertextuels](http://fr.wikipedia.org/wiki/Hypertexte) liant les différentes ressources d’[Internet](http://fr.wikipedia.org/wiki/Internet) avec des [hyperliens](http://fr.wikipedia.org/wiki/Hyperlien). Aujourd’hui, ces documents sont appelés « [page web](http://fr.wikipedia.org/wiki/Page_Web) ». En [août](http://fr.wikipedia.org/wiki/Ao%C3%BBt_1991) [1991](http://fr.wikipedia.org/wiki/1991_en_informatique), lorsque [Tim Berners-Lee](http://fr.wikipedia.org/wiki/Tim_Berners-Lee) annonce publiquement le web sur [Usenet](http://fr.wikipedia.org/wiki/Usenet), il ne cite que le langage [SGML](http://fr.wikipedia.org/wiki/Standard_Generalized_Markup_Language), mais donne l’[URL](http://fr.wikipedia.org/wiki/Rep%C3%A8re_uniforme_de_ressource) d’un document de suffixe .html. Dans son livre *Weaving the web*[[3]](http://fr.wikipedia.org/wiki/Hypertext_Markup_Language#cite_note-3), [Tim Berners-Lee](http://fr.wikipedia.org/wiki/Tim_Berners-Lee) décrit la décision de baser HTML sur SGML comme étant aussi « diplomatique » que technique : techniquement, il trouvait SGML trop complexe, mais il voulait attirer la communauté [hypertexte](http://fr.wikipedia.org/wiki/Hypertexte) qui considérait que SGML était le langage le plus prometteur pour standardiser le format des documents hypertexte.

**1995-1996 : Html 2.0 :**

En [mars](http://fr.wikipedia.org/wiki/Mars_1995) [1995](http://fr.wikipedia.org/wiki/1995_en_informatique), le [*World Wide Web Consortium*](http://fr.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C) nouvellement fondé propose le résultat de ses recherches sur HTML+ : le brouillon HTML 3.0. Il comprend notamment le support des tables, des figures et des expressions mathématiques. Ce brouillon expire le [28](http://fr.wikipedia.org/wiki/28_septembre) [septembre](http://fr.wikipedia.org/wiki/Septembre_1995) [1995](http://fr.wikipedia.org/wiki/1995_en_informatique) sans donner de suites directes. Fin [1995](http://fr.wikipedia.org/wiki/1995_en_informatique), le [RFC 1866](http://tools.ietf.org/html/rfc1866) décrivant HTML 2.0 est finalisé. Le principal éditeur est [Dan Connolly](http://fr.wikipedia.org/wiki/Dan_Connolly). Ce document décrit HTML tel qu’il existait avant [juin](http://fr.wikipedia.org/wiki/Juin_1994) [1994](http://fr.wikipedia.org/wiki/1994_en_informatique), donc sans les nombreuses additions de Netscape Navigator.

**1997 : Html 3.2 et 4.0**

Le [14](http://fr.wikipedia.org/wiki/14_janvier) [janvier](http://fr.wikipedia.org/wiki/Janvier_1997) [1997](http://fr.wikipedia.org/wiki/1997), le W3C publie la spécification HTML 3.2. Elle décrit la pratique courante observée début [1996](http://fr.wikipedia.org/wiki/1996_en_informatique)[], donc avec une partie des additions de Netscape Navigator et d’Internet Explorer. Ses plus importantes nouveautés sont la standardisation des tables et de nombreux éléments de présentation. HTML 3.2 précède de peu HTML 4.0 et contient des éléments en prévision du support des styles et des scripts. Le [18](http://fr.wikipedia.org/wiki/18_d%C3%A9cembre) [décembre](http://fr.wikipedia.org/wiki/D%C3%A9cembre_1997) [1997](http://fr.wikipedia.org/wiki/1997), le W3C publie la spécification HTML 4.0 qui standardise de nombreuses extensions supportant les styles et les scripts, les cadres (*frames*) et les objets (inclusion généralisée de contenu). HTML 4.0 apporte également différentes améliorations pour l’accessibilité des contenus dont principalement la possibilité d’une séparation plus explicite entre structure et présentation du document, ou le support d’informations supplémentaires sur certains contenus complexes tels que les formulaires, les tableaux ou les sigles. HTML 4.0 introduit trois variantes du format, destinées à favoriser l’évolution vers un balisage plus signifiant, tout en tenant compte des limites temporaires des outils de production :

* la variante stricte (*strict*) exclut des éléments et attributs dits « de présentation », destinés à être remplacés par les styles CSS, ainsi que les éléments applet et frame qui sont remplacés par l’élément object réputé plus apte à l’interopérabilité et à l’accessibilité ;
* la variante transitoire (*transitional*) étend la variante stricte en reprenant les éléments et attributs dépréciés de HTML 3.2, dont les éléments de présentation sont couramment utilisés par les [éditeurs HTML](http://fr.wikipedia.org/wiki/%C3%89diteur_HTML) de l’époque ;
* la variante *frameset* normalise la technique des jeux de cadres composant une ressource unique à partir de plusieurs pages web assemblées par le navigateur.

Ces variantes perdurent par la suite sans modifications notables en HTML 4.01 et dans le format de transition XHTML 1.0 issu de HTML. La dernière spécification de HTML est la 4.01 datant du [24](http://fr.wikipedia.org/wiki/24_d%C3%A9cembre) [décembre](http://fr.wikipedia.org/wiki/D%C3%A9cembre_1999) [1999](http://fr.wikipedia.org/wiki/1999_en_informatique). Elle n’apporte que des corrections mineures à la version 4.0.

**2007 à nos jours : Html5 et abandon de XHtml 2 :**

En [mars](http://fr.wikipedia.org/wiki/Mars_2007) [2007](http://fr.wikipedia.org/wiki/2007_en_informatique), tirant la conséquence des réticences d’une partie de l’industrie et des concepteurs de contenus web face à XHTML 2.0[], le W3C relance le développement de HTML et crée un nouveau groupe de travail encadré par Chris Wilson ([Microsoft](http://fr.wikipedia.org/wiki/Microsoft)) et initialement [Dan Connolly](http://fr.wikipedia.org/wiki/Dan_Connolly) ([W3C](http://fr.wikipedia.org/wiki/World_Wide_Web_Consortium)), maintenant [Michael Smith](http://fr.wikipedia.org/wiki/Michael_Smith) ([W3C](http://fr.wikipedia.org/wiki/World_Wide_Web_Consortium)). Il s’agit notamment :

* de faire évoluer HTML pour décrire la sémantique des documents mais aussi les applications en ligne ;
* de parvenir à un langage extensible via XML tout en maintenant une version non XML compatible avec les analyseurs syntaxiques (*parsers*) HTML des navigateurs actuels ;
* et d’enrichir les interfaces utilisateurs avec des contrôles spécifiques : barres de progrès, menus, champs associés à des types de données spécifiques.

##### Quelques caractéristiques du HTML

* Le langage HTML est un langage permettant de définir la structure logique d'un document à l'aide de différents éléments de base (titres, paragraphes, liens vers d'autres documents, ...) ;
* Cette structuration se fait à l'aide d'indications de mise en page (que nous appellerons directives dans la suite du document). En règle générale, une directive s'écrit sous la forme:  
  **<TAG>**texte sur lequel porte la directive **</TAG>** où **TAG** est le nom de la directive ;
* HTML n'est pas un langage de traitement de texte en ce sens qu'il ne permet pas de décrire les détails de mise en forme des divers éléments composant un document ;
* La mise en forme dépend du navigateur et des options choisies par le lecteur. Le résultat de la visualisation peut donc varier mais la structuration logique est préservée dans tous les cas ;
* HTML permet d'écrire des hypertextes grâce à des liens pointant sur d'autres documents qui peuvent être des documents HTML ou des documents d'une autre nature. Ces documents sont accédés grâce à leur URL.

##### Structure général d’un document HTML

Voici un document minimaliste mais ... complet :

<HTML>

<HEAD>

<TITLE> Titre de la page. </TITLE>

</HEAD>

<BODY>

Contenu de la page.

</BODY>

</HTML>

On voit qu'un document HTML se divise en deux parties:

**L’en-tête (<HEAD> ... </HEAD>)**

Il doit comprendre un titre (pas trop long) qui sera utilisé dans les **signets** (**bookmarks**) pour désigner le document.

**Attention :** l'en-tête n'étant pas forcément visualisé, pensez à éventuellement reprendre le titre dans le corps du document.

**Le corps (<BODY> ... </BODY>)**

Il contient le document structuré. Toutes les indications de mises en page ou directives que nous verrons ultérieurement sont à placer dans le corps du document.

#### Langage JavaScript [4] [K]

##### Définition

Le **JavaScript** est un langage informatique utilisé sur les pages web. C’est aussi un langage de script orienté objet principalement utilisé dans les pages [HTML](http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-html-480/). Ce langage à la particularité de s'activer sur le poste client, en d'autres mots c'est votre ordinateur qui va recevoir le code et qui devra l'exécuter. C'est en opposition à d'autres langages qui sont activé côté serveur. L'exécution du code est effectuée par votre navigateur internet tel que Firefox ou Internet Explorer.

L'une des choses primordiale à savoir est de bien se rendre compte que le JavaScript ***n'a aucun rapport avec le Java qui est un autre langage informatique***.

##### Historique

Le langage a été créé en dix jours en mai 1995 par [Brendan Eich](https://fr.wikipedia.org/wiki/Brendan_Eich) pour le compte de la [Netscape Communications Corporation](https://fr.wikipedia.org/wiki/Netscape_Communications_Corporation), qui s'est inspiré de nombreux langages, notamment de [Java](https://fr.wikipedia.org/wiki/Java_%28langage%29) mais en simplifiant la syntaxe pour les débutants.

##### Particularités

La particularité du JavaScript consiste à créer des petits scripts sur une page [HTML](http://glossaire.infowebmaster.fr/html/) dans le but d'ajouter une petite animation ou un effet particulier sur la page. Cela permet en général d'améliorer l'ergonomie ou l'interface utilisateur, mais certains scripts sont peu utile et servent surtout à ajouter un effet esthétique à la page.

L'intérêt du JavaScript est d'exécuté un code sans avoir à recharger une nouvelle fois la page.

La technique AJAX (Asynchronous JavaScript And [XML](http://glossaire.infowebmaster.fr/xml/)) utilise grandement le JavaScript dans le but d'interagir sur la page de manière dynamique.

##### Versions du langage

Les versions récentes du langage JavaScript ont pour origine les spécifications de la norme ECMA-262 définissant ECMAScript. JavaScript est un sur-ensemble d'ECMAScript développé par la fondation Mozilla et par Adobe lui ajoutant les fonctionnalités suivantes :

* **Version 1.5 :** Interpréteur basé sur les spécifications ECMA-262 3e édition.
* **Version 1.6 :** 
  + extension de l'objet Array ;
  + rapprochement des objets String et Array.
* **Version 1.7 :** 
  + générateurs (instruction yield) ;
  + itérateurs ;
  + définition de tableaux par compréhension (var evens = [i for (i in range(0, 21)) if (i% 2 == 0)];) ;
  + définition de portée locale (instruction let) ;
  + assignation déstructurante (renvoi de valeurs multiples : [a,b] = maFonction()).
* **Version 1.8 :**
  + extension des fermetures d'expressions ;
  + extension des expressions génératrices ;
  + plus d'extras pour les tableaux.
* **Version 2.0 :** Le développement d'une version 2 de JavaScript basée sur les spécifications du langage [ECMAScript](https://fr.wikipedia.org/wiki/ECMAScript) Édition 4 en collaboration avec [Adobe](https://fr.wikipedia.org/wiki/Adobe_Systems), le [projet Tamarin](https://fr.wikipedia.org/wiki/Projet_Tamarin), a aujourd'hui été abandonné. La norme un temps rédigée par l'[ECMA](https://fr.wikipedia.org/wiki/Ecma_International) dans le document ECMA-262 4e édition est en effet obsolète et a été remplacée par ES5.

#### MySQL [6] [7] [L]

##### Définition

MySQL est un [**système de gestion de bases de données**](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es) **relationnelles** (SGBDR). Il est distribué sous une double licence [GPL](https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU) et [propriétaire](https://fr.wikipedia.org/wiki/Logiciel_propri%C3%A9taire). Il fait partie des logiciels de gestion de [base de données](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es) les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec [Oracle](https://fr.wikipedia.org/wiki/Oracle_Database), [Informix](https://fr.wikipedia.org/wiki/Informix) et [Microsoft SQL Server](https://fr.wikipedia.org/wiki/Microsoft_SQL_Server).



Figure 23: Logo MySQL (source: Google image)

##### Historique

La première version de MySQL est apparue le 23 mai 1995. Il a d'abord été créé pour un usage personnel à partir de [mSQL](https://fr.wikipedia.org/wiki/MSQL) en s'appuyant sur le langage de bas niveau [ISAM](https://fr.wikipedia.org/wiki/Organisation_s%C3%A9quentielle_index%C3%A9e) qu'ils trouvaient trop lent et trop rigide. Ils ont créé une nouvelle interface [SQL](https://fr.wikipedia.org/wiki/Structured_Query_Language) en gardant la même [API](https://fr.wikipedia.org/wiki/Interface_de_programmation) que mSQL.

MySQL est passé en [licence GPL](https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU) à partir de la version 3.23.19 (juin 2000) :

* Version 4.0 : première version en octobre 2001, [stable](https://fr.wikipedia.org/wiki/Version_d%27un_logiciel#Version_finale_.28ou_stable.29) depuis mars 2003 ;
* Version 4.1 : première version en avril 2003, stable depuis octobre 2004 ;
* Version 5.0 : première version en décembre 2003, stable depuis octobre 2005 ;
* Version 5.1 : première version en novembre 2005, [Release Candidate](https://fr.wikipedia.org/wiki/Version_d%27un_logiciel#Release_candidate_.28RC.29_ou_pre-release) distribuée depuis septembre 2007 ;
* Version 5.2 : distribuée en avant-première (ajout du nouveau [moteur](https://fr.wikipedia.org/wiki/Moteur_de_base_de_donn%C3%A9es) de stockage *Falcon*) en février 2007, cette ligne a ensuite été renommée 6.0 ;
* Version 5.5 : Version stable depuis octobre 2010 ;
* Version 5.6 : Version stable depuis février 2013 ;
* Version 5.7 : Version stable depuis octobre 2015 ;
* Version 6.0 : première version [alpha](https://fr.wikipedia.org/wiki/Version_d%27un_logiciel#Version_alpha) en avril 2007, abandonnée depuis le rachat de MySQL par oracle en décembre 2010.

##### Caractéristiques

* MySQL est un serveur de [bases de données relationnelles](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_relationnelle) [SQL](https://fr.wikipedia.org/wiki/Structured_Query_Language) développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est [**multi-thread**](https://fr.wikipedia.org/wiki/Processus_l%C3%A9ger) et **multi-utilisateur**;
* C'est un [**logiciel libre**](https://fr.wikipedia.org/wiki/Logiciel_libre), [open source](https://fr.wikipedia.org/wiki/Open_source), développé sous double licence selon qu'il est distribué avec un produit libre ou avec un produit propriétaire. Dans ce dernier cas, la licence est payante, sinon c'est la [licence publique générale GNU](https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU) (GPL) qui s'applique ;
* MySQL fonctionne sur de nombreux [systèmes d'exploitation](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation) différents, incluant [AIX](https://fr.wikipedia.org/wiki/AIX), IBM i-5, [BSDi](https://fr.wikipedia.org/w/index.php?title=BSDi&action=edit&redlink=1), [FreeBSD](https://fr.wikipedia.org/wiki/FreeBSD), [HP-UX](https://fr.wikipedia.org/wiki/HP-UX), [Linux](https://fr.wikipedia.org/wiki/Linux), [Mac OS X](https://fr.wikipedia.org/wiki/Mac_OS_X), [NetWare](https://fr.wikipedia.org/wiki/Novell_NetWare), [NetBSD](https://fr.wikipedia.org/wiki/NetBSD), [OpenBSD](https://fr.wikipedia.org/wiki/OpenBSD), [OS/2](https://fr.wikipedia.org/wiki/OS/2) Warp, [SGI](https://fr.wikipedia.org/wiki/Silicon_Graphics) [IRIX](https://fr.wikipedia.org/wiki/IRIX), [Solaris](https://fr.wikipedia.org/wiki/Solaris_%28syst%C3%A8me_d%27exploitation%29), [SCO](https://fr.wikipedia.org/wiki/SCO_Group) [OpenServer](https://fr.wikipedia.org/wiki/SCO_OpenServer), SCO [UnixWare](https://fr.wikipedia.org/wiki/UnixWare), [Tru64](https://fr.wikipedia.org/wiki/Tru64) Unix, [Windows](https://fr.wikipedia.org/wiki/Microsoft_Windows) ;
* Les bases de données sont accessibles en utilisant les [langages de programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation) [C](https://fr.wikipedia.org/wiki/C_%28langage%29), [C++](https://fr.wikipedia.org/wiki/C%2B%2B), [VB](https://fr.wikipedia.org/wiki/Visual_Basic), [VB .NET](https://fr.wikipedia.org/wiki/Visual_Basic_.NET), [C#](https://fr.wikipedia.org/wiki/C_sharp), [Delphi/Kylix](https://fr.wikipedia.org/wiki/CodeGear_Delphi), [Eiffel](https://fr.wikipedia.org/wiki/Eiffel_%28langage%29), [Java](https://fr.wikipedia.org/wiki/Java_%28langage%29), [Perl](https://fr.wikipedia.org/wiki/Perl_%28langage%29), [PHP](https://fr.wikipedia.org/wiki/PHP), [Python](https://fr.wikipedia.org/wiki/Python_%28langage%29), [WinDev](https://fr.wikipedia.org/wiki/Windev), [Ruby](https://fr.wikipedia.org/wiki/Ruby) et [Tcl](https://fr.wikipedia.org/wiki/Tool_Command_Language) ; une [API](https://fr.wikipedia.org/wiki/Application_Programming_Interface) spécifique est disponible pour chacun d'entre eux. Une interface [ODBC](https://fr.wikipedia.org/wiki/Open_Database_Connectivity) appelée [MyODBC](https://fr.wikipedia.org/w/index.php?title=MyODBC&action=edit&redlink=1) est aussi disponible.

### Outils de développement

#### Xampp

##### Définition

XAMPP est *un ensemble de* [*logiciels*](https://fr.wikipedia.org/wiki/Logiciel) permettant de mettre en place facilement un [serveur Web confidentiel](https://fr.wikipedia.org/w/index.php?title=Serveur_Web_confidentielle&action=edit&redlink=1), un [serveur FTP](https://fr.wikipedia.org/wiki/Serveur_FTP) et un [serveur de messagerie électronique](https://fr.wikipedia.org/wiki/Serveur_de_messagerie_%C3%A9lectronique). Il s'agit d'une distribution de [logiciels libres](https://fr.wikipedia.org/wiki/Logiciel_libre) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les [systèmes d'exploitation](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation) les plus répandus.



Figure 24: Logo de Xampp (source: [www.apachefriends.org](http://www.apachefriends.org))

##### Descriptions

Tableau 6: Tableau de description de Xampp (source: Wikipédia)

|  |  |
| --- | --- |
| Développeur | Apache Friends |
| Dernière version | 5.6.20 (21 Avril 2016) |
| Environnement | * Linux * Windows * Mac OS X * Solaris |
| Type | WAMP |
| Licence | Licence publique générale GNU |
| Site Web | |  |  | | --- | --- | |  | [www.apachefriends.org](http://www.apachefriends.org/fr/xampp.html) | |

#### Sublime Text [M]

##### Définition

Sublime Text est un [**éditeur de texte**](https://fr.wikipedia.org/wiki/%C3%89diteur_de_texte) générique codé en [C++](https://fr.wikipedia.org/wiki/C%2B%2B) et [Python](https://fr.wikipedia.org/wiki/Python_%28langage%29), disponible sur [Windows](https://fr.wikipedia.org/wiki/Windows), [Mac](https://fr.wikipedia.org/wiki/Mac_OS) et [Linux](https://fr.wikipedia.org/wiki/Linux). Le logiciel a été conçu tout d'abord comme une extension pour [Vim](https://fr.wikipedia.org/wiki/Vim), riche en fonctionnalités.



Figure 25: Logo de Sublime Text (source: Wikipédia)

##### Historiques

En 2007, son créateur, [Jon Skinner](https://fr.wikipedia.org/w/index.php?title=Jon_Skinner&action=edit&redlink=1), quitta son travail chez [Google](https://fr.wikipedia.org/wiki/Google) pour poursuivre un de ses rêves : créer un meilleur éditeur de texte. Il se donna trois principes pour ce logiciel:

* Discret, interface minimum : on doit pouvoir se focaliser sur le texte et non une myriade de barres d'outils ;
* Ne pas cacher le texte par des fenêtres ;
* Utiliser toute la place possible : plein écran, multi-écrans, édition de fichiers côte à côte devraient être possibles.

Depuis la version 2.0, sortie le [26](https://fr.wikipedia.org/wiki/26_juin) [juin](https://fr.wikipedia.org/wiki/Juin_2012) [2012](https://fr.wikipedia.org/wiki/2012_en_informatique), l'éditeur prend en charge 44 [langages de programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation) majeurs, tandis que des plugins sont souvent disponibles pour les langages plus rares.

#### ArgoUML [N]

##### Définition

ArgoUML est un [logiciel libre](https://fr.wikipedia.org/wiki/Logiciel_libre) de **création de diagrammes** [**UML**](https://fr.wikipedia.org/wiki/Unified_Modeling_Language), programmé en [Java](https://fr.wikipedia.org/wiki/Java_%28langage%29). Il est multilingue, supporte la génération de code et l'[ingénierie inverse](https://fr.wikipedia.org/wiki/Ing%C3%A9nierie_inverse).



Figure 26: Logo de l'application ArgoUML (source: Wikipédia)

##### Fonctionnalité

* ArgoUML supporte sept types de diagramme : [cas d'utilisation](https://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation), [classes](https://fr.wikipedia.org/wiki/Diagramme_de_classes), [séquence](https://fr.wikipedia.org/wiki/Diagramme_de_s%C3%A9quence), [état](https://fr.wikipedia.org/wiki/Diagramme_%C3%A9tats-transitions), [collaboration](https://fr.wikipedia.org/wiki/Diagramme_de_collaboration), [activité](https://fr.wikipedia.org/wiki/Diagramme_d%27activit%C3%A9) et [déploiement](https://fr.wikipedia.org/wiki/Diagramme_de_d%C3%A9ploiement) ;
* La génération de code à partir de diagrammes de classes est supportée dans les langages suivants : [Java](https://fr.wikipedia.org/wiki/Java_%28langage%29), [C++](https://fr.wikipedia.org/wiki/C%2B%2B), [PHP](https://fr.wikipedia.org/wiki/PHP), [C#](https://fr.wikipedia.org/wiki/C_sharp) et [SQL](https://fr.wikipedia.org/wiki/Structured_Query_Language).

# Résultat obtenu

### Présentation des interfaces graphiques

#### Page d’accueil



Figure 27: Page d'accueil de G-Sante

#### Page « Organigramme » de l’établissement



Figure 28: Page "Organigramme de l'établissement"

#### Page « G-Santé TV »

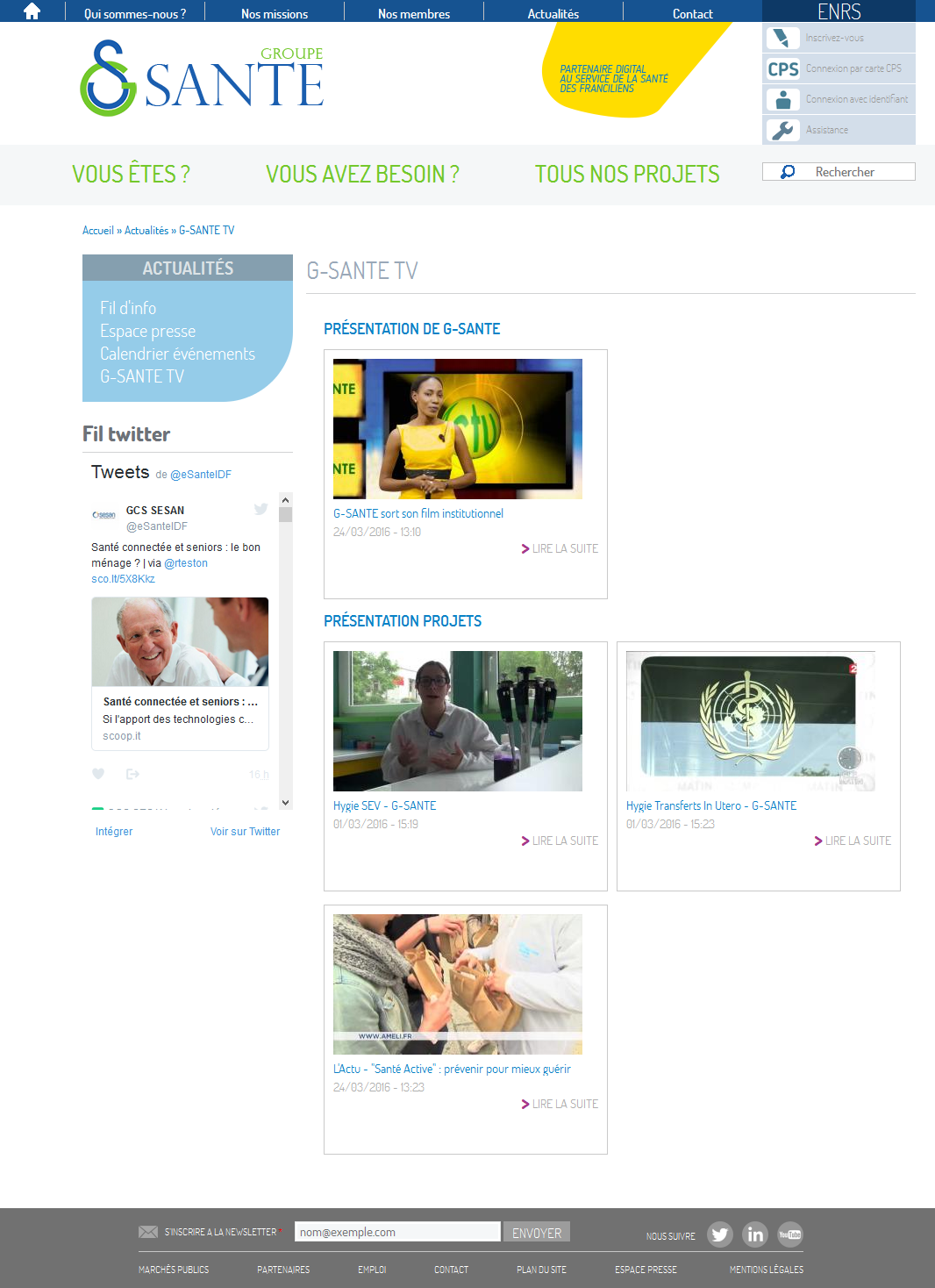


Figure 29: Page G-Santé TV

#### Page « Nos partenaires »



Figure 30: Page "Nos partenaires"

#### Page « Contact »

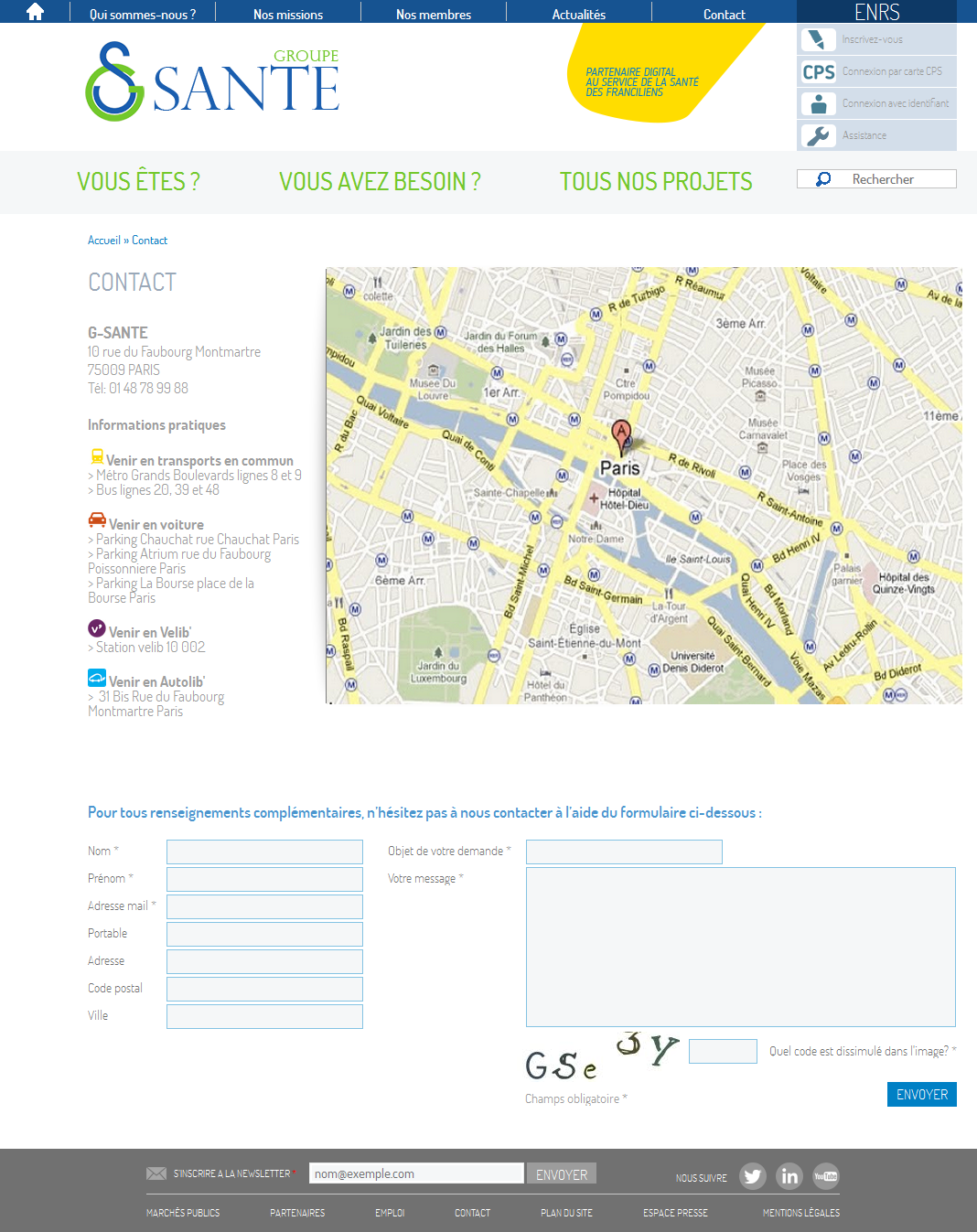


Figure 31: Page "Contact"

#### Page « Carte des membres G-SANTE »

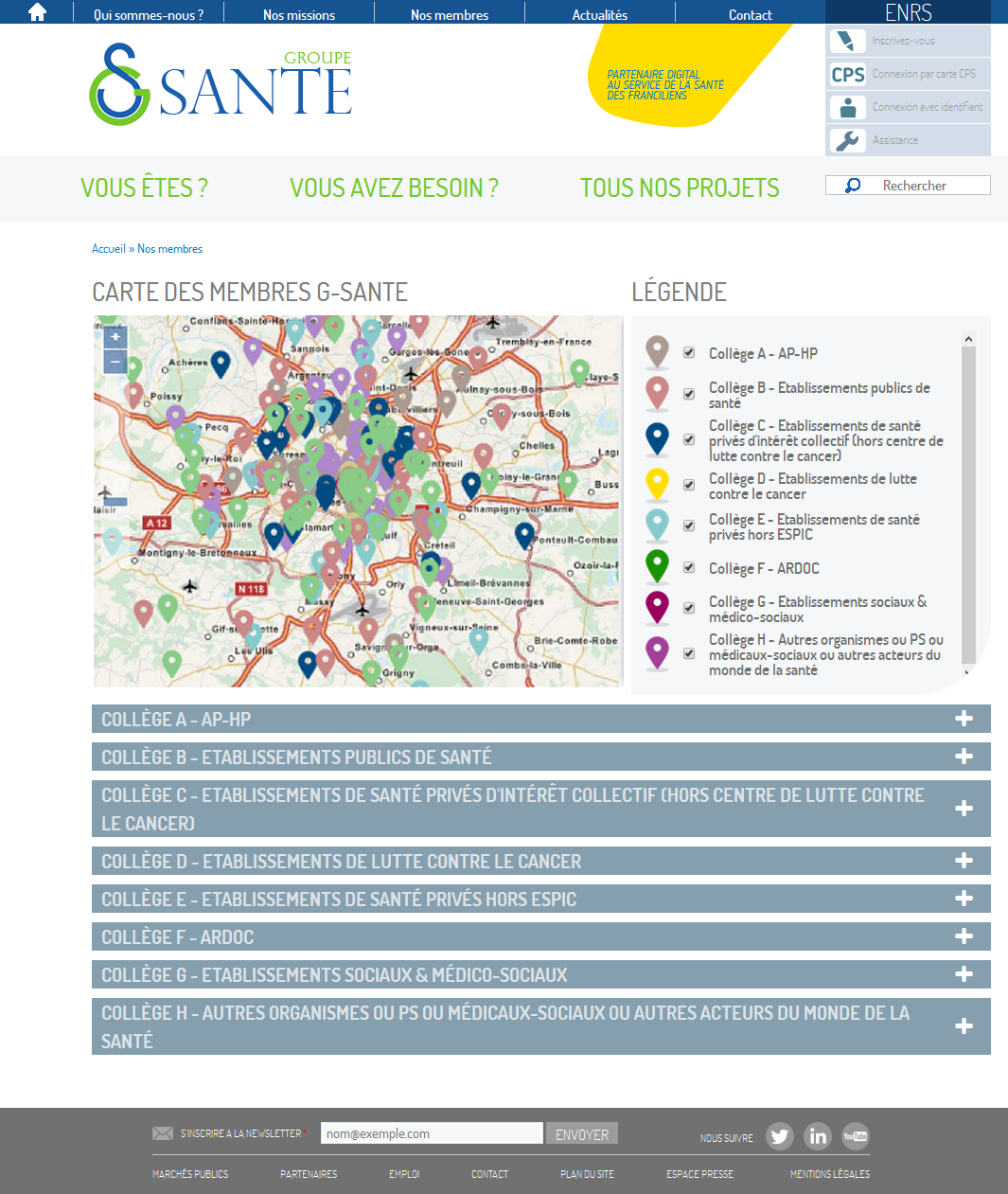


Figure 32: Page "Carte des membres G-SANTE"

#### Page « Calendrier des événements »

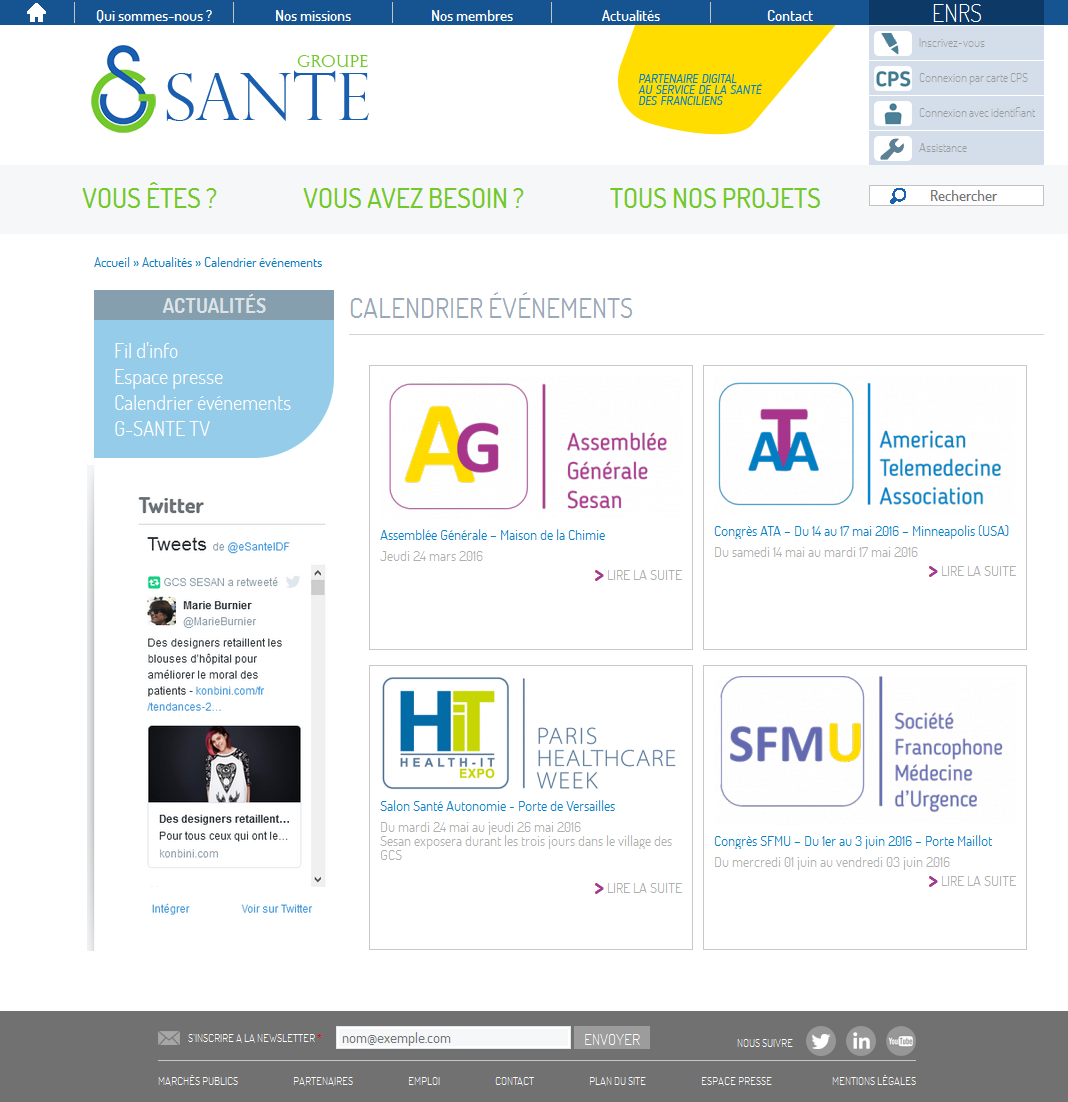


Figure 33: Page "Calendrier des événements"

### Extrait de code

#### html.tpl.php

<!DOCTYPE html>

<html lang=**"**<?php **print** $language->language; ?>**"** dir=**"**<?php **print** $language->**dir**; ?>**"**<?php **print** $rdf\_namespaces;?> xmlns:fb=**"http://ogp.me/ns/fb#"**>

<head profile=**"**<?php **print** $grddl\_profile; ?>**"**>

<meta charset=**"utf-8"** />

<meta http-equiv=**"X-UA-Compatible"** content=**"IE=edge"** />

<meta name=**"viewport"** content=**"width=device-width, initial-scale=1"** />

<?php **print** $head; ?>

<title><?php **print** $head\_title; ?></title>

<?php **print** $styles; ?>

<?php **print** $scripts; ?>

</head>

<body class=**"**<?php **print** $classes; ?>**"** <?php **print** $attributes;?>>

<div id=**"skip-link"**>

<a href=**"#main-content"** class=**"element-invisible element-focusable"**><?php **print** t('Skip to main content'); ?></a>

</div>

<?php **print** $page\_top; ?>

<?php **print** $page; ?>

<?php **print** $page\_bottom; ?>

</body>

</html>

#### page.tpl.php

<div id=**"page-wrapper"**>

<div id=**"page"**>

<div id=**"header"** class=**"**<?php **print** $secondary\_menu ? 'with-secondary-menu': 'without-secondary-menu'; ?>**"**>

<div class=**"header-top"**>

<div class=**"menu-top"**>

<div class=**"section"**>

<?php **print** render($page['menuhaut']); ?>

</div>

</div>

<div class=**"header-mid"**>

<div class=**"section clearfix"**>

<div class=**"col left"**>

<?php **if** ($logo): ?>

<a href=**"**<?php **print** $front\_page; ?>**"** title=**"**<?php **print** t('Home'); ?>**"** rel=**"home"** id=**"logo"**>

<img src=**"**<?php **print** $logo; ?>**"** alt=**"**<?php **print** t('Home'); ?>**"** />

</a>

<?php **endif**; ?>

</div>

<div class=**"col right"**>

<?php **print** render($page['menu\_enrs']); ?>

</div>

</div>

</div>

</div>

<div class=**"header-bottom"**>

<div class=**"section"**>

<?php **print** render($page['header']); ?>

</div>

</div>

<?php **if** ($secondary\_menu): ?>

<div id=**"secondary-menu"** class=**"navigation"**>

<?php **print** theme('links\_\_system\_secondary\_menu', **array**(

'links' => $secondary\_menu,

'attributes' => **array**(

'id' => 'secondary-menu-links',

'class' => **array**('links', 'inline', 'clearfix'),

),

'heading' => **array**(

'text' => t('Secondary menu'),

'level' => 'h2',

'class' => **array**('element-invisible'),

),

)); ?>

</div><!-- /#secondary-menu -->

<?php **endif**; ?>

</div><!-- /.section, /#header -->

<?php **if** ($messages): ?>

<div id=**"messages"**><div class=**"section clearfix"**>

<?php **print** $messages; ?>

</div></div><!-- /.section, /#messages -->

<?php **endif**; ?>

<?php **if** ($page['featured']): ?>

<div id=**"featured"**><div class=**"section clearfix"**>

<?php **print** render($page['featured']); ?>

</div></div><!-- /.section, /#featured -->

<?php **endif**; ?>

<div id=**"main-wrapper"** class=**"**<?php **echo** (**isset**($node) && $node->nid==4198) ? 'main-wrapper-home' : '' ?>**"**>

<div id=**"main"** class=**"clearfix"**>

<?php

**if**((**isset**($node) && $node->nid!=4198)&& $node->nid!=5739 || !**isset**($node)):?>

<?php **if** ($breadcrumb): ?>

<div id=**"breadcrumb"** class=**"clearfix"**><?php **print** $breadcrumb; ?></div>

<?php **endif**;?>

<?php **endif**;?>

<?php

**if**($page['addthis\_top']) :

**print** render($page['addthis\_top']);

**endif**;

?>

<div class=**"clearfix"**></div>

<?php **if** ($page['sidebar\_first']): ?>

<div id=**"sidebar-first"** class=**"column sidebar"**><div class=**"section"**>

<?php **print** render($page['sidebar\_first']); ?>

</div></div><!-- /.section, /#sidebar-first -->

<?php **endif**; ?>

<?php

**if**(**isset**($node)):

**if**($node->nid!=4198):

**if**($node->type=="projet"):

?>

<div class=**"header-projet"**>

<h2 class=**"node-title"**>**Projets**</h2>

</div>

<?php

**endif**;

**endif**;

**endif**;

?>

<div id=**"content"** class=**"clearfix column** <?php **echo** (**isset**($node) && $node->nid==4198 ? 'content-hp' : '') ?>**"**>

<div class=**"section** <?php **echo** (**isset**($node) && $node->nid==4198 ? 'section-actu-hp' : '') ?>**"**>

<?php **if** ($page['highlighted']): ?><div id=**"highlighted"**><?php **print** render($page['highlighted']); ?></div><?php **endif**; ?>

<a id=**"main-content"**></a>

<?php **print** render($title\_prefix); ?>

<?php

**if**(**isset**($node)):

**if**($node->nid!=4198):

**if**($title):

**if**($node->type=="article"):

?>

<h2 class=**"title\_actu node-title"** id=**"page-title-actu"**>**Actualités**</h2>

<h3 class=**"title"** id=**"page-title"**><?php **print** $title; ?></h3>

<?php

**endif**;

?>

<?php **if**($node->type!="article"): ?>

<h1 class=**"title"** id=**"page-title"**><?php **print** $title; ?></h1>

<?php **endif** ?>

<?php

**endif**;

**endif**;

**endif**;

?>

<?php **if**(**isset**($node)){

**if**($node->nid==4198){?>

<h1 class=**"actu-hp-title"**><a href = **"**<?php **print** base\_path().drupal\_get\_path\_alias('node/5231'); ?>**"**>**Actualités**</a></h1>

<?php

}

}

?>

<?php **print** render($title\_suffix); ?>

<?php **if** ($tabs): ?>

<div class=**"tabs"**>

<?php **print** render($tabs); ?>

</div>

<?php **endif**; ?>

<?php **print** render($page['help']); ?>

<?php **if** ($action\_links): ?>

<ul class=**"action-links"**>

<?php **print** render($action\_links); ?>

</ul>

<?php **endif**; ?>

<?php

**if**($page['openlayer']):

**print** render($page['openlayer']);

**endif**;

**print** render($page['content']);

?>

<?php

**if**($page['actu\_hp']) :

**print** render($page['actu\_hp']);

**endif**;

?>

<?php **print** $feed\_icons; ?>

</div></div><!-- /.section, /#content -->

<?php **if** ($page['sidebar\_second']): ?>

<div id=**"sidebar-second"** class=**"column sidebar** <?php **echo** (**isset**($node) && $node->nid==4198 ? 'sidebar-hp' : '') ?>**"**>

<div class=**"section"**>

<?php **print** render($page['sidebar\_second']); ?>

</div>

</div><!-- /.section, /#sidebar-second -->

<?php **endif**; ?>

</div></div><!-- /#main, /#main-wrapper -->

<?php **if** ($page['triptych\_first'] || $page['triptych\_middle'] || $page['triptych\_last']): ?>

<div id=**"triptych-wrapper"**><div id=**"triptych"** class=**"clearfix"**>

<?php **print** render($page['triptych\_first']); ?>

<?php **print** render($page['triptych\_middle']); ?>

<?php **print** render($page['triptych\_last']); ?>

</div></div><!-- /#triptych, /#triptych-wrapper -->

<?php **endif**; ?>

<div id=**"footer-wrapper"**>

<?php **if** ($page['footer\_firstcolumn'] || $page['footer\_secondcolumn'] || $page['footer\_thirdcolumn'] || $page['footer\_fourthcolumn']): ?>

<div id=**"footer-columns"** class=**"clearfix"**>

<?php **print** render($page['footer\_firstcolumn']); ?>

<?php **print** render($page['footer\_secondcolumn']); ?>

<?php **print** render($page['footer\_thirdcolumn']); ?>

<?php **print** render($page['footer\_fourthcolumn']); ?>

</div><!-- /#footer-columns -->

<?php **endif**; ?>

<?php **if** ($page['footer']): ?>

<div id=**"footer"** class=**"clearfix section"**>

<?php **print** render($page['footer']); ?>

</div><!-- /#footer -->

<?php **endif**; ?>

</div><!-- /.section, /#footer-wrapper -->

</div>

<?php

**if**($page['filter\_project']):

**print** render($page['filter\_project']);

**endif**;

?>

</div><!-- /#page, /#page-wrapper -->

<a class=**"back-top"** href=**"#"**>*&rsaquo;&rsaquo;*</a>

#### node.tpl.php

<div id=**"node-**<?php **print** $node->nid; ?>**"** class=**"**<?php **print** $classes; ?> **clearfix"**<?php **print** $attributes; ?>>

<?php **print** render($title\_prefix); ?>

<?php **if** (!$page): ?>

<h2<?php **print** $title\_attributes; ?>>

<a href=**"**<?php **print** $node\_url; ?>**"**><?php **print** $title; ?></a>

</h2>

<?php **endif**; ?>

<?php **print** render($title\_suffix); ?>

<?php **if** ($display\_submitted): ?>

<div class=**"meta submitted"**>

<?php **print** $user\_picture; ?>

<?php **print** $submitted; ?>

</div>

<?php **endif**; ?>

<div class=**"content clearfix"**<?php **print** $content\_attributes; ?>>

<?php

// We hide the comments and links now so that we can render them later.

hide($content['comments']);

hide($content['links']);

**print** render($content);

?>

</div>

<?php

// Remove the "Add new comment" link on the teaser page or if the comment

// form is being displayed on the same page.

**if** ($teaser || !**empty**($content['comments']['comment\_form'])) {

**unset**($content['links']['comment']['#links']['comment-add']);

}

// Only display the wrapper div if there are links.

$links = render($content['links']);

**if** ($links):

?>

<div class=**"link-wrapper"**>

<?php **print** $links; ?>

</div>

<?php **endif**; ?>

<?php **print** render($content['comments']); ?>

</div>

#### node--webform.tpl.php

<div id=**"node-**<?php **print** $node->nid; ?>**"** class=**"**<?php **print** $classes; ?> **clearfix"**<?php **print** $attributes; ?>>

<?php **print** render($title\_prefix); ?>

<?php **if** (!$page): ?>

<h2<?php **print** $title\_attributes; ?>>

<a href=**"**<?php **print** $node\_url; ?>**"**><?php **print** $title; ?></a>

</h2>

<?php **endif**; ?>

<?php **print** render($title\_suffix); ?>

<?php **if** ($display\_submitted): ?>

<!--<div class="meta submitted">

<?php **print** $user\_picture; ?>

<?php **print** $submitted; ?>

</div>-->

<?php **endif**; ?>

<?php **if**(**isset**($intro)):?>

<p class=**"webform-intro"**><?php **print** $intro; ?></p>

<?php **endif**; ?>

<div class=**"content clearfix"**<?php **print** $content\_attributes; ?>>

<?php

// We hide the comments and links now so that we can render them later.

hide($content['comments']);

hide($content['links']);

**print** render($content);

?>

</div>

<?php

// Remove the "Add new comment" link on the teaser page or if the comment

// form is being displayed on the same page.

**if** ($teaser || !**empty**($content['comments']['comment\_form'])) {

**unset**($content['links']['comment']['#links']['comment-add']);

}

// Only display the wrapper div if there are links.

$links = render($content['links']);

**if** ($links):

?>

<div class=**"link-wrapper"**>

<?php **print** $links; ?>

</div>

<?php **endif**; ?>

<?php **print** render($content['comments']); ?>

</div>

#### block--views--actualites-block-2.tpl.php

<div id=**"**<?php **print** $block\_html\_id; ?>**"** class=**"**<?php **print** $classes; ?>**"**<?php **print** $attributes; ?>>

<?php **print** render($title\_prefix); ?>

<?php **if** ($block->subject): ?>

<input type=**"checkbox"** id=**"check\_toggle\_actu\_projet"** class=**"check-actu-projet"** />

<h2<?php **print** $title\_attributes; ?>><label for=**"check\_toggle\_actu\_projet"**><?php **print** $block->subject ?></label></h2>

<?php **endif**;?>

<?php **print** render($title\_suffix); ?>

<div class=**"content"**<?php **print** $content\_attributes; ?>>

<?php **print** $content ?>

</div>

</div>

#### views-view-fields--organigramme.tpl.php

<div class=**"organigramme-item-wrapper"**>

<?php **foreach** ($fields **as** $id => $field): ?>

<?php **if** (!**empty**($field->separator)): ?>

<?php **print** $field->separator; ?>

<?php **endif**; ?>

<?php **print** $field->wrapper\_prefix; ?>

<?php **print** $field->label\_html; ?>

<?php **print** $field->content; ?>

<?php **print** $field->wrapper\_suffix; ?>

<?php **endforeach**; ?>

</div>

#### views-view-field--field\_photo\_interlocuteur.tpl.php

<div class=**"org-photo"** data-url=**"**<?php **print** drupal\_get\_path\_alias("node/".$row->nid); ?>**?width=640&height=480"**>

<?php **if**($row->field\_data\_field\_niveau\_field\_niveau\_value!=1) {?>

<a class=**"colorbox-node"** href = **"**<?php **print** drupal\_get\_path\_alias("node/".$row->nid); ?>**?width=620&height=435"**>

<?php **print** $output; ?>

</a>

<?php }**else**{

**print** $output;

}?>

</div>

#### views-view-field--field\_enrs\_lien.tpl.php

<?php

$options = **array**('absolute' => **TRUE**);

$nid = $row->nid; // Node ID

$url = url('node/' . $nid, $options);

?>

<?php **if** ($nid==4688 || $nid==4687){ ?>

<a class=**"colorbox-node"** href=**"**<?php **print** $url; ?>**?width=730&height=500"**>

<?php **print** $output; ?>

</a>

<?php }**else**{

$node=node\_load($nid);

$lien\_url=$node->field\_enrs\_lien["und"][0]["url"];

?>

<a href=**"**<?php **print** $lien\_url; ?>**"** target=**"\_blank"**>

<?php **print** $output; ?>

</a>

<?php } ?>

#### index.php

<?php

/\*\*

\* @file

\* The PHP page that serves all page requests on a Drupal installation.

\*

\* The routines here dispatch control to the appropriate handler, which then

\* prints the appropriate page.

\*

\* All Drupal code is released under the GNU General Public License.

\* See COPYRIGHT.txt and LICENSE.txt.

\*/

/\*\*

\* Root directory of Drupal installation.

\*/

**define**('DRUPAL\_ROOT', **getcwd**());

**require\_once** DRUPAL\_ROOT . '/includes/bootstrap.inc';

drupal\_bootstrap(DRUPAL\_BOOTSTRAP\_FULL);

menu\_execute\_active\_handler();

# Discussion

### Etude du système d’information

#### Introduction

Le [système d’information](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%E2%80%99information) est un élément essentiel de l’entreprise de nos jours. Les entreprises sont prêtes à investir de fortes sommes afin d’obtenir un système d’information performant et adapté à leur activité. C’est également un enjeu majeur pour la compétitivité de l’entreprise.

Un **système d'Information** (noté *SI*) représente l'ensemble des éléments participant à la gestion, au traitement, au transport et à la diffusion de l'information au sein de l'organisation.

La réalisation d'un site dynamique doit être impérativement précédée d'une méthodologie de conception permettant d'offrir une vision claire des différents éléments de l'application et de l'interaction entre ces derniers.

C’est à ce stade que nous devons clarifier la vue globale, en décrivant l’architecture générale que nous allons suivre dans la partie réalisation de notre projet. Nous détaillerons ensuite notre choix conceptuel à travers plusieurs types de diagrammes.

#### Etapes de développement logiciel

Le processus de développement logiciel contient un certain nombre d’étapes :

1. Définir les besoins et les exigences du client et des utilisateurs ;
2. Analyser le système ;
3. Concevoir le système ;
4. Programmer le logiciel ;
5. Tester le logiciel ;
6. Déployer ;
7. Maintenir le système.

Figure 34: Etapes de développement logiciel

Ces étapes ne sont pas forcément utilisées de façon linéaire. On parle souvent de cycles de vie, qui ont pour but d’organiser ces étapes de différentes manières en fonction d’un certain nombre de critères relatifs au projet de développement. Ci-dessous, quelques exemples de cycle de vie.

* La cascade ;
* Le modèle en V ;
* Le modèle en W ;
* La spirale ;
* Le RAD.

Dans la conception du site web G-SANTE, nous allons utiliser le modèle de **cycle de vie en V**. Les besoins et les exigences risquent d’évoluer, d’où le choix de ce modèle. Ce cycle ne met pas en évidence l’étape de **Maintenance**.

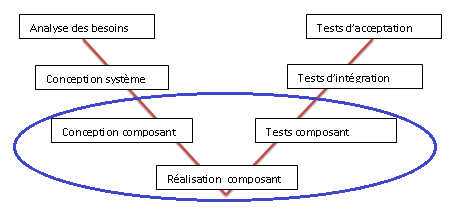


Figure 35: Cycle de vie en V

#### Cahier des charges

##### Informations générales

Le site aura un nom de domaine attribué qui sera le suivant : **www.g-sante.loc**

L’administration du site se fera sur un espace d’administration DRUPAL.

L’ensemble du site sera accessible par le protocole HTTPS (HyperText Transfert Protocol Secure). Ce système va crypter les échanges des données entre le navigateur web de l’internaute (Internet Explorer, Mozilla Firefox, Google Chrome, Safari…) et le serveur du site interne.

Le logiciel Google Analytics est un outil de statistiques qui vous permettra d'analyser plus en profondeur l'audience de votre site (sources des visiteurs, taux de rebond, temps passé, nombre de pages vues…).

##### Utilisateurs

Le site G-SANTE se compose de trois (3) types de profils :

* Un profil « *publique* », tous les utilisateurs ont accès à l’ensemble des informations disponibles sur le site ;
* Un profil « *administrateur* », permettant de gérer et de paramétrer tout le site à partir du backoffice Drupal, et permettant l’édition de tout type de contenu, la gestion des newsletters, la gestion des utilisateurs souhaitant un accès à l’espace professionnel ;
* Un profil « *adhérent* », pour l’accès à l’espace professionnel.

##### BackOffice

###### Gestion de la newsletter

Le module simplenews de Drupal sera utilisé. Grâce à ce module, les administrateurs du site pourront avoir une liste des différents inscrits à la newsletter, créer de nouveaux mails, et gérer leurs envois.

Grâce à ce module, les administrateurs du site pourront avoir une liste des différents inscrits à la newsletter, créer de nouveaux mails, et gérer leurs envois.

###### Gestion des articles

Les articles seront gérés directement par le gestionnaire d’articles de Drupal.

L’administrateur se connectera sur l’interface, et pourra au choix :

* voir la liste des articles déjà présents sur le site ;
* modifier un article ;
* ajouter de nouveaux articles, à publier sur le site du G-SANTE.

Lors de l’ajout d’un article, l’administrateur devra renseigner dans quel menu (ou sous menu) l’article devra être inséré.

###### Gestion des pièces jointes

Chaque article pourra avoir des pièces jointes des types suivants:

* Image ;
* document Word, Excel, Powerpoint ;
* fichier PDF.

Pour chacun de ces types d’articles, l’administrateur doit pouvoir en ajouter un nombre illimité sur chaque article, qui devront s’afficher esthétiquement sur la page de l’article.

###### Gestion des actualités

Une actualité sera un article de type “actualité”. Une fois l’article écrit, il devra être mis en “actualité” afin qu’il soit ajouté automatiquement à la page d’accueil.

###### Gestion des formulaires

Pour chaque formulaire (demande de contact, demande de devis, demande d’informations) etc. le module “Webform” de Drupal sera utilisé.

Une fois que l’utilisateur aura répondu au formulaire, un email sera envoyé aux administrateurs du site et les réponses seront historiées au sein du Backoffice de Drupal pour garder un historique complet.

Un seul formulaire peut être créé pour l’ensemble du site avec la possibilité de choisir dans un menu déroulant l’objet de la demande avec le titre « Quel est l’objet de votre message ». Dans la liste déroulante les choix seront les suivants : participer à un atelier, proposer un nouveau projet, demander un devis, demander des informations, retour d’expérience, autre(s). Cette liste déroulante doit pouvoir être administrable à partir du backoffice.

##### Informations techniques

Le CMS à utiliser sera Drupal 7 ou 8, dans sa dernière version stable.

Le site ne devra pas être prévu spécifiquement pour tablettes (en responsive design) mais devra fonctionner parfaitement sur iPad sans scroll horizontal (résolution minimum : 1024 x 768 px) ainsi sur que sur téléphone en mode portait et paysage.

Le site devra fonctionner sans anomalies sur les navigateurs suivants :

* Internet Explorer 10 et supérieur ;
* Chrome (versions > ou égales à 2014) ;
* Firefox (versions > ou égales à 2014).

Un système de gestion de contenu (CMS) sera utilisé pour limiter les temps de développement et permettra aux administrateurs d’avoir une interface simple, ergonomique et simple d’utilisation. Aucune modification profonde du CMS ne sera faite, permettant de profiter pleinement des mises à jour fournies par l’éditeur du CMS. Toutes les modifications seront apportées sous forme de modules ou de Template.

#### Les diagrammes UML utilisés

La définition d’un logiciel peut être grossièrement divisée en deux parties :

* L’étape de l’analyse (analyse des besoins, du domaine,…) ;
* La conception de la solution.

Dans chacune de ces deux parties, nous utiliserons un certain nombre de diagrammes UML :

* **Diagramme de cas d’utilisation** : pour définir les besoins et pour indiquer de quelles façons les acteurs utiliseront le système. Il est aussi utilisé dans l’analyse du domaine ;
* **Diagramme de classe** : utilisé dans l’analyse du domaine et pour préciser les informations nécessaires pour les actions d’un cas d’utilisation ;
* **Diagramme de séquences** : utile dans l’analyse de la solution. Il est aussi utilisé afin de détailler le déroulement d’un cas d’utilisation tout en indiquant les informations à utiliser ;
* **Diagramme d’activité** : afin de décrire le déroulement des cas d’utilisation.

#### Diagramme de cas d’utilisation

##### Les acteurs

Après une analyse approfondie du cahier des charges, on a pu en déduire les acteurs et les cas d’utilisation suivants (voir tableau) :

Tableau 7: Liste des acteurs et leur cas d'utilisation

|  |  |
| --- | --- |
| ACTEURS | CAS D’UTILISATION |
| Lecteur | *Visiter et consulter le site* |
| *Lire les actualités* |
| *Partager un lien* |
| *Participer à un projet* |
| *Bénéficier d’un projet* |
| Adhérant | *Accéder à l’espace professionnel* |
| *S’authentifier (cas secondaire)* |
| *Faire les mêmes opérations que le lecteur* |
| Administrateur | *Gérer les contenus* |
| *Gérer les utilisateurs* |
| *Gérer les newsletters* |
| *Effectuer les mêmes opérations que l’adhérant* |

##### Le diagramme de cas d’utilisation de notre SI

Figure 36: Diagramme de cas d'utilisation de notre SI

#### Diagrammes de séquence

##### Diagramme de séquence « visiter et consulter le site »