



Nantes1900

Livable unique – 26 Mai 2011

Polytech'Nantes – département Informatique

Binôme étudiant :

- Olivier CATRY
- Vincent MEUNIER

Tuteur enseignant :

- Guillaume Raschia

Coordinateur :

- Philippe Leray

Organisme commanditaire : Château des ducs de Bretagne

Tuteurs industriels :

- Florent Laroche
- Jean-Louis Kerouanton

Table des matières

Devoir de confidentialité.....	4
Charte contre la fraude et le plagiat.....	5
Introduction.....	6
Contexte du projet.....	7
1. Le port de Nantes.....	7
2. La maquette.....	12
3. Le projet Nantes 1900.....	15
Cahier des charges.....	18
I. Fondement du projet.....	18
1. But du projet.....	18
2. Personnes et organismes impliqués dans le projet.....	18
3. Utilisateurs du produit.....	19
II. Contraintes sur le projet.....	19
III. Exigences fonctionnelles.....	22
1. Portée du travail.....	22
2. Portée du produit.....	22
3. Exigences fonctionnelles et exigences sur les données.....	31
IV. Exigences non fonctionnelles.....	33
1. Ergonomie, utilisation et facteurs humains.....	33
2. Fonctionnement.....	35
3. Maintenance et portabilité.....	35
4. Standards.....	35
Etude bibliographique.....	36
Introduction.....	36
I. Contexte du projet.....	37
1. Le port de Nantes.....	37
2. La maquette.....	42
3. Le projet Nantes 1900.....	45
II. Projets similaires.....	47
1. Reproduction de Nantes.....	47
2. Exemple de site dynamique : reduction-image.....	49
III. Outils techniques.....	51
1. Internet et communication client-serveur.....	51
2. Bases de données.....	53
IV. Conclusion.....	56
V. Glossaire.....	57
Procédure de Recette.....	58
Estimation des charges et planification.....	65
Estimation des charges.....	65
Planning effectif (phase bibliographique + conception).....	67
Planning effectif phase 3.....	68
Modélisation.....	69
I - Conception.....	69
1 - L'application.....	69
2 - Les données.....	71
3 - L'interface.....	73

II – Réalisation (méthode agile).....	74
1 – Le chercheur.....	74
2 – L'administrateur.....	74
Réalisation.....	75
Partie 1 – Outils.....	75
I – La base de données.....	75
II – Le programme.....	75
Partie 2 – Déroulement du logiciel.....	75
Partie 3 – Mise en œuvre.....	85
I – La base de données.....	85
II – Le programme en PHP.....	87
Partie 4 – Tests.....	91
Manuel d'utilisation.....	93
S'authentifier.....	93
Ajouter des données.....	95
Retirer des données.....	98
Mettre en relation des données.....	99
Modérer des propositions.....	100
Dossier de déploiement.....	101
PV de Recette.....	102
Manuel d'installation.....	113
Dimension organisationnelle, commerciale ou environnementale.....	114
Fiche :.....	114
Bilan du projet.....	121
Conclusion.....	122
Références.....	123
Partie bibliographie.....	123

Devoir de confidentialité

Le devoir de discrétion est une règle absolue. A remplir et signer dès le début du projet

Les élèves-ingénieurs :

M. Vincent Meunier, né le 28 Avril 1988 à Rennes

et

M. Olivier CATRY, né le 28 Octobre 1988 à Agen

s'engagent à ne pas publier ni divulguer de quelque façon que ce soit les informations scientifiques, techniques ou commerciales recueillies ou obtenues par eux au cours de la réalisation du projet décrit dans ce présent rapport, sans l'accord écrit préalable de l'organisme commanditaire.

Cet engagement vaut pour la durée du projet et les 12 mois qui suivent son expiration.

Les élèves-ingénieurs s'engagent à ne conserver, emporter ou prendre copie d'aucun document ou logiciel, de quelque nature que ce soit, appartenant à l'organisme commanditaire, sauf accord de ce dernier.

Cette confidentialité peut s'appliquer aux soutenances de projet des phases 1 et 3 qui dans ce cas, et sur demande écrite de l'organisme commanditaire, se dérouleront à huis clos.

A Nantes, le 22 Novembre 2010

"Lu et approuvé"

Signature

Lu et approuvé
Vincent Meunier

"Lu et approuvé"

Signature

Lu et approuvé
Olivier Catry

Charte contre la fraude et le plagiat

Rappel de la charte signée lors de l'inscription, et que vous vous êtes engagé à appliquer :

Définitions :

La fraude : moyen quelconque pour ne pas être honnête lors d'un devoir surveillé, d'un rendu de projet ou de TP, seul ou en groupe. Pour chaque évaluation réalisée des élèves ingénieurs, la note personnelle ou de groupe doit refléter au mieux l'état des connaissances ou compétences acquises.

Le plagiat : c'est l'utilisation non mentionnée de contenu intellectuel déjà réalisé par une tierce personne ou groupe de personnes en vue de réutilisabilité illicite pour ne pas avoir soi même à développer ce contenu. Le plagiat n'est pas plus tolérable ni acceptable que la fraude : en plus de faire croire que l'on est l'auteur de ce que l'on n'a pas fait, on dépouille le véritable auteur de ses droits intellectuels ce qui devient un délit dans la société du savoir. La bonne attitude consiste à beaucoup se documenter mais toujours citer ses sources (textes, code, rendu de tp, etc.).

La fraude et le plagiat sont passibles de sanctions qui peuvent aller jusqu'à l'expulsion de l'Université.

La bonne attitude consiste à beaucoup se documenter mais à toujours citer ses sources.

- Tout travail d'un(e) étudiant(e) doit être personnel.
- Lorsque l'on utilise un passage d'un livre, d'une revue ou d'une page Web (traduit ou non), il doit être mis entre guillemets avec mention de la source et de la date.
- Lorsque l'on utilise des images, des graphiques, des données, etc. provenant de sources externes, celles-ci doivent être mentionnées.
- Lorsqu'un travail produit pour un cours est réutilisé pour un autre cours, il convient d'en demander l'autorisation.

Introduction

Le projet transversal en informatique à Polytech' Nantes est un projet réalisé pour une entreprise par des étudiants en quatrième. Notre projet a été pour le Château des Ducs et dirigé par M. Kerouanton et M. Laroche, et encadré coté école par M. Raschia.

Ce livrable unique couvre le développement du projet d'interface de base de donnée pour le projet Nantes 1900 qui a pour but de valoriser la maquette du port de Nantes exposée au Château des Ducs de Bretagne à Nantes. Une description de ce projet est faite à la suite de cette introduction.

Il présente les trois phases du développement du projet, à savoir la phase bibliographique qui couvre la découverte du contexte du projet et la recherche des outils pour mettre en place une solution à la problématique posée, la phase de conception qui présente les diagrammes et plans du projet ainsi que son articulation prévue, et la phase de réalisation qui présente le déroulement de la programmation du site et le travail effectué sur la base de données. Le point est fait tout au long de ces trois phases sur le planning prévu et sur les méthodes de gestion du projet.

Par la suite, un rapport est fait sur les différents tests pratiqués et leur validation, et un manuel d'utilisation est présenté. Le manuel de déploiement permet l'installation sur un serveur. Tout ceci est groupé dans la dernière partie de ce livrable qui se penche ainsi sur les aboutissants du projet.

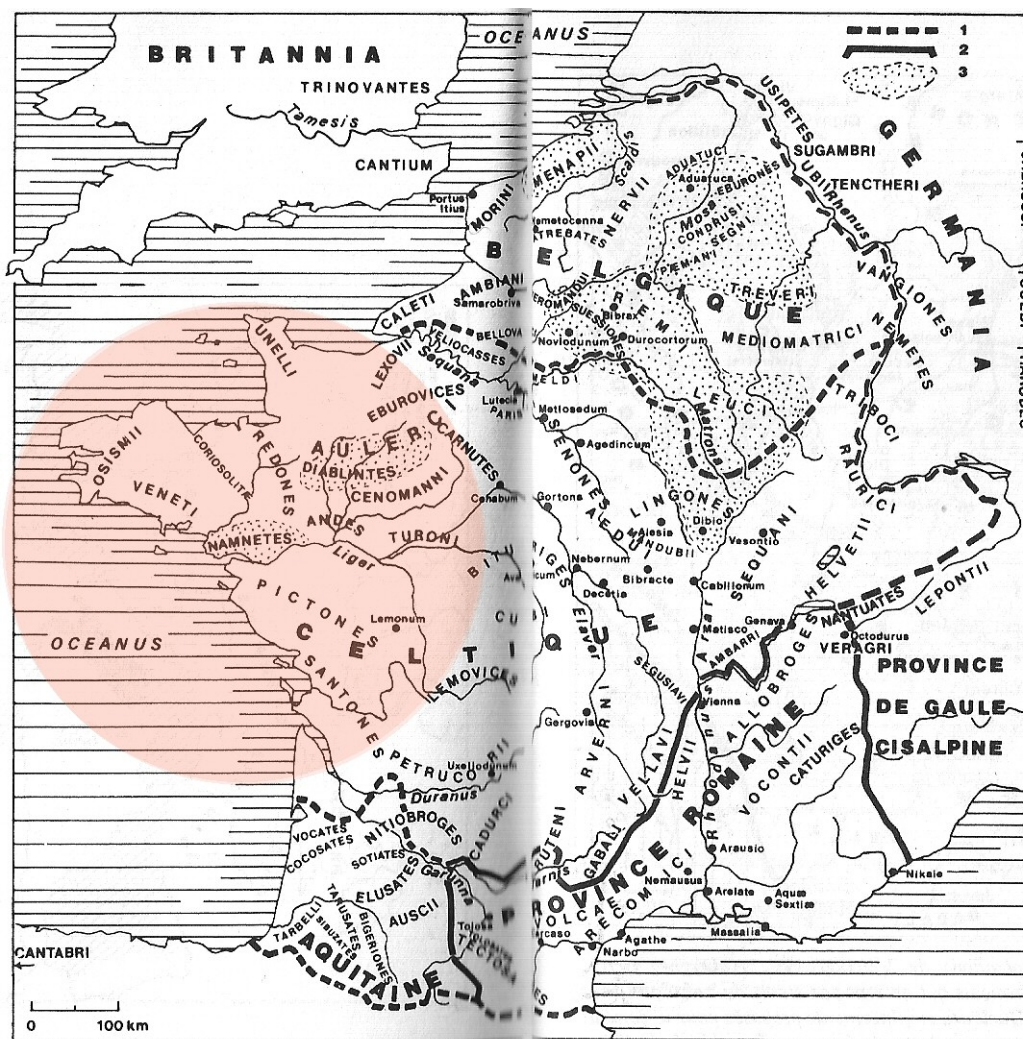
Enfin, une présentation de la partie organisationnelle est faite afin de mettre en avant l'originalité du projet Nantes 1900 et la façon dont notre travail s'est adapté à ceci. Ceci est suivi par un bilan de l'expérience acquise grâce à ce projet transversal de six mois.

Contexte du projet

1. Le port de Nantes

a. Portus Namnetum

Entre 600 et 200 av. J.-C., durant les invasions celtiques, de nombreux peuples se sont installés en Gaule. Au niveau de l'actuel département de Loire-Atlantique, le Nord de la Loire était occupé par les Namnètes. Ce peuple était un peuple de commerçants prudents et avisés et d'hardis navigateurs fluviaux comme de nombreux peuples gaulois. Le Sud de la Loire était occupé par les Pictons.



Situation des peuples de la Gaule en -60. Focus coloré sur la région Armoricaîne et la côte Atlantique

En -57, les Namnètes ont combattu l'invasion romaine et ont été maté par l'armée de César alliée aux Pictons. Cette défaite a donné naissance au Port de Ratiatum (Port de Rezé) tenu par les Pictons et qui est l'actuel quartier Saint-Lupien de la commune de Rezé. En face, au Nord de la Loire, se trouvait Portus Namnetum, bourg Gallo-Romain qui deviendra la Ville de Nantes. L'actuel

Château des Ducs était un Castrum, camp Romain auquel le port était exposé.

Durant la phase Armoricaine de la Guerre des Gaules en -56, les Vénètes ont invité les peuples Gaulois à se soulever contre l'occupation romaine et ont exigé de Crassus qu'il rende les otages. Jules César s'est alors allié avec les Osismes, les Lexovii, les Ambiliates, les Morins, les Diablintes, les Ménapes et notamment les Namnètes, afin de mener la guerre contre les Vénètes. [1]

À cette époque, des ponts en bois permettaient de circuler entre les îles de Saulzaie, de la Madeleine et de Biesse. Portus Namnetum assurait le lien entre l'Océan Atlantique et la mer Méditerranée via la Loire et le Rhône.

Le bourg prospérera par la négoce de métaux, vins italiens, huiles d'Espagne, sels et poissons. [2], [3]

Au Ve siècle, la Ville se christianise et devient un évêché. Les incursions saxonnes (desquels on a retrouvé les vestiges de pirogues monoxyles et de drakkars dans la Loire) au IVe siècle et des Vikings en IXe siècle ont forcé la ville à se fortifier de plus en plus. Une cathédrale est bâtie près de la porte Saint-Pierre. En 470, les Celtes de Grande-Bretagne migrent en Armorique, c'est la naissance de la Bretagne.

b. Port Commercial

Au XIVème siècle, la guerre de succession a obligé le déplacement des activités portuaires de Nantes vers l'Erdre qui était plus sécurisé que la Loire. Après la fin de l'indépendance des Bretons, Nantes est devenu port Français. Son principal commerce est celui de vin et du sel

Au XVème siècle, François II ordonne l'édification d'un château à l'emplacement du Castrum de Nantes. Selon Joseph Stany Gauthier, conservateur du Château des Ducs dans les années 60, il existe des traces plus anciennes de l'édification d'un fort, notamment un procès à l'empiètement sur les terrains de l'évêque aux environs de 1212. Stratégiquement, le château était bien placé pour défendre le port.

Au XVIème siècle, de nombreux produits textiles ont été ajoutés aux transactions au port de Nantes durant la période de « Boom du textile ». Il est à noter qu'en 1598, l'Edit de Nantes est signé par Henris IV, offrant la liberté de culte aux protestants. [4]

Au XVIIème siècle, des produits de luxe et du sucre venant du Brésil sont venus s'ajouter au commerce du port. Celui-ci est devenu également en partie un port poissonnier avec l'arrivée des Espagnols installés à la Fosse. De nombreux mouvements xénophobes ont été menés contre les portugais et les hollandais. Ces derniers sont accusés notamment d'être à l'origine des problèmes de tonnage en jetant du lest dans l'eau. Mais on suppose aujourd'hui que l'ensablement du fleuve avait pour origine la déforestation du Massif Central et du Morvan. De plus, le port de Nantes était alors reconnu à cette époque comme Port International. En conséquence, on a émis l'idée de le doter d'un avant-poste. On notera la visite de Louis XIII et Richelieu en 1614 et de Louis XIV en 1661 puis sa révocation de l'Edit de Nantes en 1685. [3]

Au XVIIIème siècle, le Code Noir émis par Louis XIV a permis au port de faire de nouvelles transactions : celles du tabac et des esclaves. Le sucre est resté le produit dominant avec de nouvelles importations depuis Saint-Domingue (en actuelle République Dominicaine) . Le Port a acquis la dénomination de port colonial et premier port négrier de France. Avec le commerce triangulaire, Nantes s'est accaparé de 40% des expéditions négrières françaises jusqu'à l'abolition de l'esclavage en France en 1848. Le port s'est agrandi : une fonderie de canons royale a été construite en 1777 et de nombreux ponts avec chaussée traversaient le lit de la Loire. Les premiers armateurs Nantais remontent à cette époque : les Mautondouin.

c. Port Industriel

Durant la première moitié du XIXe siècle, le port de Nantes a commencé son industrialisation :

La stérilisation d'Appert a été à l'origine de l'implantation des conserveries Pierre Joseph Colin en 1822. Joseph Collin était reconnu comme fondateur de la conserverie industrielle. Nantes est ainsi devenu capitale de la sardine en boîte. Le pâtissier Jean-Romain Lefèvre est arrivé à Nantes en 1846 et s'est marié à une demoiselle Utile. Leur fils Louis Lefèvre-Utile a fondé alors sur l'île de la Madeleine la biscuiterie LU. La fonderie de canons royale s'est transformée en 1828 en usine de machines à vapeur. Les usines de raffinage du sucre se sont transformées en métallurgies. Au milieu du XIXe siècle, la basse Loire lançait le quart des navires français. On a créé en 1838 le port de Saint-Nazaire, résultat du projet de construire un avant-poste au port.



Durant la seconde moitié du XIXe siècle, le port importait principalement du sucre, du riz et de la houille et exportait des métaux, des produits métallurgiques et des produits alimentaires. Il est devenu le 6ème port de France. En 1868, constatant que le trafic était plus important à Saint-Nazaire, on a décidé créer un canal latéral à la Loire qui sera finalisé en 1892. Le canal de la Martinière mesurait 15 km de long et a dû être abandonné en 1910, devenant un cimetière pour 3 mâts. On a vu à cette époque la construction du pont transbordeur, monument impressionnant de l'ère industrielle, qui permettait de traverser la Loire au moyen de plateformes et des câbles.

On notera la construction de Belem [5][6][7] en 1896 par les chantiers Dubigeon à Chantenay à l'emplacement de l'actuel ancien bâtiment de la Bourse. Le trois-mâts était chargé de l'importation de cacao depuis le Brésil, puis a appartenu à Guinness, en Irlande. Il faudra attendre 1986 pour son retour à la ville de Nantes. Un plan de destruction à la dynamite des îles est mis en place afin d'élargir la Loire et abaisser son fond. Il est exécuté de 1903 à 1914.



Le Belem. Tableau du Musée des Salorges, Nantes. Image numérisée par Laurent Claude Gloaguen (lcg.net)

Les problèmes de déblayage du lit de la Loire resteront dans les esprits jusqu'à de nos jours. Par exemple, en 1979, il a été nécessaire de déblayer 2000 mètres cubes de roche dans le lit de la Loire car les nouveaux navires de l'époque avaient besoin de 9 mètres sous la ligne de flottaison. Cela a permis l'arrivée de navires transporteurs d'agrumes venus du Maroc. [8]

2. La maquette

a. Son histoire

L'Exposition Universelle de 1900 a eu lieu à Paris, et c'est dans ce cadre que la Chambre de Commerce a commandé la réalisation d'une maquette du port de Nantes. L'objectif était de présenter l'aménagement de la Loire au niveau de la ville de Nantes et les industries et monuments qui résidaient dans ces environs. Elle a été modifiée jusqu'en 1913 et la ville de Nantes l'a acquise en 1927. Le musée de Nantes l'a exposée jusqu'en 1960. On a pu encore l'admirer de 1974 à 1994 dans le musée du Château des Ducs. En 1996, elle a été restaurée et c'est depuis la réouverture du musée du Château des Ducs en 2007 qu'elle a de nouveau été exposée au public. [9]



L'exposition du Champs de Mars au Trocadéro - 1900

Ce qui a motivé la réalisation de la maquette était la puissance et la célébrité de Nantes à cette époque. Des trains traversaient la ville par 30 passages à niveau, 350 usines étaient installées, et de nombreux chantiers étaient bâtis le long de la Loire. Nantes était connue dans toute la France, c'est d'ailleurs la seconde ville du pays la plus chantée. Elle était également célèbre pour ses biscuiteries (LU, B.N.). Enfin, le port était le 4ème plus grande France et le plus grand bassin d'emploi jusqu'à la Première Guerre Mondiale. [2]

b. Ses caractéristiques

La maquette, réalisée par le menuisier Paul Duchesne, couvre la superficie depuis l'ancienne Bourse de Nantes jusqu'aux environs de Chantenay, soit 3 km sur 500 m. Ses dimensions sont de 920 sur 185 cm. Elle a été réalisée à partir de 6 matériaux : du bois, des métaux, des fibres textiles, du papier, du carton et du verre. [10]

On peut partager la maquette en trois grandes modélisation 3D (cf II-2). [2]

parties sensiblement égales en superficie : Les Chantiers à l'Est, le Butte Sainte Anne au centre

et la zone industrielle à l'Ouest. Sur les cours La photo ci-contre montre l'étendue de la d'eau, de nombreux bateaux ont été modélisés, superficie couverte, et la finesse de la pour la plupart des 3 mats à voile et en fer. Le modélisation telle qu'on peut la voir au premier réseau ferroviaire et routier a été reconstitué de plan. On regarde ici la maquette depuis l'Est. façon détaillée et tous les bâtiments ressortent comme des volumes. Ce qui a été matière à la

c. Son exposition

La maquette est exposée en Salle 21 du Château des Ducs, et est placée dans un angle. Cette disposition et la vitres protectrice empêche le visiteur d'avoir d'autre point de vue que de l'Est et le Sud de la maquette.



Maquette du Port de Nantes

Le problème a alors été posé : le visiteur ne peut pas apprécier la finesse des détails sur toute la superficie de la maquette. Anne Guigun, une stagiaire d'Histoire, a mené une étude auprès des visiteurs du musée, qui a révélé que de nombreuses personnes avaient du mal à se repérer sur la maquette et désiraient un plus haut degré d'interactivité avec celle-ci.

C'est pourquoi le projet Nantes 1900 a été lancé, afin de palier à ces lacunes, en numérisant la maquette en 3D et en ayant pour objectif d'offrir au visiteur la possibilité d'obtenir de nombreuses informations sur tous les éléments de la maquette avec un système interactif de projecteurs et d'écrans.

3. Le projet Nantes 1900

a. Commanditaires

Le château des Ducs de Bretagne [11] est le commanditaire du projet Nantes 1900, qui est sous la responsabilité de Bertrand Guillet et Christophe Courtin. L'objectif est de valoriser la maquette d'un point de vue historique et muséographique.

Bertrand Guillet est conservateur en chef du patrimoine et directeur adjoint du château des Ducs de Bretagne. Christophe Courtin est chargé du multimédia et de l'audiovisuel au château. Ils ont confié la partie technique du projet Nantes 1900 au centre François Viète, à Jean-Louis Kérouanton et Florent Laroche. Jean-Louis Kérouanton est docteur en histoire et maître de conférence à l'Université de Nantes et Florent Laroche est maître de conférence à l'école Centrale de Nantes et chercheur au laboratoire IRCCyN ¹. Ils nous ont chargé de réaliser l'interface pour la base de données relative à la maquette, suite du projet transversal précédent confié à François Pérez et Lucie Thourault.

b. Centre François Viète.

Le centre François Viète [12] [13] [14] [15] se trouve sur le campus de la Fac de Science de Nantes. C'est un pôle de recherche dans l'épistémologie et l'histoire des sciences et des techniques.

Le centre réaliste neuf thèmes de recherche qui sont toutes axées autour de l'épistémologie. Il enseigne aussi aux étudiants en vue d'obtenir un master 2 d'Histoire des Sciences et des Techniques qui s'organisent autour de l'étude épistémologique et historique des sciences en général. Enfin, le centre soutient et encadre des thèses sur l'histoire des sciences et des techniques depuis 1996.

Les membres de l'équipe publient de nombreux ouvrages et articles et organisent des colloques et des séminaires. La revue « Cahier François Viète » publie également certains de leurs résultats de recherche.

c. Organisation du projet

Le projet est ouvert et évolutif et laisse libre cours à l'imagination et la créativité. La demande du client est traduite et les objectifs évoluent tout au long du projet de façon à offrir un résultat à la pointe de la technologie selon les attentes du Château des Ducs. Le projet est pluridisciplinaire, c'est à dire qu'il implique les compétences d'acteurs de l'école Polytechnique, Centrale et de l'Université de Nantes. C'est un lien étroit qui est fait entre histoire, culture et technique.

Pour le projet, une plateforme collaborative de travail (team work place of Lotus) est utilisée. Elle permet de centraliser sur un serveur l'ensemble des travaux et rapports, et de mettre en communication les différents acteurs de l'équipe. Ainsi, la traçabilité est assurée. [16]

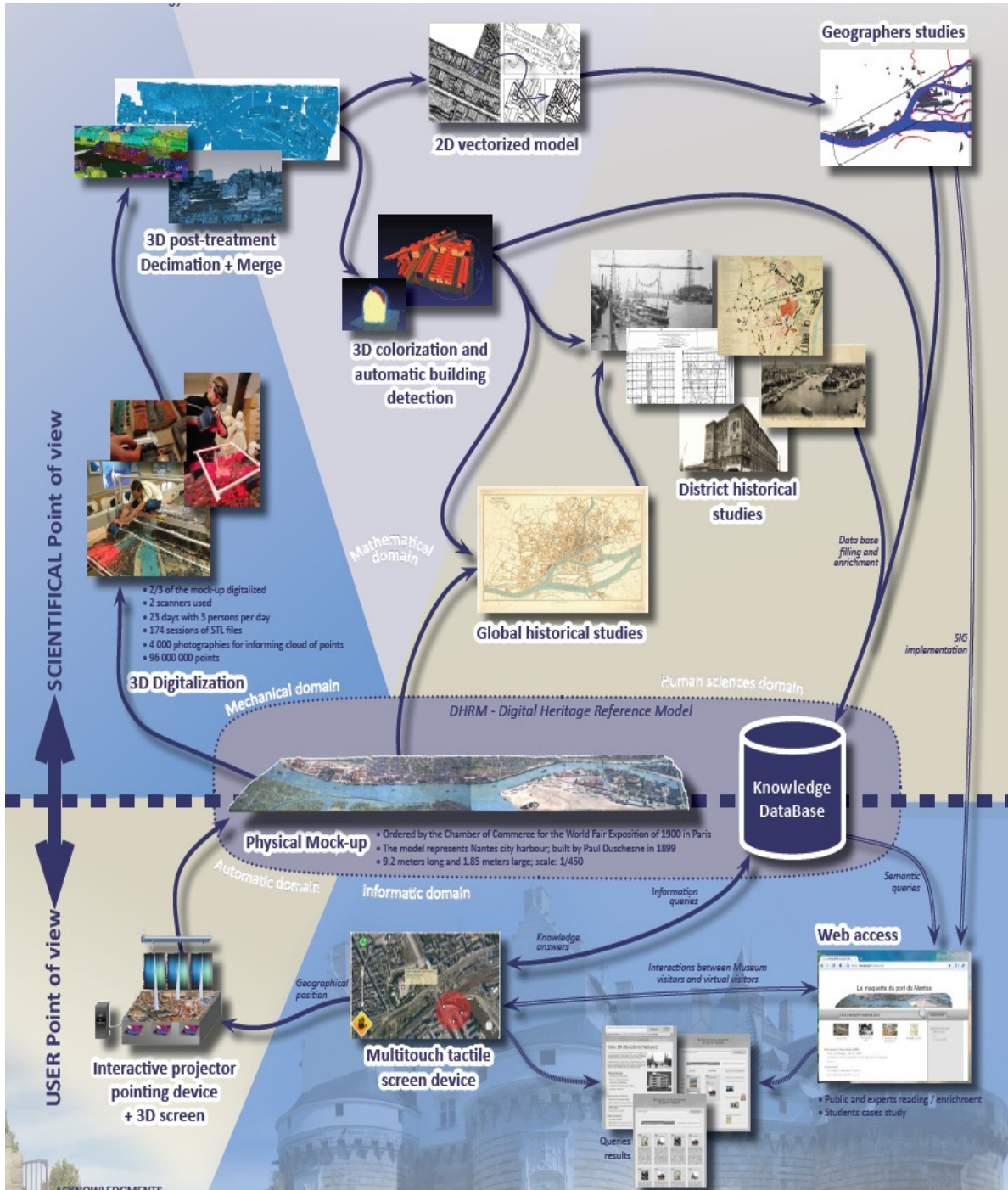


Schéma récapitulatif de l'ensemble du projet et des différentes tâches et interactions entre les équipes.

Selon ce schéma récapitulatif, nous sommes chargés de la partie « Web access ». On peut voir le volume et la diversité des travaux que met en jeu ce projet.

Cahier des charges

I. Fondement du projet

1. But du projet

a. Contexte du projet

Une maquette du port de Nantes a été construite pour la Chambre de Commerce en 1900. Elle est exposée aujourd'hui au Château des Ducs de Bretagne à Nantes en salle 21. En 2008, Le musée d'histoire de Nantes a décidé de valoriser d'un point de vue scientifique et muséographique cette maquette par l'intermédiaire du projet Nantes 1900. Plusieurs équipes ont déjà travaillé sur ce projet dont la fin est prévue pour 2012.

Ainsi, la maquette a été numérisée en 3D en Avril 2009. Pour informatiser toutes les informations fournies par cette maquette, la structure d'une base de donnée a été construite par Tayib Faraj et Omar Minet, Elise Lecoq et améliorée par François Perez et Lucie Thourault. Ces deux derniers ont conçu un moteur de recherche sémantique permettant de réaliser des requêtes à la base de donnée.

L'objectif de cette base de donnée est de fournir des informations sur les éléments visibles dans la maquette et les mettre en relation avec un ensemble de connaissances supplémentaires.

b. Objectif

Le besoin est d'avoir une interface permettant à des personnes de remplir, modifier et récupérer des informations de la base de donnée, et à des administrateurs de contrôler les modifications réalisées.

2. Personnes et organismes impliqués dans le projet

a. Maîtres d'ouvrage

- Jean-Louis Kerouanton : Docteur en histoire et maître de conférence
- Florent Laroche : Docteur en génie mécanique et maître de conférence à l'école centrale de Nantes

b. Acheteurs

L'acheteur est le Château des Ducs de Bretagne par l'intermédiaire de la Société d'Économie Mixte Nantes Culture et Patrimoine.

c. Autres parties prenantes

Nous sommes pour cette période la seule équipe sur le projet.

Pour une liste des différentes parties prenantes dans l'histoire du projet, se référer au rapport de bibliographie, section contexte.

3. Utilisateurs du produit

Le visiteur du château peut consulter les informations depuis une dalle tactile, devant la maquette. Les contraintes apportées par un public de musée (groupes, enfants, personnes âgées, personnes handicapées, touristes étrangers etc...) seront à prendre en compte. On peut y assimiler l'internaute qui a accès à la même interface mais qui ne voit pas la maquette.

Le chercheur a un accès complet à la base de données et peut y réaliser des recherches approfondies.

L'informateur utilise les fonctionnalités d'ajout et modification de la base de données. Il dispose d'informations relatives à la maquette, et son objectif est l'apport de connaissance.

Le modérateur vérifie et contrôle les informations apportées ou modifiées par l'informateur.

L'administrateur contrôle les droits d'accès, les informations contenues dans la base de données, et sa structure.

II. Contraintes sur le projet

a. Contraintes sur la conception de la solution

vocabulaire client/serveur (une machine « client » se connecte à une machine « serveur »)

- L'interface doit être sous forme de page web.

Technologie : HTML 5

Navigateurs compatibles : firefox (2 minimum), internet explorer (6 minimum)

Systèmes d'exploitation compatibles : Windows XP

- Les pages web doivent être construites par un moteur coté serveur.

Technologie : PHP 5

- Les pages web doivent être dynamiques, coté client.

Technologie : javascript

- Utilisation de PostGreSQL et de SolR pour former les requêtes

La communication avec la base de données doit suivre un protocole

Technologie : PDO (PDO_PGSQL)

- L'aspect graphique de l'interface doit être similaire à celle du site de l'université de Nantes ou du château des Ducs. En attente d'une décision, il doit être modifiable indépendamment de son contenu.

Technologie : CSS

L'interface web doit être réalisée et fonctionnelle sur la machine qui nous est fournie.

Un procédé d'augmentation sémantique doit être implémenté. Il sera détaillé après une entrevue avec François Perez et Lucie Thourault.

Des vidéos, sons et images contenues dans la base de données doivent être affichés avec les données récupérées par les visiteurs. Leur intégration sera détaillé après une entrevue avec M. Courtin, directeur du Château des Ducs.

b. Environnement de fonctionnement du système actuel

La maquette est l'élément central du projet. Elle dispose d'une dalle tactile qui permet de récupérer des informations sur la base de données contenant toutes les connaissances associées à la maquette. La base de donnée est stockée sur une machine serveur au Château des Ducs. L'interface web pour cette base de donnée sera mise en ligne sur le même serveur.

c. Applications partenaires

L'interface web est constituée de pages HTML. Ceci implique une coopération avec le navigateur du client, car les pages doivent être affichées et conserver leur intégrité quel que soit le navigateur internet.

Le moteur qui construit les pages de l'interface doit être installé sur la machine serveur du musée. Il faut donc y installer un serveur PHP qui soit compatible avec le PHP5.

La base de donnée est déjà installée et fonctionnelle, donc il n'y a pas de mention d'applications partenaires à ce sujet.

L'interface web doit être affichée dans son intégrité à l'écran de l'utilisateur, qui n'est pas forcément un écran de taille suffisante pour afficher toute la page. Or, certains éléments graphiques seront en haute définition, et ne pourront être affichés intégralement à l'écran.

L'interface sera diffusée sur internet. Il faut donc que les pages ne pèsent pas plus de 1 Mo (car elles peuvent contenir des vidéos et de son dont on conserve les miniatures pour l'affichage et la haute définition sera disponible sur demande via un clic sur les miniatures)

d. Composants commerciaux

Techniquement, le code source des pages, incluant le code javascript, constituant l'interface, sera accessible à tous les internautes.

Le code source du moteur PHP ne sera pas accessible et sera la propriété du Musée d'Histoire de Nantes.

e. Lieu de fonctionnement

La base de donnée et le moteur de génération de l'interface tourneront sur la machine serveur du musée.

Les pages générées constituant l'interface web seront transmises sur la machine de l'utilisateur, le code javascript sera exécuté dessus, et l'affichage de la page sera également prise en charge par celle-ci.

f. Temps imparti

Le projet dure l'année scolaire, prend fin mi Mai 2011 et se déroule en trois phases : Bibliographie, conception et implantation + tests de recette.

g. Budget

Les besoins techniques ne nécessitent pas de dépenses.

Nantes Culture et Patrimoine finance le projet et peut décider d'une aide si besoin est.

h. Glossaire

Client – Serveur :

L'architecture client/serveur désigne un mode de communication entre plusieurs ordinateurs d'un réseau qui distingue un ou plusieurs postes clients du serveur : chaque logiciel client peut envoyer des requêtes à un serveur. Un serveur peut être spécialisé en serveur d'applications, de fichiers, de terminaux, ou encore de messagerie électronique.

HTML :

Ou Hyper Text Markup Language. C'est le langage de paramétrage des pages Internet. Un site web est constitué de documents HTML, que l'on appelle des pages.

Interface :

Une interface définit la frontière de communication entre deux entités, comme des éléments de logiciel, des composants de matériel informatique, ou un utilisateur. Elle se réfère généralement à une image abstraite qu'une entité fournit d'elle-même à l'extérieur.

Protocole :

Dans les réseaux informatiques et les télécommunications, un protocole de communication est une spécification de plusieurs règles pour un type de communication particulier. Initialement, on nommait protocole ce qui est utilisé pour communiquer sur une même couche d'abstraction entre deux machines différentes.

Serveur :

Ordinateur central d'un réseau informatique où sont stockées les données et applications accessibles par les autres postes connectés.

III. Exigences fonctionnelles

1. Portée du travail

a. La situation actuelle

Le projet a commencé en 2008. La dalle tactile pour la maquette a été réalisée ainsi que la gestion de requête via moteur sémantique. L'état actuel concernant la partie base de donnée est le manque d'une interface qui permette un contrôle de plus haut niveau sur la base de données. C'est à dire qu'il faut une application qui gère des utilisateurs, qui leur permette de réaliser certaines taches sur la base de donnée et qui soit graphique. Le principe est expliqué ci-dessous.

b. Contexte du travail

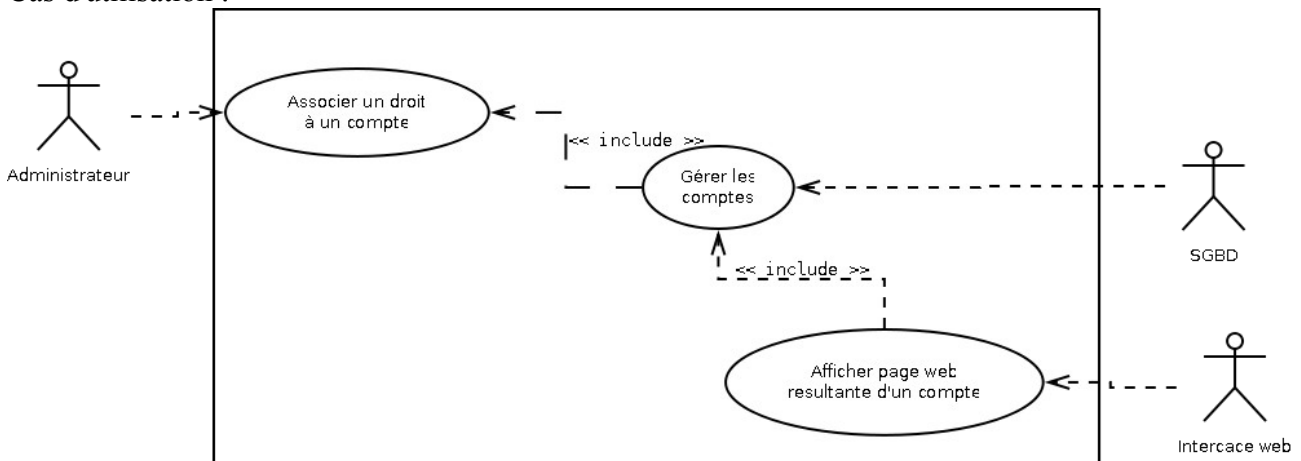
Nous devons nous entretenir avec les différents acteurs du projet, à savoir le directeur du Château des Ducs qui nous donnera des directives concernant l'intégration des éléments graphiques, ainsi que François Perez et Lucie Thourault qui nous informeront sur l'aspect technique de la base de donnée, que nous pourrons aussi étudier sur la machine qui nous sera fournie.

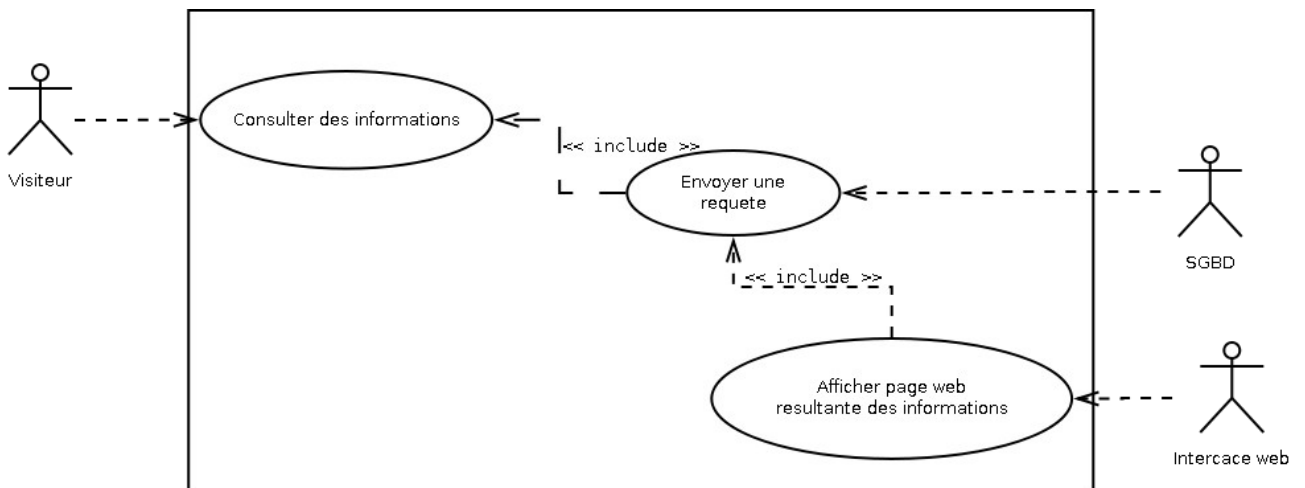
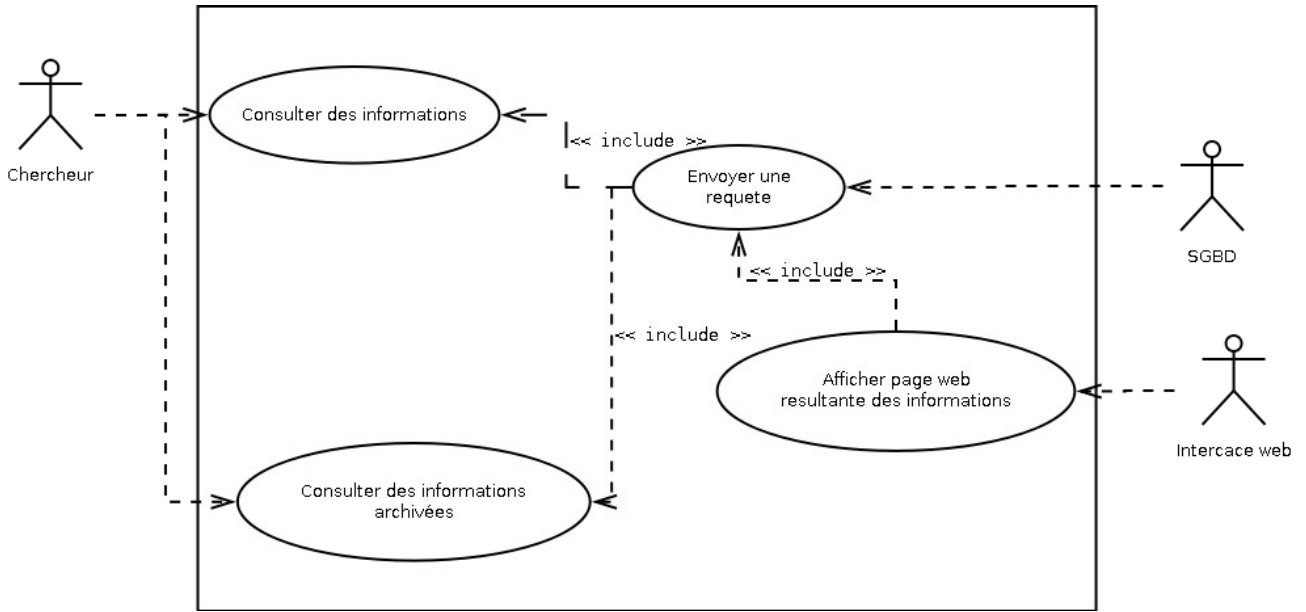
Nous devons aussi nous familiariser avec la partie historique du projet, c'est-à-dire étudier la maquette et les éléments relatifs à celle-ci exposés au Château des Ducs, et réaliser des recherches sur cette époque et sur les éléments constituant les informations recueillies par la base de donnée. Ceci nous permettra d'avoir des connaissances plus affinées sur le contexte de ce projet.

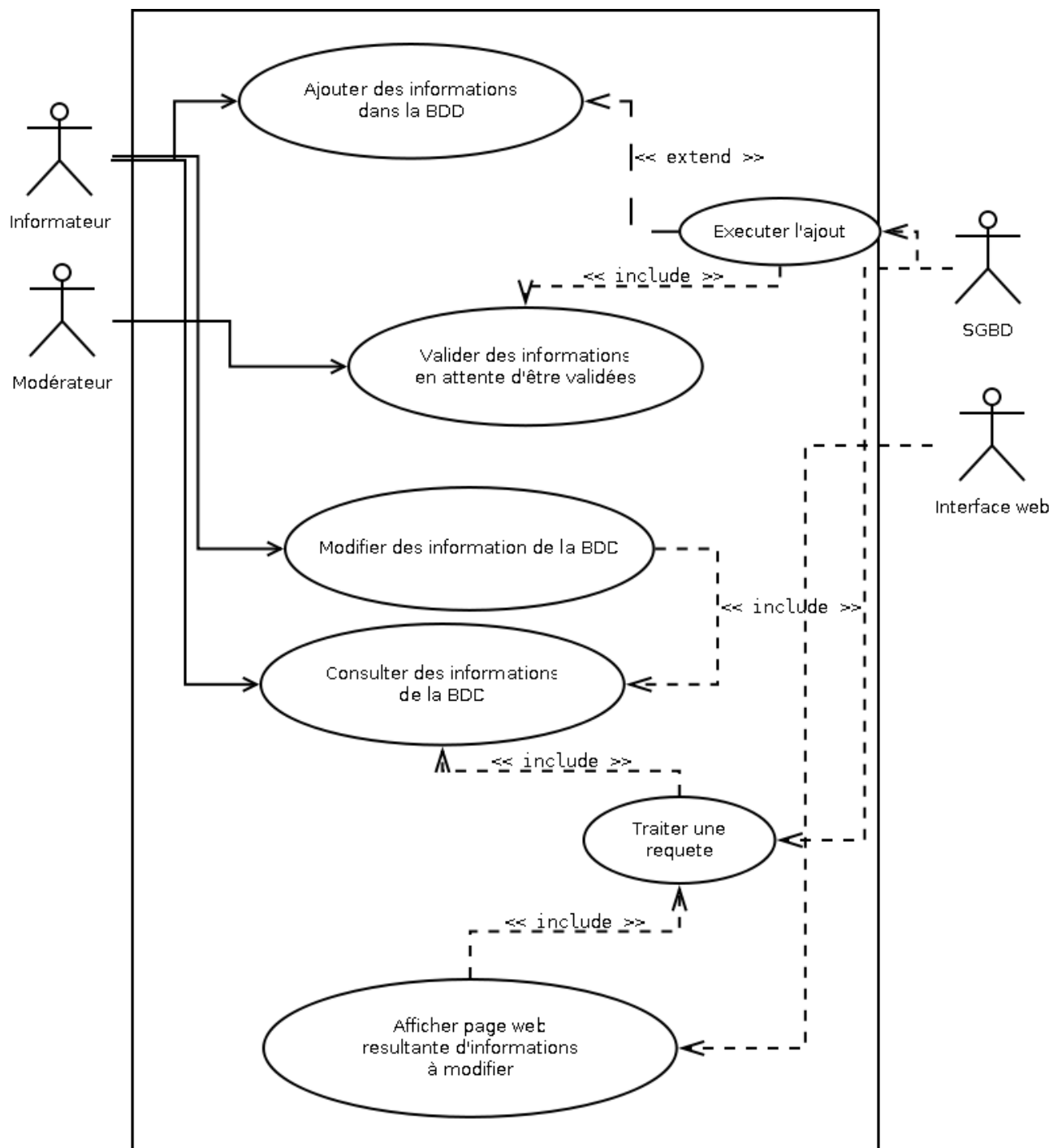
2. Portée du produit

a. Limite du produit

Cas d'utilisation :







Détail :Visiteur :Consulter des informations : Le visiteur utilise le moteur de recherche sémantique : il inscrit dans le cadre le terme de sa recherche, et lorsqu'il valide sa recherche, il reçoit une page avec les résultats de sa recherche.

Tout ce que peut faire le visiteur, c'est de demander des pages d'information, et récupérer des pages d'information.

Lorsqu'une recherche est exécutée, elle est convertie sous forme de requête, qui est envoyée au SGBD. Le SGBD traite la requête et retourne les données correspondantes au moteur PHP.

Le moteur PHP (interface web) construit une page en fonction de ce contenu et l'envoie au visiteur.

Chercheur :

Le principe est le même que pour le visiteur, à la différence qu'il peut en plus accéder aux informations de type archive.

Informateur :

Ajouter des informations :

Par cette action, l'informateur peut ajouter des informations sur les données. Il sélectionne l'attribut à renseigner (comme les données sont orientées objet) et il écrit l'information qu'il souhaite ajouter. En acceptant son ajout, l'information est stockée dans les ajouts en attente de validation.

Modifier des informations :

Par cette action, l'informateur modifie des informations déjà existantes. Il s'agit bien ici de supprimer une partie de l'information contenue pour la remplacer par une autre information (ou aucune). En acceptant son ajout, l'information est stockée dans les modifications en attente de validation.

Modérateur

Valider des ajouts :

Un écran indique au modérateur la liste des ajouts. Il peut lire l'ajout et choisir de le valider ou non. S'il le valide, alors l'information à ajouter est ajoutée à la base de données. Sinon, elle est supprimée des informations d'ajout en attente de validation.

Si l'ajout est validée, alors la donnée prend pour date de création de donnée la date de validation.

Valider des modifications :

Un écran indique au modérateur la liste des modifications. Deux champs apparaissent : l'un est constitué de l'information originale, et l'autre de l'information modifiée. En comparant les deux champs, il peut déterminer si oui ou non l'information en attente d'être modifiée est validée. Si elle l'est, alors la donnée de la base de données associée à cette information est modifiée. Sinon, l'information est supprimée des modifications en attente.

Si la modification est validée, alors à l'ancienne donnée on renseigne le champ de date d'expiration de donnée selon la date de validation de son remplacement, et celle-ci est placée en archive. La nouvelle donnée prend pour date de création de donnée la date à laquelle elle a été validée.

Administrateur :

Associer un droit à un compte :

L'administrateur a un écran lui listant la liste des comptes. Pour chaque compte est associé un droit. La liste des droits elle la liste donnée en b : visiteur, chercheur, informateur, modérateur, administrateur, super utilisateur. Chaque compte donne la possibilité à son utilisateur de réaliser des actions bien précises. Il n'y a pas de notion de promotion ou de dégradation, donc l'administrateur doit déterminer précisément le rôle du compte pour lui donner son droit. Il est à noter que

l'administrateur ne peut pas modifier son propre droit : seul un autre administrateur peut changer son droit.

Créer un compte :

Par le même écran, il peut créer un nouveau compte, lui associer identifiant et mot de passe et, comme précédemment, lui associer un droit.

Super utilisateur :

Le super utilisateur est le compte donné à celui qui gère le programme. C'est un compte technique qui peut réaliser la totalité des actions citées précédemment. Il ne sert que pour la maintenance du système et il sera judicieux de n'en conserver aucun lorsque le système sera fonctionnel, par raison de sécurité. L'administrateur pourra toujours créer un compte super utilisateur.

Instances des cas d'utilisation :

Informateur :

Ajout d'information :

L'informateur réaliser une recherche : « conserverie collin »

Il récupère les informations liées à la conserverie collin dont en voici un fragment :

Recherche : Conserverie Collin

Nom : Conserverie Collin
Description : N/A

Il souhaite ajouter une information : il veut ajouter des données au champ « description »

Il clic sur « description » et accède à cette page :

Conserverie Collin
Description

Soumettre

Conserverie Collin

Description

Créée par Joseph Collin

Soumettre

Le champ de texte lui permet d'écrire l'information qu'il souhaite ajouter. Une fois le texte écrit, il clique sur valider pour soumettre cet ajout : Modifier de l'information :

Admettons maintenant un autre informateur qui réalise la même recherche, et que l'ajout précédent a été enregistré. Il arrive sur cette page :

Recherche : Conserverie Collin

Nom : Conserverie Collin

Description : Créée par Joseph Collin

Il décide de modifier l'information relative à la description. Il clique sur description. Le système constate que le champ description est déjà renseigné, il entre donc en mode « modification ». L'informateur arrive sur cette page :

Conserverie Collin

Description

Créée par Joseph Collin

Soumettre

L'information déjà inscrite est préalablement rentrée dans son champ d'édition. Il peut alors modifier l'information. Il décide de le faire ainsi :

Conserverie Collin

Description

Créée par Maurice Collin

Soumettre

Puis il clic sur Soumettre pour soumettre sa modification à la modération.

Note : admettons maintenant que la demande d'ajout ne soit pas prise encore en compte. Lorsque le second informateur arrive sur la page liée à la conserverie Collin, il n'y a rien de renseigné pour la description. Sa demande sera donc une demande d'ajout. Le modérateur devra traiter avec deux demandes d'ajout. Nous verrons ce qu'il se passe s'il en valide une.

Voyons ce qu'il se passe du coté du modérateur :

Modérateur :

Situons nous au moment du premier ajout.

Le modérateur se connecte et voit cet écran :

Demandes d'ajout :

Nom	Champs	Utilisateur	date
Conserverie Collin	Description	Informateur A	05/05/15

Il clic sur la demande d'ajout qu'il désire examiner. Ici, celle concernant la conserverie Collin.

Cet écran apparaît :

Conserverie Collin

Description :

Créée par Joseph Collin

Valider

Refuser

Il ne peut pas modifier le champs affiché. Il peut seulement valider ou refuser.

En cliquant sur valider, l'information est ajoutée à l'attribut Description de l'objet relatif à la conserverie Collin.

En cliquant sur refuser, la demande est supprimée.

Dans les deux cas, il revient à son écran de liste et constate que la demande n'y est plus.

Nom	Champs	Utilisateur	date

Plus tard, le second informateur réalise la modification (en supposant donc que l'ajout a été validé).

L'administrateur, sur la même page, constate que dans le deuxième champ, celui listant les modifications, une demande a été ajoutée. (On notera que l'administrateur doit rafraichir la page pour voir la demande ajoutée).

Demands d'ajout

Nom	Champs	Utilisateur	date

Demands de modification

Nom	Champs	Utilisateur	date
Conserverie Collin	Description	Informateur B	16/15/15

Il clic sur la ligne relative à la conserverie Collin.

Il arrive sur la page de demande de validation de modification :

Modification

Conserverie Collin
Description

Avant modification

Créée par Joseph Collin

Après modification

Créée par Maurice Collin

Valider Refuser

Il choisit alors de valider ou de refuser l'information. S'il la valide, alors il se passe ce que nous avons décrit, à savoir l'archivage de l'ancienne donnée, et la mise à jour avec la nouvelle donnée. (Dans ce cas, nous montrons que la modération est importante puisque l'information donnée par l'informateur B est fausse)

Précédemment, nous avons évoqué la possibilité que deux demandes d'ajout se fassent :

Demande d'ajout

Nom	Champs	Utilisateur	date
Conserverie Collin	Description	Informateur A	15/15/15
Conserverie Collin	Description	Informateur B	16/15/15

Demande de modification

Nom	Champs	Utilisateur	date
-----	--------	-------------	------

Le modérateur peut valider au plus l'une des deux demandes.

Admettons qu'il choisisse de valider la demande de l'informateur A.

Il revient alors à la page listant les demandes et constatera alors ceci :

Demande d'ajout

Nom	Champs	Utilisateur	date
-----	--------	-------------	------

Demande de modification

Nom	Champs	Utilisateur	date
Conserverie Collin	Description	Informateur B	16/15/15

L'ancienne demande d'ajout de l'informateur B deviendra une demande de modification, et sera traitée comme telle.

La raison : Il y a de fortes chances que les deux ajouts soient des informations relativement similaires. Les traiter toutes deux comme de l'ajout entraînerait à forte chance une redondance des données. Remplacer l'une par l'autre est donc bien plus judicieux.

b. Description des cas d'utilisation

Actions réalisables par les différents acteurs :

	Visiteur	Chercheur	Informateur	Modérateur	Administrateur	Super Utilisateur
Consultation	X	X	X			X
Consultation d'archives		X				X
Modification			X			X
Validation				X		X
Gestion des droits					X	X
Modification de la BDD					X	X

La consultation d'informations : L'utilisateur exécute une recherche via le moteur de recherche sémantique intégré à la page web, et les informations relatives contenues dans la base de données sont alors affichées dans la page.

La consultation d'archives : Il s'agit du même principe que pour la consultation d'informations. Cependant, des informations supplémentaires sont affichées. Ces informations supplémentaires sont classifiées d'archives, et ne peuvent être récupérées par les acteurs non concernés.

La modification d'informations : Ceci implique également l'ajout d'informations. Toute information modifiée est mise en attente d'archivage.

La validation : Toute information mise en attente d'archivage peut être validée ou non validée par un modérateur. Si elle est validée, alors elle est mise à jour dans la base de donnée et supprimée des informations en attente d'archivage. Sinon, elle est supprimée des informations en attente d'archivage.

La gestion des droits : L'administrateur a le droit de promouvoir tout utilisateur à tout poste cités dans le tableau.

La modification de la base de données : L'administrateur a également la possibilité de modifier la structure de la base de données, impliquant une mise à jour de l'interface si nécessaire.

3. Exigences fonctionnelles et exigences sur les données

a. Exigences fonctionnelles

Des exigences supplémentaires pourront être définies durant le projet avant la fin de la phase de conception dans la mesure du possible.

Critères d'appréciation

Rapidité, efficacité, ergonomie.

Besoin que l'interface tourne sur un ordinateur muni d'une connexion internet/ou intranet

Besoin que peu de ressources soient demandées.

Besoin que l'application puisse tourner sur un système vierge avec au moins un système d'exploitation et un navigateur internet.

Besoin d'une ergonomie familière.

b. Exigences sur les données

Nous travaillerons avec la base de données conçue par Marie-Elise Lecoq. Nous avons récupéré son modèle dans le rapport final de François Perez et Lucie Thourault

IV. Exigences non fonctionnelles

1. Ergonomie, utilisation et facteurs humains

a. Interface

Le projet est un interface qui exige une certaine ergonomie que nous détaillons :

L'aspect graphique de l'interface peut être similaire à celui du site du château des Ducs ou du site de l'université de Nantes. Il n'y a pas de demande précise à ce sujet, et c'est une fois que la partie graphique de l'interface sera entamée que nous discuterons de la disposition, la forme et la couleur des menus.

Nous pouvons déjà établir quelques règles de base de l'ergonomie web :

L'interface doit être intuitive et facile d'accès. C'est-à-dire qu'il doit y avoir affordance sur les composants et les actions qu'ils exécutent. L'affordance est le principe selon lequel l'aspect visuel d'un objet suggère également sa fonction. Il faut tendre vers l'objectif des trois clics, c'est-à-dire que chaque page est accessible en maximum trois clics. Il faut maximiser le respect des règles de coloration ergonomique, comme par exemple, préférer le noir sur blanc au blanc sur noir, préférer des couleurs claires et non criardes pour le fond. Éviter les contacts de deux couleurs opposées Posons cela formellement :

Une couleur criarde est une couleur telle que selon les composantes teinte, saturation, luminosité, sur une échelle de 0 à 256, la saturation est supérieure à 128.

Une couleur claire est une couleur telle que selon les composantes teinte, saturation, luminosité, sur une échelle de 0 à 256, la luminosité est supérieure à 128.

Deux couleurs opposées sont deux couleurs telles que selon les composantes teinte, saturation, luminosité, sur une échelle de 0 à 256, si la couleur de la première est X, alors la couleur de la seconde est $Y = X + 128$ modulo 256. Nous fixons arbitrairement une marge de 32 (si la valeur de la seconde couleur est à une distance de moins de 32 de Y, alors on la considère comme opposée.)

b. Style du produit

Le site s'adresse à un publique d'internautes, capables d'utiliser un site internet, capables de lire des boutons et de cliquer dessus, capables d'écrire au clavier les termes de leur recherche et capables de cliquer sur des liens. Il s'agira d'un publique couvrant à la fois des personnes typiques du publique d'un musée, des personnes en vue de réaliser des recherches personnelles, des historiens et des étudiants. Le style du site devra donc être peu coloré, sobre et peu créatif afin de satisfaire au maximum l'utilisateur dans sa recherche et de faire en sorte que l'interface soit le moins possible un frein à son travail de recherche.

c. Facilité d'utilisation

La prise en main du site doit se faire sans apprentissage, en respectant la règle d'affordance citée, et en proposant un lot minimaliste et efficace d'action. Pour cela, chaque action réalisable doit être munie d'un texte qui indique clairement ce qu'elle réalise. Il faut également utiliser les standards du web pour réaliser toutes les entrées.

d. Internationalisation

Le site sera en français par défaut.

Le site devra être traduisible dans d'autres langues, et pourra être traduit dans d'autres langues, pour que des utilisateurs étrangers puissent l'utiliser.

e. Facilité d'apprentissage

Concernant les utilisateurs de type visiteur et chercheur, nous avons déjà fait le point à leur sujet : ils ne sont pas tenus de connaître l'utilisation du site car leurs actions sont sans danger pour la base de donnée, et ils peuvent apprendre en direct à l'utiliser grâce aux règles que nous avons déjà posées.

Concernant les utilisateurs de type modérateur ou administrateurs, ils sont tenus de déjà connaître l'utilisation du site afin de ne pas se tromper dans la réalisation des tâches.

Concernant les utilisateurs de type super utilisateurs, ils sont tenus de déjà connaître l'utilisation du site ainsi que son fonctionnement.

f. Politesse

Pour une utilisation rapide et scientifique du site, il faut prohiber toute formule de politesse et toute formule qui s'adresse directement à l'utilisateur. Le site se doit de décrire de façon formelle les actions qu'il propose.

Par exemple, au lieu d'un message d'erreur « Vous devez s'il vous plait remplir le champ afin de proposer une demande d'ajout », il faudra un message d'erreur « Le champ doit être remplis pour proposer une demande d'ajout ».

g. Accessibilité

Comme la culture est ouverte à tous, le site devra penser aux personnes handicapées en proposant des boutons de grande taille, (au moins 3 fois la taille de leur contenu en hauteur) et un lettrage au minimum en taille 12.

2. Fonctionnement

a. Temps de latence

La latence dépendra du nombre d'utilisateurs connectés. Nous devons donc raisonner en local : les requêtes ne doivent pas prendre plus de 30 secondes pour retourner un résultat. Au delà, il faudrait des barres de chargement, et le développement web proscrit ce genre de pratique. L'utilisateur ne devra donc pas attendre plus de 30 secondes pour l'exécution coté serveur des tâches qu'il réalisera.

b. Disponibilité

Le site ne devra pas avoir besoin de maintenance, et fonctionner seul, 24 h / 24 selon la disponibilité du serveur.

c. Capacité de stockage

Les données qui pourront être ajoutées ou modifiées ne devront pas dépasser plus de 20 ko si ce n'est que du texte. Les vidéos et sons sont à la charge du musée et ne pourront être ajoutées que par celui-ci, mais leur ajout ne dépend pas de l'interface que l'on réalise.

3. Maintenance et portabilité

a. Maintenance

Le site ne doit pas demander de maintenance. Si l'on ne met pas à jour la version de PHP et la version de la base de données, alors il doit théoriquement fonctionner sans aucune maintenance.

Dans le cas d'une mise à jour de PHP, il faudra s'assurer de la portabilité des fonctions utilisées. Il faudra l'aider d'un informaticien pour se faire.

b. Portabilité

Le moteur PHP doit fonctionner sur windows et sur les distributions linux adéquates au fonctionnement d'un serveur PHP, et le code ne doit pas avoir à être changé lors d'un portage.

c. Sécurité

L'application devra empêcher l'injection de code de la part des utilisateurs. Ils ne pourront qu'entrer des chaînes de caractères ne contenant pas de code, ni de balise, ni de morceaux de requête PostgreSQL.

Les mots de passe devront être chiffrés par un clef MD5 avant d'être sauvegardés.

4. Standards

Les pages HTML devront être valides W3C.

Etude bibliographique

Introduction

La quatrième année en informatique à l'école Polytechnique de l'Université de Nantes nous amène entre autre à réaliser tout au long de celle-ci un projet en collaboration avec une entreprise. Nous y apprenons à mettre en application les techniques de gestion de projet que l'on nous a enseignées, et nous y découvrons les différentes étapes de la réalisation d'un projet.

Le centre François Viète de l'Université de Nantes constitue l'entreprise pour laquelle nous travaillons. Leur projet global nommé « Nantes 1900 » est de valoriser la maquette du port de Nantes. Nous devons réaliser une interface web afin de faciliter les différentes transactions avec la base de données.

Cette bibliographie présente plus en détail ce que nous venons de brièvement introduire, et a pour but de nous aider dans la réalisation du cahier des charges, et dans la compréhension des enjeux du projet. Le contexte du projet est très important puisque le projet Nantes 1900 a la particularité d'être mené essentiellement par des étudiants dans le but d'approfondir leur connaissances techniques, mais aussi historiques et culturelles. Donc ce rapport présente en premier lieu l'histoire liée au port de Nantes et à la maquette, ainsi que l'organisation du projet, que nous détaillons plus en détail dans la fiche organisationnelle.

Puis ce rapport présente les projets existants qui peuvent approfondir certains points abordés. Des cartes de Nantes nous en apprennent davantage sur le contexte historique et surtout sur les données susceptibles d'apparaître dans la base de donnée, ce qui nous fournira en jeux de test. L'entrevue avec le réalisateur d'un site contrôlé par un moteur PHP nous permet de découvrir les contraintes et les possibilités de cet outil technique.

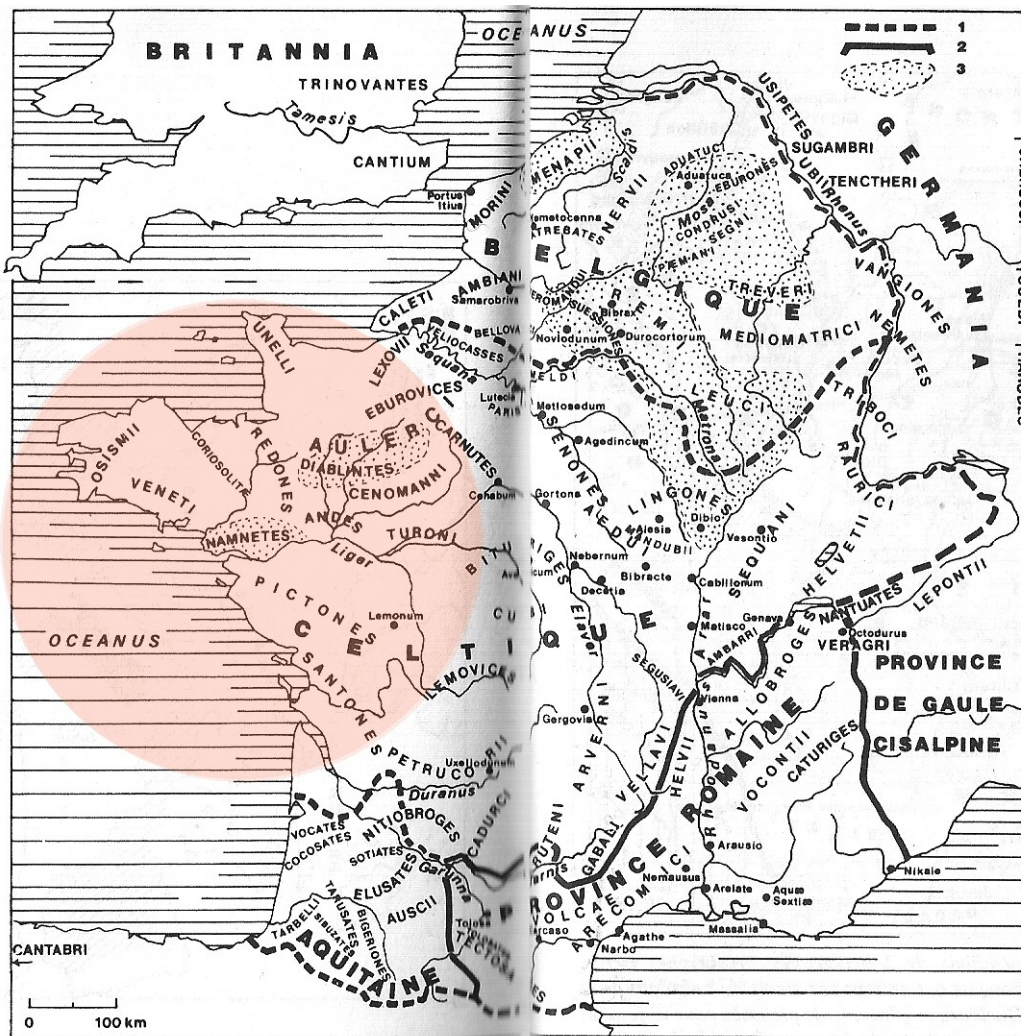
Enfin, nous terminons le rapport sur la partie technique, qui présente tous les outils que nous avons découvert et dont nous nous servirons pour réaliser ce projet. Leur utilisation étant plus précisément détaillée dans le cahier des charges, nous présenterons les avantages et inconvénients ainsi que les possibilités théoriques.

I. Contexte du projet

1. Le port de Nantes

a. Portus Namnetum

Entre 600 et 200 av. J.-C., durant les invasions celtiques, de nombreux peuples se sont installés en Gaule. Au niveau de l'actuel département de Loire-Atlantique, le Nord de la Loire était occupé par les Namnètes. Ce peuple était un peuple de commerçants prudents et avisés et d'hardis navigateurs fluviaux comme de nombreux peuples gaulois. Le Sud de la Loire était occupé par les Pictons.



Situation des peuples de la Gaule en -60. Focus coloré sur la région Armoricaîne et la côte Atlantique

En -57, les Namnètes ont combattu l'invasion romaine et ont été maté par l'armée de César alliée aux Pictons. Cette défaite a donné naissance au Port de Ratiatum (Port de Rezé) tenu par les Pictons et qui est l'actuel quartier Saint-Lupien de la commune de Rezé. En face, au Nord de la Loire, se trouvait Portus Namnetum, bourg Gallo-Romain qui deviendra la Ville de Nantes. L'actuel

Château des Ducs était un Castrum, camp Romain auquel le port était exposé.

Durant la phase Armoricaïne de la Guerre des Gaules en -56, les Vénètes ont invité les peuples Gaulois à se soulever contre l'occupation romaine et ont exigé de Crassus qu'il rende les otages. Jules César s'est alors allié avec les Osismes, les Lexovii, les Ambiliates, les Morins, les Diablintes, les Ménapes et notamment les Namnètes, afin de mener la guerre contre les Vénètes. [1]

À cette époque, des ponts en bois permettaient de circuler entre les îles de Saulzaie, de la Madeleine et de Biesse. Portus Namnetum assurait le lien entre l'Océan Atlantique et la mer Méditerranée via la Loire et le Rhône.

Le bourg prospérera par la négoce de métaux, vins italiens, huiles d'Espagne, sels et poissons. [2], [3]

Au Ve siècle, la Ville se christianise et devient un évêché. Les incursions saxonnes (desquels on a retrouvé les vestiges de pirogues monoxyles et de drakkars dans la Loire) au IVe siècle et des Vikings en IXe siècle ont forcé la ville à se fortifier de plus en plus. Une cathédrale est bâtie près de la porte Saint-Pierre. En 470, les Celtes de Grande-Bretagne migrent en Armorique, c'est la naissance de la Bretagne.

b. Port Commercial

Au XIVème siècle, la guerre de succession a obligé le déplacement des activités portuaires de Nantes vers l'Erdre qui était plus sécurisé que la Loire. Après la fin de l'indépendance des Bretons, Nantes est devenu port Français. Son principal commerce est celui de vin et du sel

Au XVème siècle, François II ordonne l'édification d'un château à l'emplacement du Castrum de Nantes. Selon Joseph Stany Gauthier, conservateur du Château des Ducs dans les années 60, il existe des traces plus anciennes de l'édification d'un fort, notamment un procès à l'empiètement sur les terrains de l'évêque aux environs de 1212. Stratégiquement, le château était bien placé pour défendre le port.

Au XVIème siècle, de nombreux produits textiles ont été ajoutés aux transactions au port de Nantes durant la période de « Boom du textile ». Il est à noter qu'en 1598, l'Edit de Nantes est signé par Henri IV, offrant la liberté de culte aux protestants. [4]

Au XVIIème siècle, des produits de luxe et du sucre venant du Brésil sont venus s'ajouter au commerce du port. Celui-ci est devenu également en partie un port poissonnier avec l'arrivée des Espagnols installés à la Fosse. De nombreux mouvements xénophobes ont été menés contre les portugais et les hollandais. Ces derniers sont accusés notamment d'être à l'origine des problèmes de tonnage en jetant du lest dans l'eau. Mais on suppose aujourd'hui que l'ensablement du fleuve avait pour origine la déforestation du Massif Central et du Morvan. De plus, le port de Nantes était alors reconnu à cette époque comme Port International. En conséquence, on a émis l'idée de le doter d'un avant-poste. On notera la visite de Louis XIII et Richelieu en 1614 et de Louis XIV en 1661 puis sa révocation de l'Edit de Nantes en 1685. [3]

Au XVIIIème siècle, le Code Noir émis par Louis XIV a permis au port de faire de nouvelles transactions : celles du tabac et des esclaves. Le sucre est resté le produit dominant avec de nouvelles importations depuis Saint-Domingue (en actuelle République Dominicaine) . Le Port a acquis la dénomination de port colonial et premier port négrier de France. Avec le commerce triangulaire, Nantes s'est accaparé de 40% des expéditions négrières françaises jusqu'à l'abolition de l'esclavage en France en 1848. Le port s'est agrandi : une fonderie de canons royale a été construite en 1777 et de nombreux ponts avec chaussée traversaient le lit de la Loire. Les premiers armateurs Nantais remontent à cette époque : les Mautondouin.

c. Port Industriel

Durant la première moitié du XIXe siècle, le port de Nantes a commencé son industrialisation :

La stérilisation d'Appert a été à l'origine de l'implantation des conserveries Pierre Joseph Colin en 1822. Joseph Collin était reconnu comme fondateur de la conserverie industrielle. Nantes est ainsi devenu capitale de la sardine en boîte. Le pâtissier Jean-Romain Lefèvre est arrivé à Nantes en 1846 et s'est marié à une demoiselle Utile. Leur fils Louis Lefèvre-Utile a fondé alors sur l'île de la Madeleine la biscuiterie LU. La fonderie de canons royale s'est transformée en 1828 en usine de machines à vapeur. Les usines de raffinage du sucre se sont transformées en métallurgies. Au milieu du XIXe siècle, la basse Loire lançait le quart des navires français. On a créé en 1838 le port de Saint-Nazaire, résultat du projet de construire un avant-poste au port.



165 NANTES. — Le Pont Transbordeur. — LL.

Durant la seconde moitié du XIXe siècle, le port importait principalement du sucre, du riz et de la houille et exportait des métaux, des produits métallurgiques et des produits alimentaires. Il est devenu le 6ème port de France. En 1868, constatant que le trafic était plus important à Saint-Nazaire, on a décidé créer un canal latéral à la Loire qui sera finalisé en 1892. Le canal de la Martinière mesurait 15 km de long et a dû être abandonné en 1910, devenant un cimetière pour 3 mâts. On a vu à cette époque la construction du pont transbordeur, monument impressionnant de l'ère industrielle, qui permettait de traverser la Loire au moyen de plateformes et des câbles.

On notera la construction de Belem [5][6][7] en 1896 par les chantiers Dubigeon à Chantenay à l'emplacement de l'actuel ancien bâtiment de la Bourse. Le trois-mâts était chargé de l'importation de cacao depuis le Brésil, puis a appartenu à Guinness, en Irlande. Il faudra attendre 1986 pour son retour à la ville de Nantes. Un plan de destruction à la dynamite des îles est mis en place afin d'élargir la Loire et abaisser son fond. Il est exécuté de 1903 à 1914.



Le Belem. Tableau du Musée des Salorges, Nantes. Image numérisée par Laurent Claude Gloaguen (lcg.net)

Les problèmes de déblayage du lit de la Loire resteront dans les esprits jusqu'à de nos jours. Par exemple, en 1979, il a été nécessaire de déblayer 2000 mètres cubes de roche dans le lit de la Loire car les nouveaux navires de l'époque avaient besoin de 9 mètres sous la ligne de flottaison. Cela a permis l'arrivée de navires transporteurs d'agrumes venus du Maroc. [8]

2. La maquette

a. Son histoire

L'Exposition Universelle de 1900 a eu lieu à Paris, et c'est dans ce cadre que la Chambre de Commerce a commandé la réalisation d'une maquette du port de Nantes. L'objectif était de présenter l'aménagement de la Loire au niveau de la ville de Nantes et les industries et monuments qui résidaient dans ces environs. Elle a été modifiée jusqu'en 1913 et la ville de Nantes l'a acquise en 1927. Le musée de Nantes l'a exposée jusqu'en 1960. On a pu encore l'admirer de 1974 à 1994 dans le musée du Château des Ducs. En 1996, elle a été restaurée et c'est depuis la réouverture du musée du Château des Ducs en 2007 qu'elle a de nouveau été exposée au public. [9]



L'exposition du Champs de Mars au Trocadéro - 1900

Ce qui a motivé la réalisation de la maquette était la puissance et la célébrité de Nantes à cette époque. Des trains traversaient la ville par 30 passages à niveau, 350 usines étaient installées, et de nombreux chantiers étaient bâtis le long de la Loire. Nantes était connue dans toute la France, c'est d'ailleurs la seconde ville du pays la plus chantée. Elle était également célèbre pour ses biscuiteries (LU, B.N.). Enfin, le port était le 4ème plus grande France et le plus grand bassin d'emploi jusqu'à la Première Guerre Mondiale. [2]

b. Ses caractéristiques

La maquette, réalisée par le menuisier Paul Duchesne, couvre la superficie depuis l'ancienne Bourse de Nantes jusqu'aux environs de Chantenay, soit 3 km sur 500 m. Ses dimensions sont de 920 sur 185 cm. Elle a été réalisée à partir de 6 matériaux : du bois, des métaux, des fibres textiles, du papier, du carton et du verre. [10]

On peut partager la maquette en trois grandes modélisation 3D (cf II-2). [2]

parties sensiblement égales en superficie : Les

Chantiers à l'Est, le Butte Sainte Anne au centre

et la zone industrielle à l'Ouest. Sur les cours La photo ci-contre montre l'étendue de la d'eau, de nombreux bateaux ont été modélisés, superficie couverte, et la finesse de la pour la plupart des 3 mats à voile et en fer. Le modélisation telle qu'on peut la voir au premier réseau ferroviaire et routier a été reconstitué de plan. On regarde ici la maquette depuis l'Est.

façon détaillée et tous les bâtiments ressortent comme des volumes. Ce qui a été matière à la

c. Son exposition

La maquette est exposée en Salle 21 du Château des Ducs, et est placée dans un angle. Cette disposition et la vitres protectrice empêche le visiteur d'avoir d'autre point de vue que de l'Est et le Sud de la maquette.



Maquette du Port de Nantes

Le problème a alors été posé : le visiteur ne peut pas apprécier la finesse des détails sur toute la superficie de la maquette. Anne Guigun, une stagiaire d'Histoire, a mené une étude auprès des visiteurs du musée, qui a révélé que de nombreuses personnes avaient du mal à se repérer sur la maquette et désiraient un plus haut degré d'interactivité avec celle-ci.

C'est pourquoi le projet Nantes 1900 a été lancé, afin de palier à ces lacunes, en numérisant la maquette en 3D et en ayant pour objectif d'offrir au visiteur la possibilité d'obtenir de nombreuses informations sur tous les éléments de la maquette avec un système interactif de projecteurs et d'écrans.

3. Le projet Nantes 1900

a. Commanditaires

Le château des Ducs de Bretagne [11] est le commanditaire du projet Nantes 1900, qui est sous la responsabilité de Bertrand Guillet et Christophe Courtin. L'objectif est de valoriser la maquette d'un point de vue historique et muséographique.

Bertrand Guillet est conservateur en chef du patrimoine et directeur adjoint du château des Ducs de Bretagne. Christophe Courtin est chargé du multimédia et de l'audiovisuel au château. Ils ont confié la partie technique du projet Nantes 1900 au centre François Viète, à Jean-Louis Kérouanton et Florent Laroche. Jean-Louis Kérouanton est docteur en histoire et maître de conférence à l'Université de Nantes et Florent Laroche est maître de conférence à l'école Centrale de Nantes et chercheur au laboratoire IRCCyN ¹. Ils nous ont chargé de réaliser l'interface pour la base de données relative à la maquette, suite du projet transversal précédent confié à François Pérez et Lucie Thourault.

b. Centre François Viète.

Le centre François Viète [12] [13] [14] [15] se trouve sur le campus de la Fac de Science de Nantes. C'est un pôle de recherche dans l'épistémologie et l'histoire des sciences et des techniques.

Le centre réaliste neuf thèmes de recherche qui sont toutes axées autour de l'épistémologie. Il enseigne aussi aux étudiants en vue d'obtenir un master 2 d'Histoire des Sciences et des Techniques qui s'organisent autour de l'étude épistémologique et historique des sciences en général. Enfin, le centre soutient et encadre des thèses sur l'histoire des sciences et des techniques depuis 1996.

Les membres de l'équipe publient de nombreux ouvrages et articles et organisent des colloques et des séminaires. La revue « Cahier François Viète » publie également certains de leurs résultats de recherche.

c. Organisation du projet

Le projet est ouvert et évolutif et laisse libre cours à l'imagination et la créativité. La demande du client est traduite et les objectifs évoluent tout au long du projet de façon à offrir un résultat à la pointe de la technologie selon les attentes du Château des Ducs. Le projet est pluridisciplinaire, c'est à dire qu'il implique les compétences d'acteurs de l'école Polytechnique, Centrale et de l'Université de Nantes. C'est un lien étroit qui est fait entre histoire, culture et technique.

Pour le projet, une plateforme collaborative de travail (team work place of Lotus) est utilisée. Elle permet de centraliser sur un serveur l'ensemble des travaux et rapports, et de mettre en communication les différents acteurs de l'équipe. Ainsi, la traçabilité est assurée. [16]

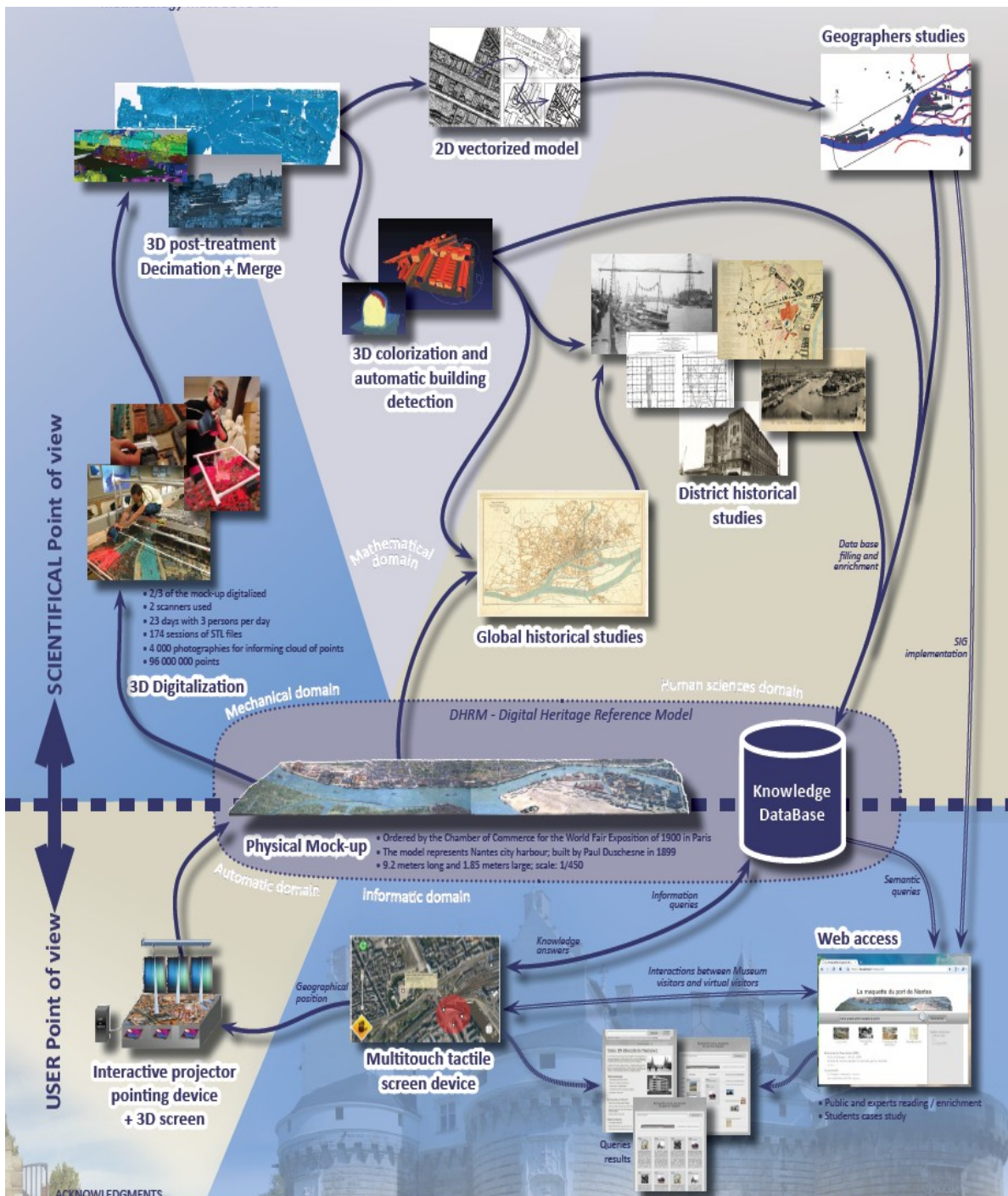


Schéma récapitulatif de l'ensemble du projet et des différentes tâches et interactions entre les équipes.

Selon ce schéma récapitulatif, nous sommes chargés de la partie « Web access ». On peut voir le volume et la diversité des travaux que met en jeu ce projet.

II. Projets similaires

1. Reproduction de Nantes

a. Le nouveau plan de Nantes monumental, industriel et commercial

Le plan monumental de Nantes a été produit en 1895 et publié à Nantes par Théodore Veloppé. Il avait pour objectif d'établir une nomenclature des établissements industriels et commerciaux et de les rehausser au niveau de monuments. Il présente le plan de la ville et particulièrement au niveau du port, sous un angle de vue plongeante à 90°, avec une représentation des établissements reconnus en vue cavalière, comme la vue « oblique » qu'offraient les montgolfières à cette époque. Le fond du plan est anamorphique afin de donner un effet de perspective. De nombreux établissements ne sont pas représentés directement sur la carte, et apparaissent alors sous forme de vignette dans les marges. C'était un plan touristique plus qu'un plan urbain.



Plan monumental de Nantes.

L'intérêt de ce plan est qu'il vient compléter notre étude sur le contexte de la maquette. En effet, il apporte de nouvelles informations sur la Nantes de l'époque qui nous intéresse.

Le centre-ville était vierge d'industries, qui se sont principalement développées au bord des voies d'eau et à proximité des chemins de fer, essentiellement au niveau de la Prairie au Duc, des l'île de Gloriette, du quartier Launay, de la Ville-en-Bois, du nord de Chantenay et du Sud du bourg.

On peut énumérer les usines présentes : des métallurgies et des usines d'engrais et produits chimiques, côtoyant des industries alimentaires, des conserveries, des biscuiteries, des fabriques de boîtes de conserve, des raffineries de sucre, des usines de traitement de produits exotiques, des huileries et des savonneries, qui étaient la tradition commerciale de Nantes comme en témoignent les échanges que nous avons énuméré précédemment.

D'autre pôle était présent : les filatures, tissages et usines de chaussures, ainsi que les

brasseries, distilleries et fabriques de liqueurs.

Le plan s'arrête à l'Ouest au niveau du boulevard circulaire, tracé sous l'agent voyer en chef du département Eugène-Orieux, et qui était la nouvelle limite formelle de la ville de Nantes. Doulon n'était pas contre l'annexion, ce qui n'était pas le cas de Chantenay à cette époque.

On peut notamment voir la gare de l'État, sur le site industriel de la prairie au Duc, implantée en 1886 et longeant le Boulevard Babin-Chevaye qui traverse l'île. C'est un élément majeur de la partie Est de la maquette. [17]

b. Le plan Delattre de Nantes

En 1909, la mairie de Nantes a émis la volonté de montrer la force de la métropole nantaise. Etienne Coutan, un architecte de l'époque, a proposé ses services à la ville de Nantes pour instituer une politique cohérente des transformations urbaines. Il est devenu directeur des services des bâtiments communaux et des plantations urbaines. E. Chevrier, agent technique principal du projet, assura alors la paternité du plan officiel de la ville.

L'objectif était de montrer le territoire et la banlieue de la métropole nantaise, organisée sur de grands axes de circulation autour de fleuves et de rivières.

L'intérêt de ce plan dans le cadre de notre projet est qu'il montre la ville de Nantes à un instant particulier, l'année 1909, au milieu des différentes mises à jour de la maquette. On peut notamment apprendre l'acquisition de l'île de Cheviré, la présence en masse d'HBM (Habitations à Bon Marché) au niveau du Bois Hardy, selon la Loi Siegfried de 1894, et la tentative de donner un nouveau centre à la commune industrielle au niveau du Boulevard de l'égalité lors de l'annexion de Chantenay.



*Plan Delattre. Archives
municipales de Nantes*

Une note intéressante nous apprend que les automobiles pouvaient rouler jusqu'à 20km/h, 10km/h en agglomération et 5 km/h dans le centre, et que le coup de trompe était obligatoire à chaque croisement. Cette mise en ambiance de l'époque nous a permis de nous rendre compte d'une chose qui a pu échapper à de nombreux visiteurs du château : Les bateaux y sont excessivement représentés, mais il y a absence totale d'automobile sur la maquette. [18]

2. Exemple de site dynamique : reduction-image

a. Présentation

Le site de réduction d'image www.reduction-image.com est un site dynamique dont les pages sont générées par un moteur PHP. Nous avons pu obtenir une entrevue avec le créateur du site afin de mieux comprendre les enjeux de la création d'un site dynamique et de bénéficier des conseils de développement qu'il a à nous offrir.

Reduction-image a été réalisé en PHP5.2, il est hébergé sur un serveur Iliad, qui est un serveur dédié, sur une distribution Ubuntu 5.04 LTS. Il est classé dans l'ordre des 700 000 sites mondiaux sur le classement Alexa et a un rang de page google de 3.

Le site permet à l'utilisateur de réaliser divers traitement sur l'image, dont le changement de dimensions, l'application de filtres, la création de gifs animés ou encore la création d'animations audio-visuelles.

Il est traduit en six langues par des interprètes professionnels.

Le concepteur du site n'est pas du métier de l'informatique mais du médical et tient à tenu à conserver l'anonymat. La création du site a totalisé 1200 heures de travail. La seule maintenance du site concerne la programmation et l'évolution de la version de php mais la maintenance reste nulle car le site peut tourner sans aucune intervention.

b. Outils techniques

Le concepteur du site a choisi le langage PHP car c'est un langage serveur qui construit des pages HTML. C'est un standard auquel répondent tous les navigateurs, et l'utilisateur n'a aucun réglage à faire à son navigateur, contrairement à ce que demanderait l'utilisation de javascript par exemple.

La partie sonore et vidéo du site ne devaient pas dépendre d'un comportement navigateur, avec une installation de module complémentaire obligée au client. C'est pourquoi ces médias sont implémentés au sein d'un serveur de son et de vidéo utilisant le principe du lecteur flash. Toutes les ressources médiatiques et les lecteurs sont fournis par le serveur et l'utilisation n'a besoin que du classique client flash 8.0 minimum. Lorsque la prochaine version d'HTML fournira les balises <video> et <audio>, le site les utilisera tout en conservant sa philosophie du tout fonctionnant coté serveur.

Le site utilise de nombreuses bibliothèques telles que GD2 (bibliothèque graphique intégrée au PHP), la librairie Image Magic, Gifcicle et ffmpeg. La librairie Image Magic peut être interfacée au PHP via des fonctions exécutables. Une autre méthode consiste en l'utilisation d'une classe associée « Imagic » qui gère la librairie, mais le concepteur ne se situant pas dans la génération de la programmation objet a préféré continuer par l'exécution de fonction. Dans notre cas, nous devons traiter avec l'utilisation de PostGreSQL et nous pourrions l'interfacer au PHP avec une classe appelée PDO mais nous y reviendrons dans la partie technique de ce rapport.

Toutes les fonctions sont placées dans une page sécurisée car chiffrée. Ce qui nous amène

aux problèmes de sécurité liés au langage PHP.

c. Conseils sur le développement en PHP

La sécurité est primordiale dans le développement de programmes en PHP. En effet, le site a déjà subi des attaques et est désormais plus robuste au niveau de la sécurité. Les principales attaques se font au niveau des formulaires et c'est pourquoi ils sont sécurisés avec des fonctions qui empêchent au maximum l'injection de code.

Le site s'est également muni d'une protection supplémentaire depuis l'attaque récente d'un algérien sur les sites hébergés par MavenHosting (l'un des principaux hébergeurs canadiens) : il est parvenu à substituer toutes les pages principales en PHP de façon à les rediriger vers d'autres sites. La question de sécurisation de la propriété intellectuelle s'est alors posée et le site reduction-image a sécurisé toutes ses fonctions en les chiffrant. Le procédé est telle que le script est déchiffré avant d'être exécuté, sachant que l'algorithme de déchiffrement vérifie en partie l'adresse IP du serveur.

Selon le concepteur du site, le PHP est un langage abondamment utilisé de nos jours et les ressources et aides documentaires sont nombreuses. Il nous a fourni plusieurs adresses afin de nous aider dans le développement :

- <http://www.phpscripts-fr.net/> : une certaine rigueur sur le développement en PHP et de nombreux conseils
- <http://www.asp-php.net/> : aide sérieuse et correspondance entre PHP et ASP
- <http://www.php.net/> : site de référence avec manuels, commentaires et améliorations fournies par les utilisateurs.

Avec l'étude de ce site et l'entrevue avec son concepteur, nous avons désormais choisi l'utilisation du langage PHP pour réaliser l'interface. Nous allons voir les outils dont nous disposons, leurs contraintes et les possibilités qu'ils nous offrent.

III. Outils techniques

1. Internet et communication client-serveur

a. Le SGBD

Le SGBD² utilisé pour le projet est PostgreSQL [19] (qui a la particularité d'être un SGBDRO³, nous reprendrons ce point par la suite). L'intégrité des données est assurée par des déclencheurs et les appels par un programme en module de redirection en Java.

Les données inscrites dans la base sont géo-référencées, c'est à dire que ce sont des objets localisés géographiquement dans le monde (ici au niveau du port de Nantes) et qu'il est possible d'utiliser ces informations pour des opérations géographiques.

Ainsi, à ce SGBD a été implanté un module SIG⁴ qui permet de faire des opérations d'ordre spatial comme les calculs d'aire ou de distance par exemple. Le SRID⁵ permet une utilisation des coordonnées selon plusieurs normes. Par exemple, il gère les coordonnées de type latitude et longitude. Pour ce projet, le système utilisé est le Lambert II qui était utilisé au XXe siècle. L'identifiant de référence est le 17582.

Les données inscrites dans la base sont également référencées bi-temporellement, avec une date de début et une date de fin, qui peuvent représenter au choix la construction, la création, la destruction, le changement de propriétaire etc... Elles sont aussi munies de date d'ajout et de modification dans la base, ce qui permet notamment de gérer l'archivage.

Ainsi nous sommes munis pour ce projet d'un SGBD gérant une base de données déjà architecturée et fonctionnelle, que nous pouvons modifier si nécessaire.

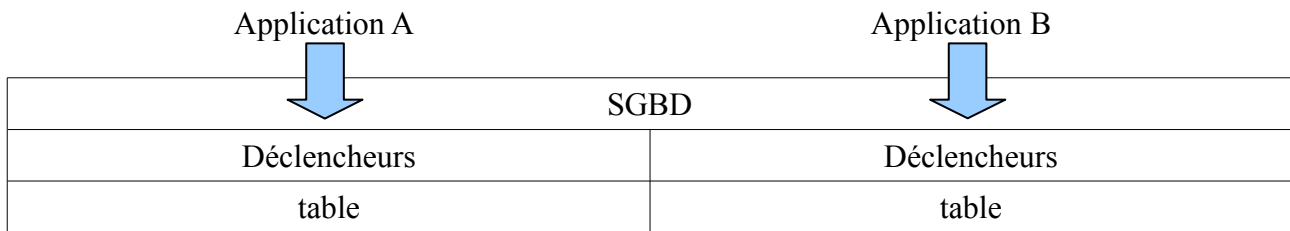
b. L'architecture client-serveur

Le projet nous a été confié au stade suivant : un moteur permet de générer des requêtes dans la base de données, et le moteur de celle-ci retourne des données brutes. Le fonctionnement est assuré à ce niveau. Là où nous intervenons, c'est dans la couche interface de ce fonctionnement. Nous devons réaliser un site qui permet aux utilisateurs d'utiliser ce moteur de recherche et de récupérer les données sur une page web avec une gestion de liens avec lesquels ils peuvent naviguer sur d'autres données.

Ce projet s'inscrit donc dans la réalisation d'une architecture client-serveur. En effet, selon cette architecture, la machine du client ne réalise par d'opérations, elle se contente d'envoyer des requêtes et de recevoir du contenu. Le moteur qui fait tourner cela fonctionne du côté du serveur. Il s'agit d'une architecture qui fait travailler un ensemble en sous effectif de machines « serveur » au service d'un ensemble abondant de machines « client ».

L'architecture client-serveur [20] permet de préserver l'intégrité des données lors de la récupération d'informations sur le SGBD à l'aide des classiques déclencheurs et contraintes, ces filtres logiques qui sont exécutés implicitement lors des accès aux tables afin d'en assurer l'intégrité. La machine cliente ne peut pas corrompre ces données puisque toutes ses actions dépendent de la

machine « serveur » qui les réalise à sa place. Chaque application coté serveur doit passer par les déclencheurs pour accéder aux tables comme le montre ce schéma :



L'architecture client-serveur apporte également une uniformisation de l'aspect du comportement ce qui accélère la prise en main. Une étude réalisée par un éditeur de traitement de texte a montré que 30% du temps passé à l'accomplissement d'une tâche est perdu au maniement du curseur. L'objectif de l'uniformisation de l'interface est de diminuer ce temps en développement de façon transparente des automatismes d'utilisation. Ainsi, nous pourrions user de menus déroulants, représentation iconique du pointeur modal. Ce qui nous amène à l'utilisation du HTML⁶ qui sera le support utilisé pour les pages que le client recevra.

c. PHP

Il nous faudra utiliser un outil qui permette de faire travailler le serveur de façon à ce qu'il génère les pages HTML que l'utilisateur désire en utilisant le SGBD, puis qu'il lui envoie ces pages construites.

Le PHP 5 [21] [22] s'avère être le langage approprié pour un tel projet. En effet, c'est un langage de script Open Source (sans licence) qui fonctionne coté serveur. C'est-à-dire que le code est exécuté par le serveur, qui construit la page HTML associée, et qui l'envoie au client (le serveur est la machine qui contient le SGBD et le client est la machine de l'utilisateur). La page reçue par le client ne contient aucun fragment du script, et l'utilisateur ne peut donc pas accéder au code qui a généré sa page.

Au niveau du code, le PHP s'intègre bien mieux qu'en C, en Perl ou tout autre langage permettant en partie de générer des pages HTML, car il s'incruste à la page avec un format de balise. Le HTML est généré en plaçant tous les éléments entre des balises du type :

```
<une balise A> contenu </une balise A>
```

Le code PHP s'intègre ainsi :

```
<? php code ?>
```

C'est une balise unique qui n'attend pas de balise de fermeture, et que l'on peut placer à tout moment dans la page HTML comme contenu.

Le PHP offre la plupart des outils de base d'un langage de programmation : constantes, variables, types, tableaux, opérateurs, structures de contrôle, fonctions ou encore classes (ce dernier est encore à l'état de prototype, mais reste utilisable et pratique, dans le cadre d'une programmation

orientée objet).

Le PHP assure également le lien avec les formulaires HTML, ce qui offre à l'utilisateur une panoplie d'entrées, lui permettant d'interagir avec le programme. Il faut néanmoins traiter avec tous les problèmes de sécurité, et restreindre les possibilités d'interaction au strict nécessaire, notamment empêcher toute injection de code.

Enfin, le PHP offre de nombreux outils d'accès et de traitement des bases de données, notamment l'outil PDO qui assure le lien entre le programme PHP et les requêtes au SGBD. L'avantage est qu'il est multi-langage.

Ainsi, toutes ces possibilités offertes par le PHP 5 couvrent l'ensemble de nos besoins technique au niveau du moteur qui doit faire un lien entre la génération de pages web et le SGBD.

2. Bases de données

a. Du SGBDR au SGBDO

Le SGBDR [23] présente un modèle de données, qui permet le stockage permanent, le partage inter-utilisateur et la gestion d'information.

Des applications permettent d'accéder aux données, les modifier et les agencer. Le SGBDR permet la gestion de données sur tous les sujets, c'est pourquoi un SGBDR est utilisé pour des données géoréférencées et historiquement localisées sur la maquette.

Les objectifs du SGBDR sont de

- Préserver l'intégrité des données : les données restent exactes et bien formées.
- Préserver l'accessibilité des données : les données doivent être accessible à de nombreux utilisateurs, et ce, au moindre coût.
- Préserver la sécurité des données : un système de gestion des utilisateurs permet la restriction des données, et un système anti-collision empêche les éditions simultanées d'une même donnée.

Le SGBD qui nous intéresse a la particularité d'être un SGBDO. [24]

Les objectifs supplémentaires du SGBDO sont de

- Permettre la définition de type d'objet : des instances de classes sont stockées.
- Prendre en compte les opérateurs génériques : les opérateurs liés aux classes sont pris en compte dans la gestion des données objet de ces classes.
- Encapsuler les données : le contenu des classes est transparent.
- Permettre le principe d'héritage : une classe d'objet peut hériter des attributs et méthodes d'une autre classe d'objet.

Les SGBDO [25] utilisent le langage de définition ODL⁷ Il va nous servir à mettre à jour les

tables ou à modifier les données. Il a l'avantage d'être proche de la programmation orientée objet.

Dans un cours de l'Université de Californie [26], nous avons trouvé un exemple basique qui montre à quel point l'ODL est proche de la programmation orientée objet, à travers la création d'une classe Client qui a différents attributs tels qu'un nom, une adresse structurée ou une collection de connaissances.

```
interface Customer {
    attribute string name; attribute integer ss#;
    attribute Struct Addr {string street, string city, int
    zip} address;
    relationship Set<Loans> borrowed
    inverse Loans::borrower;
    relationship Set<Branch> has-account-at
    inverse Branch:patrons;
    key(ss#)
}
```

Là où nous utilisons habituellement le langage SQL⁸ pour réaliser des requêtes, le SGBDRO va nous offrir l'opportunité de manier le OQL⁹. Le OQL ne bénéficie pas d'une normalisation aussi importante que le SQL, c'est pourquoi le groupe précédent s'est chargé de s'orienter vers un modèle relationnel-objet : la SGBDRO. L'avantage est de pouvoir bénéficier de nombreuses années d'expertise d'utilisation du modèle relationnel. [27]

b. Avantages et inconvénient

La base de données est très volumineuse et les requêtes ont besoin d'une optimisation particulièrement importante. Le SGBDRO offre justement la possibilité de réaliser des requêtes sur une grande quantité d'information en un temps optimisé. L'approche objet permet de restreindre la gestion des contraintes tout en préservant la cohérence des données.

L'inconvénient d'un tel fonctionnement est le poids des données qui est bien plus élevé que pour un SGBDR. L'administration est également rendue plus complexe. L'autre point négatif, nous l'avons déjà abordé, c'est le manque d'expérience de l'humanité en matière de SGBDRO. C'est pourquoi le groupe précédent a choisi d'utiliser PostgreSQL, un SGBD Relationnel-Objet.

c. Lien entre la SGBDRO et le PHP

L'approche orientée objet depuis PHP 5 va nous permettre de récupérer les données sous forme d'objet. En effet, l'outil PDO [28] dont nous avons parlé peut se voir ajouter un pilote appelé PDO_PGSQL. Cet outil nous permet de nous connecter aux bases PostgreSQL, d'y modifier et d'en récupérer les données.

Un exemple de récupération de données :

```
<?php
$db = new PDO('pgsql:dbname=test host=localhost', $user, $pass);
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$db->beginTransaction();
$stmt = $db->prepare("select oid from BLOBS where ident = ?");
$stmt->execute(array($some_id));
$stmt->bindColumn('oid', $lob, PDO::PARAM_LOB);
$stmt->fetch(PDO::FETCH_BOUND);
fpassthru($lob);
?>
```

L'objet db est une instance de connexion à la base de données, dans lequel on effectue une requête (« select oid from BLOBS where ident=? »). Une fois exécutée, la requête envoie un résultat que l'on peut récupérer dans une variable PHP. De là, nous pouvons à volonté créer la classe correspondante à l'objet récupéré et traiter toutes les données de façon orientée objet.

IV. Conclusion

Le projet Nantes 1900 est un projet très vaste qui rassemble sciences techniques et sciences humaines. L'un de ses buts est de rapprocher les acteurs du projet, des étudiants pour la plupart, de la culture et de l'histoire associées au thème de la maquette. Le thème en question, le port de Nantes, est un fragment de l'histoire de France qui remonte jusqu'à l'antiquité. On comprend ainsi mieux les causes historiques de l'état du port actuel et ce qui a motivé à la réalisation de la maquette. Comme cette maquette n'est actuellement pas mise en valeur, le projet, mené par le centre François Viète et par le Musée d'Histoire de Nantes, a pour objectif de valoriser celle-ci sur le plan muséographique et historique en faisant appel à la science.

Pour notre projet qui consiste à réaliser l'interfaçage de la base de donnée relative à la maquette via un site web, nous avons étudié au préalable deux projets existants, l'un en rapport avec le contexte du projet, et l'autre en rapport avec les besoins technique du développement web. Ainsi, les cartes de Nantes nous ont apporté de nombreuses connaissances sur les types de données susceptibles d'être traitées par la base de donnée, et l'entrevue avec le concepteur du site de réduction d'image nous a apporté de bons conseils et une bonne base de travail sur le développement web en PHP.

Ainsi nous avons les cartes en main pour trouver les outils techniques qui nous ont servi pour la réalisation du cahier des charges et que nous utiliserons dans la phase de conception du projet. Nous avons découvert le principe de fonctionnement des bases de données relationnelles objet et des pages web dynamiques. Ces deux outils imposent certaines contraintes mais s'avèrent très puissants dans le cadre du projet que l'on doit réaliser. Un développement plus détaillé de leur utilisation est décrite dans le cahier des charges. Après ce rapport et la validation du cahier des charges, nous pourrions entamer la phase de conception en prenant en compte les spécifications du cahier des charges et la bonne utilisation des outils que nous avons découverts.

V. Glossaire

¹ IRCCyN :

Institut de Recherche en Communication et en Cybernétique de Nantes.

² SGBD :

Système de Gestion de Base de Données.

³ SGBDRO :

Système de Gestion de Base de Données Relationnelles Objet.

⁴ SIG :

Système d'Information Géographique.

⁵ SRID :

Spatial Reference System Identifier.

⁶ HTML :

HyperText Markup Language.

⁷ ODL :

Object Definition Language.

⁸ SQL :

Structured Query Language.

⁹ OQL :

Object Query Language.

Procédure de Recette

Numéro du test	1
Fonction testée	Connexion visiteur
Cas de test	Tester si la page par défaut du site est l'accueil des visiteurs
Scenario	Entrer sur le site par la page principale
Comportement attendu	La page d'accueil des visiteurs est affichée
Comportement observé	

Pour réaliser les tests 2, 3 et 4, il faut réaliser le test 5 puis le test 21, puis le test 4, et enfin les tests 22, 23 et 24.

Numéro du test	2
Fonction testée	Connexion chercheur
Cas de test	Tester si la connexion d'un chercheur fonctionner
Scenario	Depuis la page principale du site, entrer un identifiant et un mot de passe chercheur
Comportement attendu	La page d'accueil des chercheurs est affichée
Comportement observé	

Numéro du test	4
Fonction testée	Connexion administrateur
Cas de test	Tester si la connexion d'un administrateur fonctionner
Scenario	Depuis la page principale du site, entrer un identifiant et un mot de passe administrateur
Comportement attendu	La page d'accueil des administrateur est affichée
Comportement observé	

Numéro du test	5
Fonction testée	Connexion super utilisateur
Cas de test	Tester si la connexion d'un super utilisateur fonctionner
Scenario	Depuis la page principale du site, entrer un identifiant et un mot de passe super utilisateur
Comportement attendu	La page d'accueil des super utilisateur est affichée
Comportement observé	

Pour tous les tests qui suivent, il faudra utiliser les comptes qui viennent d'être créé précédemment lorsque la connexion à un compte est demandée.

Numéro du test	6
Fonction testée	Récupérer des informations
Cas de test	Tester la fonction de requête et de récupération d'information pour l'usine LU
Scenario	Depuis la page principale du site, entrer le nom d'un bâtiment déjà existant dans la base de donnée (ex : L.U.) et valider la recherche
Comportement attendu	La page de description de l'élément correspondant (ex : L.U.) est affichée
Comportement observé	

Numéro du test	7
Fonction testée	Obtenir plus d'informations
Cas de test	Tester la fonction d'extension sémantique
Scenario	Depuis une page de description, cliquer sur un lien au choix afin d'obtenir plus d'informations.
Comportement attendu	Plus d'informations doivent être affichées
Comportement observé	

Numéro du test	8
Fonction testée	Visionner une video
Cas de test	tester le visionnage de video pour l'usine LU
Scenario	Se loguer en visiteur. Faire une recherche sur l'usine LU. Choisir une video et cliquer dessus.
Comportement attendu	La video se met en marche.
Comportement observé	

Numéro du test	9
Fonction testée	Accéder à l'ajout d'information
Cas de test	tester l'ajout d'information sur l'usine LU
Scenario	Se loguer en informateur. Chercher l'usine LU, sur la page d'information liée à l'usine LU, cliquer sur le champ description.
Comportement attendu	La page proposant un éditeur de texte apparait.
Comportement observé	

Numéro du test	10
Fonction testée	Proposer l'ajout d'information
Cas de test	tester l'ajout d'information sur l'usine LU (suite)
Scenario	Suite au test 9, inscrire « test 10 » dans le champs et cliquer sur valider.
Comportement attendu	la page d'information liée à l'usine LU apparaît avec un message indiquant que l'information a été proposée à l'ajout.
Comportement observé	

Numéro du test	11
Fonction testée	Confirmer l'ajout d'information
Cas de test	tester l'ajout d'information sur l'usine LU (suite)
Scenario	Suite au test 10, se loguer en tant qu'administrateur.
Comportement attendu	Sur la page principale d'administrateur, dans le champs des requêtes d'ajout, une requete d'ajout pour l'attribut description de l'usine LU doit apparaitre.
Comportement observé	

Numéro du test	13
Fonction testée	Confirmer l'ajout d'information
Cas de test	tester l'ajout d'information sur l'usine LU (suite)
Scenario	Suite au test 12, cliquer sur valider.
Comportement attendu	La page principale d'administration doit apparaitre.
Comportement observé	

Numéro du test	14
Fonction testée	multiples demande d'ajout d'information
Cas de test	tester deux ajouts d'information pour l'usine LU.
Scenario	Vider les données de test. Reprendre les tests 9 et 10 deux fois en proposant comme information « test 9 - 1 » et « test 9 - 2 ». Puis reprendre le test 11
Comportement attendu	Deux demandes d'ajout doivent etre affichées.
Comportement observé	

Numéro du test	15
Fonction testée	multiples demande d'ajout d'information
Cas de test	tester deux ajouts d'information pour l'usine LU.
Scenario	Suite au test 14, reprendre les tests 12 et 13 pour l'une des deux demandes, sans prendre en compte le resultat attendu du test 13.
Comportement attendu	Il ne doit rester plus qu'une demande qui doit se trouver dans le champs des demandes de modification.
Comportement observé	

Numéro du test	17
Fonction testée	demande de modification d'information
Cas de test	tester une demande de modification d'information pour l'usine LU. (suite)
Scenario	Cliquer sur la nouvelle demande de modification
Comportement attendu	Une page avec un champs « avant » contenant « teste 9-1 » et un champs « apres » contenant « test 16 » doit apparaitre.
Comportement observé	

Numéro du test	18
Fonction testée	demande de modification d'information
Cas de test	tester une demande de modification d'information pour l'usine LU. (suite)
Scenario	Suite au test 17, Cliquer sur valider
Comportement attendu	La page principale d'administration apparaît et l'ancienne nouvelle demande de modification n'y est plus. Il ne reste plus que la demande de modification apparue au test 15
Comportement observé	

Numéro du test	19
Fonction testée	Consultation d'archive
Cas de test	tester la consultation de l'archive liée au test 18
Scenario	Suite au test 18, se connecter en tant que chercheur, et exécuter une recherche sur l'usine LU.
Comportement attendu	Dans le champs description, deux descriptions doivent apparaître : « test 16 » comme description et « test 9-1 » comme description de type archivée.
Comportement observé	

Numéro du test	20
Fonction testée	Datation des archivations
Cas de test	tester la datation des données archivées
Scenario	Reprendre le test 19 et tous ses antécédents apres une purge des données et en réalisant la validation de la modification un jour apres l'ajout de la donnée. (la donnée test 16 est validée à la date Y, et la donnée test 9-1 est validée à la date X)
Comportement attendu	Dans le champs description, deux descriptions doivent apparaître : « test 16 » comme description et « test 9-1 » comme description de type archivée. La date de création de « test 9-1 » et la date de fin de « test 9-1 » doivent etre celles retenues X et Y
Comportement observé	

Numéro du test	21
Fonction testée	Creation de compte
Cas de test	créer le compte Administrateur
Scenario	Après avoir fait le test 5, utiliser le compte super utilisateur pour créer un compte administrateur login : admin1, mdp : mdpadmin1.
Comportement attendu	Le test 4 en se connectant sous admin1 devrait fonctionner.
Comportement observé	

Numéro du test	22
Fonction testée	Creation de compte
Cas de test	créer le compte Chercheur
Scenario	Après avoir fait le test 4, utiliser le compte administrateur sur lequel on est pour créer un compte chercheur login : chercheur1, mdp : mdpchercheur1.
Comportement attendu	Le test 2 en se connectant sous chercheur1 devrait fonctionner.
Comportement observé	

Numéro du test	23
Fonction testée	Creation de compte
Cas de test	créer le compte modérateur
Scenario	Après avoir fait le test 4, utiliser le compte administrateur sur lequel on est pour créer un compte modérateur login : modérateur1, mdp : mdpmodérateur1.
Comportement attendu	Le test 3 en se connectant sous modérateur1 devrait fonctionner.
Comportement observé	

Estimation des charges et planification

Estimation des charges

Voici l'estimation de charges selon la méthode COCOMO.

Points de fonctions

Entrées	Pondération	Coût	Sorties	Pondération	Coût
Consultation de la BDD	S	3	Résultat de la requête	S	4
Requête sur la BDD	S	3			
	Total	6		Total	4

Interrogations	Pondération	Coût	Fichiers internes	Pondération	Coût
Connexion	C	6	Site web	S	15
Administration	C	6			
	Total	12		Total	15

S = Simple,
M = Moyenne,
C = Complexe.

Dans notre projet nous n'avons pas de fichiers externes.

Total UFP = 37

Calcul de l'effort et de la durée

Paramètres utilisés :

Type organique

Nombre de lignes de code inférieur à 50000

Effort de code = $2,4 * (ISL / 1000)^{1,05}$

Durée = $2,5 * effort^{0,35}$

	Facteurs d'ajustement	Complexité (entre 0 et 5)
1	Communication des données	2
2	Traitement distribué des données	0
3	Critère de performance	2
4	Matériel hautement utilisé	0
5	Haut taux de transactions	0
6	Entrées de données en ligne	0
7	Mises à jour en ligne	0
8	Efficacité des utilisateurs	0
9	Calculs complexes	0
10	Réutilisation	3
11	Facilité d'utilisation	2
12	Facilité d'exécution	0
13	Portabilité	0
14	Maintenabilité	1
Total (Degré d'Influence)		10
Total UFP (d'après les tableaux précédents)		37
FA (= $0,65 + 0,01 * DI$)		0,75
FP (= $UFP * FA$)		27,75
ISL ($G * FP$)		1471

Pour le calcul de l'ISL, le tableau de référence est le suivant :

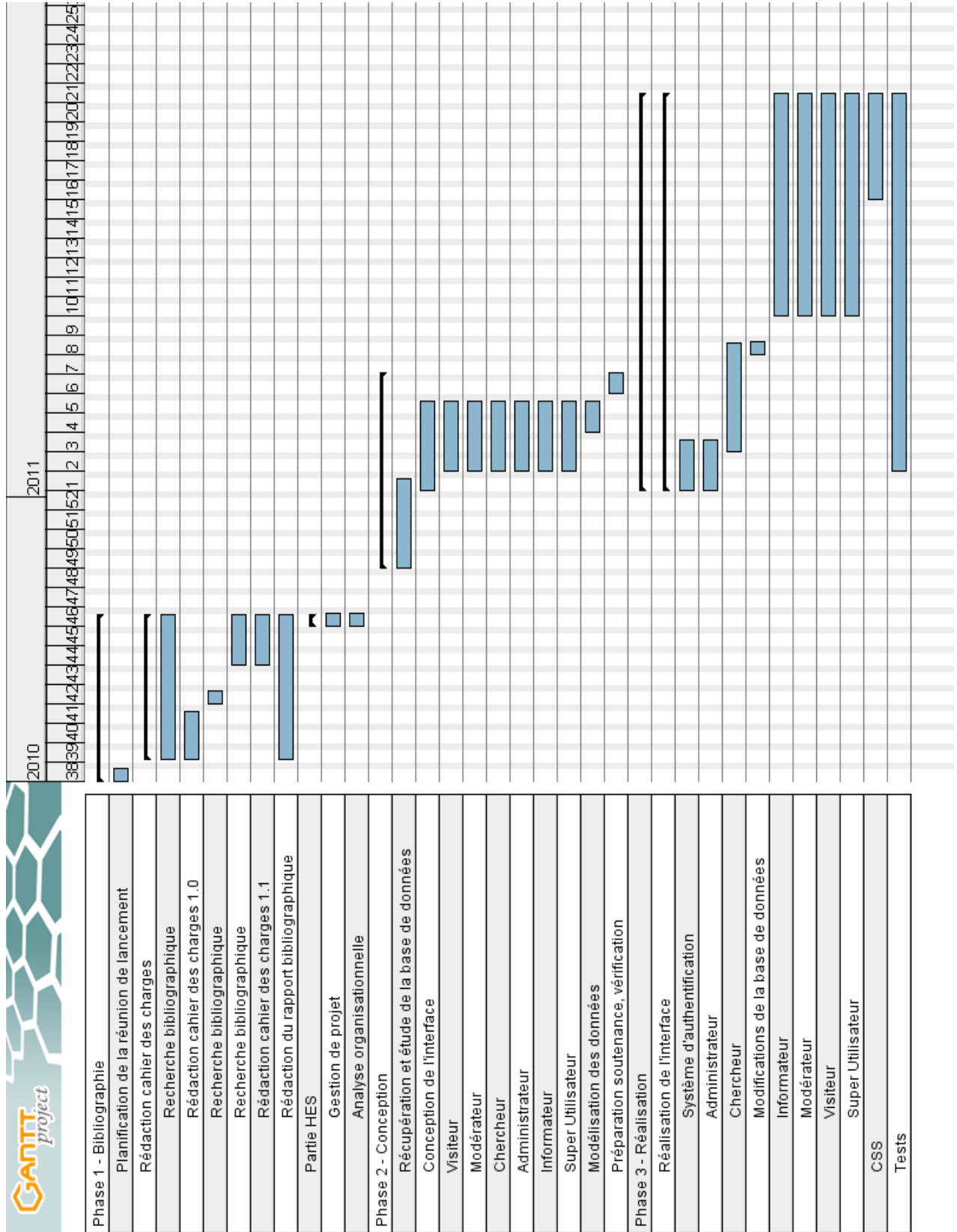
Langage	G
Assembleur	320
C	128
Cobol	105
Fortran	105
Ada	70
LISP	64
Java	53
Tableur	10

Nous avons choisi $G = 53$ car nous utilisons du PHP qui est très similaire au Java.

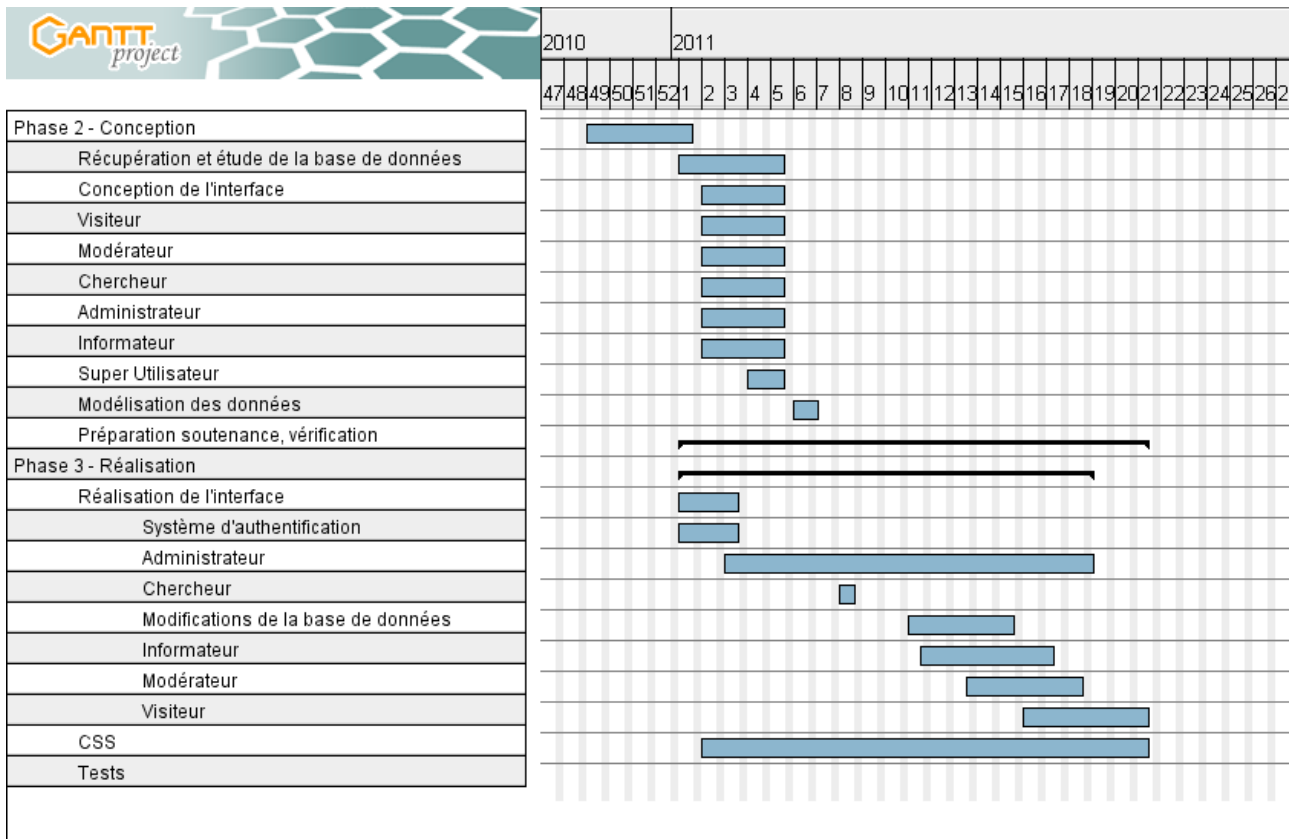
Enfin,

EFFORT	3,599	Homme mois
Durée	3,914	Mois

Planning effectif (phase bibliographique + conception)



Planning effectif phase 3

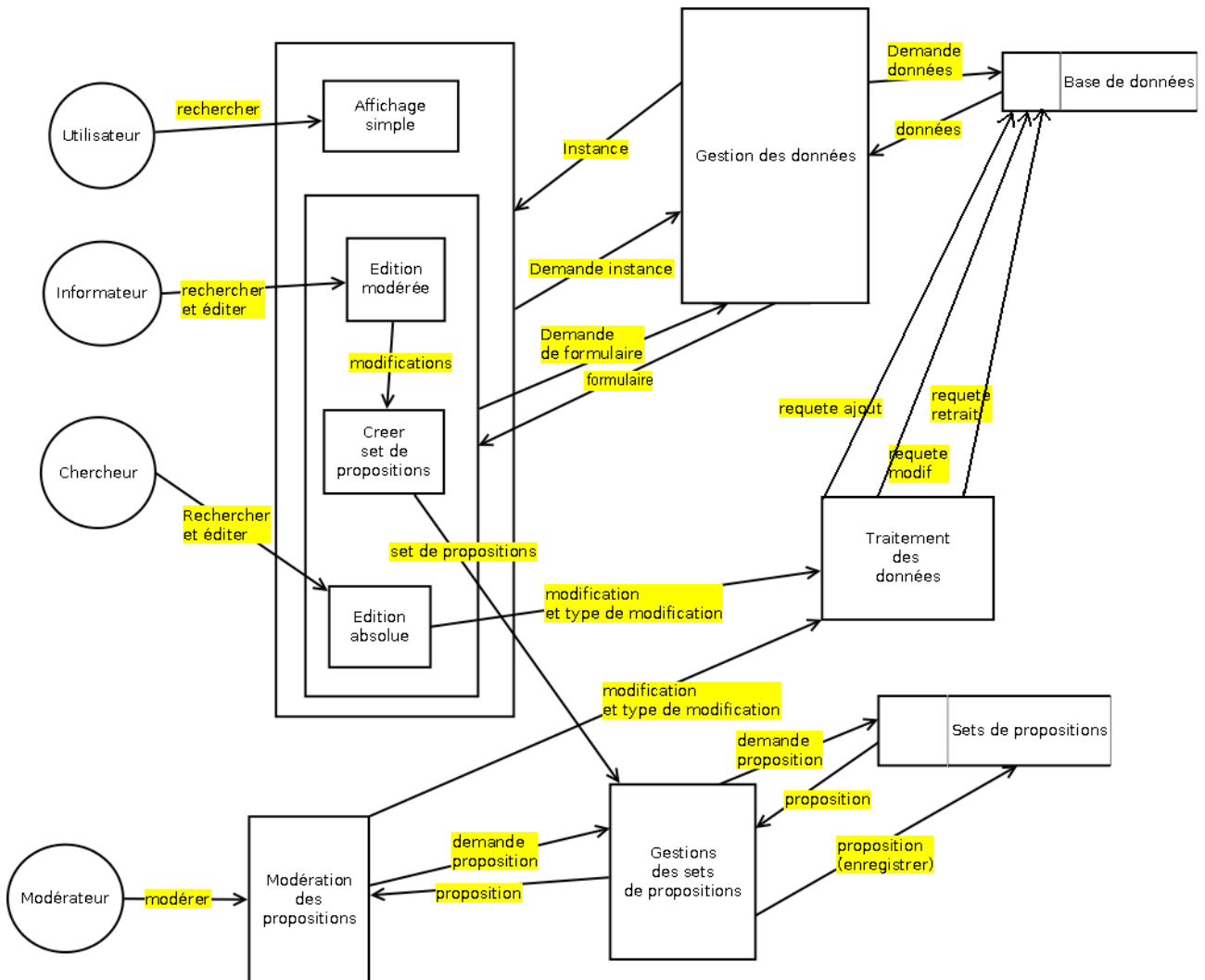


Modélisation

I - Conception

1 - L'application

a) Modélisation



b) Description

Nous nous intéressons aux quatre acteurs pouvant manipuler les données.

L'utilisateur, l'informateur et le chercheur peuvent récupérer des données via une recherche. Une instance du type de la donnée leur est retournée de façon transparente, ce qui permet l'affichage des informations et des différents formulaires propres au type de la donnée (cf partie II – Les données). La requête de sélection des données est envoyée au serveur de base de données. Les données sont

renvoyées à l'instance qui renvoie ces données rangées dans un tableau à l'interface pour qu'elle affiche ledit tableau.

L'informateur et le chercheur ont un affichage spécial étant donné qu'il permet la modification de champs. La partie application (l'instance du type de la donnée traitée) se charge de construire les formulaires de modification. Selon le type de champs, plusieurs types de formulaires existent (du texte simple à la sélection de date, l'édition de données géométriques etc...). L'interface reçoit les formulaires et les affiche.

Lorsqu'une modification est émise, c'est l'instance du type de la donnée qui se charge de créer la requête correspondantes. La requête ainsi formée est envoyée au serveur de base de donnée. La réponse du serveur (ajout/modification/retrait réussi/raté) est retournée à l'instance qui la fournit à l'interface pour que le client soit averti.

Dans le cas de l'informateur, il y a plusieurs étapes entre l'émission de modification et l'envoi de la requête de modification :

Lorsqu'il émet une modification, il émet en fait une proposition de modification. C'est une instance d'une classe qui contient comme attribut le nom de la table (ce qui permet l'instanciation coté application), l'attribut et la donnée soumise à modification, l'ancien contenu de l'attribut (cas modification/retrait) et/ou le nouveau contenu (cas ajout/modification).

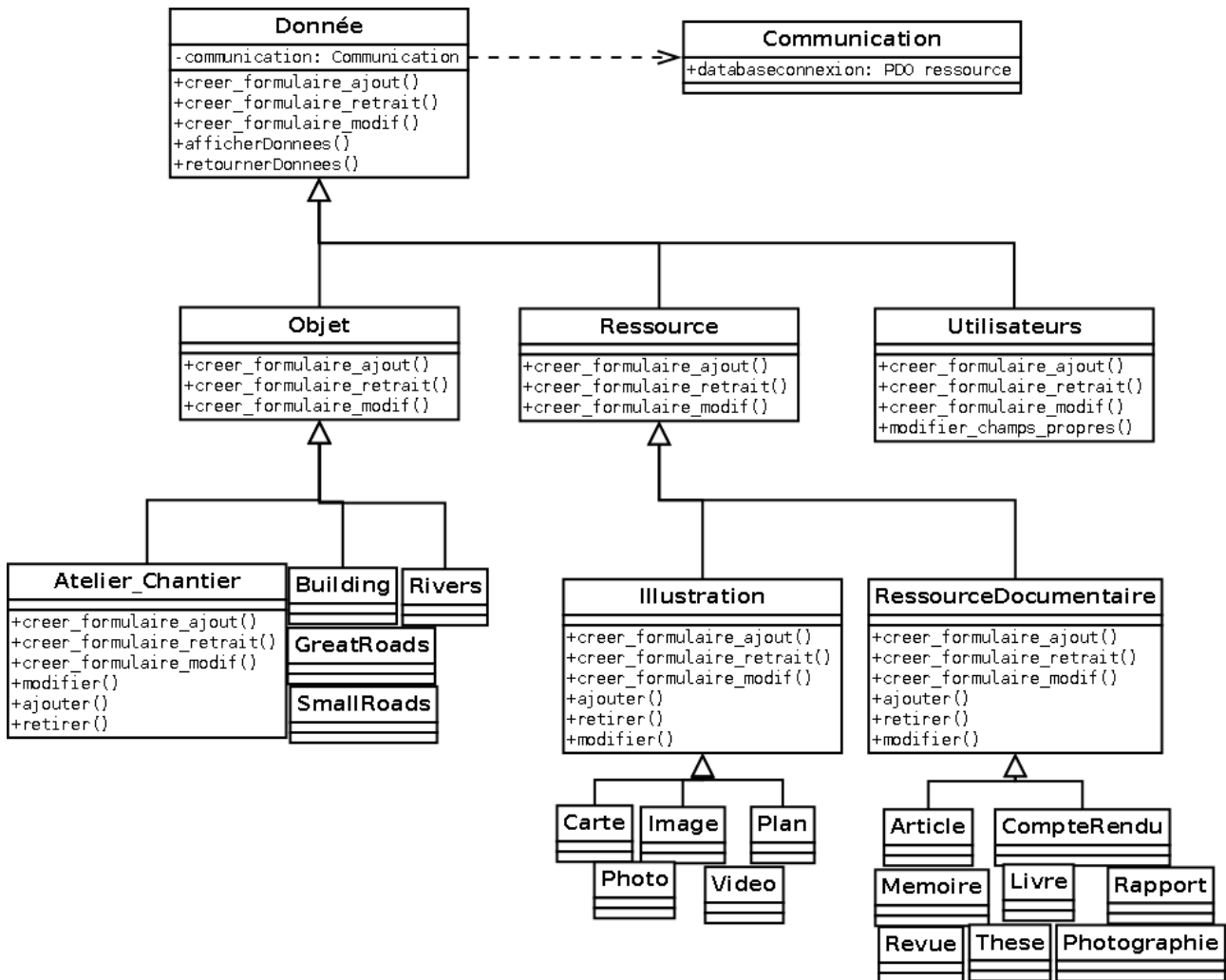
Cette proposition est envoyée depuis la partie application au serveur de base de données. Elles y sont stockées. Une tierce personne pourra les valider. Elles seront supprimées du serveur de base de données et elles seront envoyées à une instance de la table correspondante qui se charge, comme vu précédemment, de former la requête correspondante.

La tierce personne citée ci-dessus est un modérateur.

Le modérateur accède à une page qui instancie un gestionnaire des propositions. Cette instance demande au serveur de base de données la liste des propositions qui sont retournées à l'interface. Il peut alors les consulter et choisir des les valider, les refuser ou les ignorer. Dans les deux premiers cas, elles seront supprimées du serveur de base de données. Dans le cas ou elles sont validées, la suite a été expliqué dans le paragraphe précédent. Dans le troisième cas (les ignorer), le modérateur ne réalise pas d'action particulière envers ces propositions, et leur seule implication dans cette session aura été leur affichage. Elles restent intactes dans le serveur de base de données.

2 - Les données

a) Modélisation



b) Description

La base de donnée est orientée objet et organisée en arbre. Ceci implique qu'une table mère possède des attributs qu'elle fait hériter à toutes ses tables filles, qui elles-même possèdent d'autres attributs propres. Elles sont susceptibles d'avoir des filles, et de leur faire hériter la totalité de leurs attributs.

Le problème qui se soulève relève de la partie application et données : Il existent de nombreuses tables différentes à quelques détails près. Une approche irréfléchie consistera à créer toutes les requêtes et les formulaires propres à chaque table. Les grosses similitudes entre les tables entraîneront alors des redondances de code et une très mauvaises maintenance.

La solution retenue est d'organiser l'application pour qu'elle s'adapte à l'architecture des données. Ainsi, une classe par table est créée, mais toutes les classes sont organisées par un système d'héritage en suivant la hiérarchie des données.

Ainsi, toutes les classes héritent d'une classe racine « donnée ». Cette classe implémente les méthodes de création de formulaires (il s'agit ici de la partie commune à tous les formulaires) et les méthodes de récupération de données.

Dans la base de données, deux tables sont utilisées pour différencier les objets de la maquette et les documents associés à ces objets. C'est d'eux qu'héritent tous les objets et documents concrets. Donc, ces deux classes héritent de la classe données dans la partie application et détaillent à leur façon les formulaires. Elles sont capables de traiter tous les attributs communs qu'elles vont fournir à leurs filles.

La table ressource se sépare à nouveau en deux entre les illustrations et les ressources documentaires. Dans la partie application, il en va de même, ce qui permet de gérer encore plus en détail les nouveaux attributs communs.

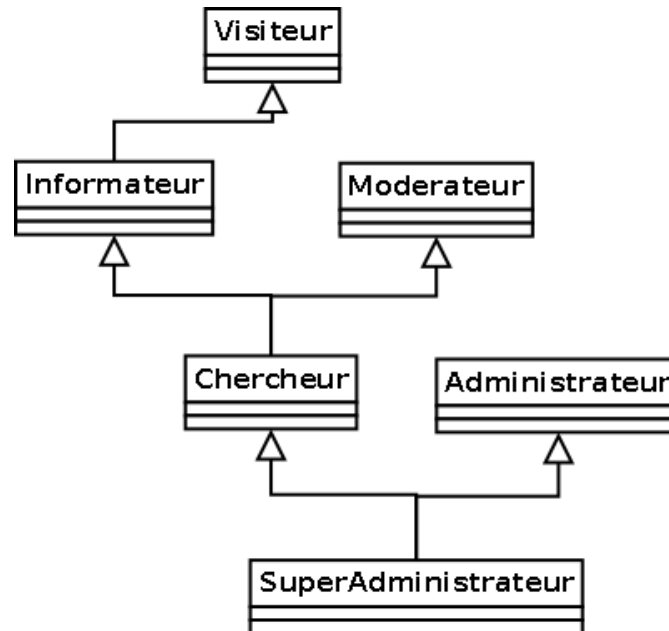
Nous arrivons alors aux classes « feuilles », qui ne proposent plus d'héritage. Chacune de ces classes est unique car les tables associées ont leur attributs propres. C'est régulièrement sur de petits détails qu'elles se différencie (par exemple la description géométrique d'une route n'est pas la même que celle d'un chantier). Elles détaillent donc leurs formulaires uniquement sur ces attributs, et pas sur le reste, qui est géré par les classes mères.

Les requêtes sont formées dans leur intégralité par les classes « feuilles ». Il y avait cette première solution, et une seconde qui consiste à former la partie commune des requêtes depuis les classes mères et former le reste par les classes feuilles. Mais une requête ne se forme pas d'une façon aussi régulière et il aurait été nécessaire de créer une organisation particulière (par exemple une grammaire syntaxique). Avec l'accumulation des données, la maintenance aurait été mauvaise avec une telle solution. De plus, les requêtes sont relativement simples, c'est pourquoi nous avons retenu la première solution.

Une particularité du côté des illustrations et des ressources documentaires est que les classes feuilles sont toutes identiques. A défaut de déterminer la différence, nous avons conçu ces deux classes comme des classes « feuilles », et leurs filles ne font rien. La réalisation sera ainsi simplifiée. Il est possible que la différence intervienne plutôt dans la partie interface (une vidéo ne s'affiche pas de la même façon qu'une carte), donc l'existence des vraies classes « feuilles » reste justifiée.

3 – L'interface

a) Conception



b) Description

La partie purement graphique de l'interface ne sera pas conçue de façon formelle car sa réalisation ne nécessite pas une mise en forme particulière et reposera, comme signalé dans le cahier des charges, sur l'affordance des éléments.

Ses mécanismes reposent fortement sur la partie application, donc cette modélisation n'est pas non plus justifiée.

Ce qui est révélateur en revanche c'est la construction par bloc de l'interface. Comme le montre le diagramme de conception ci-dessus, l'interface de recherche et d'affichage de l'utilisateur lambda est également incluse dans celle de l'informateur, du chercheur et du super administrateur. La partie de l'interface du modérateur qui consiste à valider une modification des données (se traduisant par un bouton d'émission) est couplé à la partie d'interface qui consiste à construire une modification des données et ceci est fourni au chercheur qui peut donc modifier des données sans aucune soumission à une personne tierce. Le super administrateur peut faire les actions de tout le monde. Tout ceci ne reflète pas rigoureusement les actions sur la couche application que font ces acteurs, mais reflète par contre réellement les blocs d'interface qui sont hérités (et qui n'ont, par conséquent, pas forcément la même utilité)

II – Réalisation (méthode agile)

1 – Le chercheur

L'objectif principal du projet est de fournir une interface permettant facilement de remplir la base de donnée. Pour répondre à cet objectif de façon prioritaire, nous avons réalisé un prototype de la partie application du chercheur (ce qui amène à réaliser, au passage, des prototypes pour le modérateur, l'informateur et l'utilisateur).

Avec ce prototype, nous voulons montrer la possibilité d'ajouter, retirer et modifier des données, sans pour autant gérer les effets de bord (mise à jour des dates, archivage etc ...). De même, la vérification des types n'est pas faite et il est possible de renseigner les tables sans prendre en compte certains attributs. Enfin, la gestion de la géométrie est faite mais non décidée, donc nous remplissons par défaut ces champs en indiquant un simple point dans l'espace. L'affinement sera fait en fonction des tables puisque les formulaires correspondants sont formés par les classes « feuilles ».

L'objectif de cette réalisation est principalement de montrer le bon fonctionnement de la communication avec la base de donnée et la possibilité de la traiter en ligne depuis une interface. La démonstration sera faite durant la soutenance.

2 – L'administrateur

La partie administrateur, à savoir, celle qui gère les droits des différents utilisateurs inscrits sur le site web a été réalisée. En effet, nous pensons qu'il était important de la faire en priorité afin de définir la façon dont allons gérer les différents utilisateurs au niveau de la base de données.

Pour ce faire, nous pensions tout d'abord utiliser un CMS (Système de Gestion de Contenu), mais suite à une multitude d'essais, nous nous sommes rendus compte qu'il nous serait très dur de mettre ceci en place et de l'utiliser convenablement. De plus, cela semblait inapproprié pour notre projet car bien trop volumineux et contraignant. Ainsi, après une petite entrevue avec M. Raschia, nous sommes partis vers une autre direction, à savoir : seulement prendre des morceaux de code nécessaires à nos besoins. Après quelques recherches nous avons trouvé un système de login très intéressant que nous allons présenter lors de la soutenance sur la partie conception.

Pour l'instant nous avons donc un système de login et nous pouvons gérer les différents niveaux d'utilisateurs.

Réalisation

Partie 1 – Outils

I – La base de données

La base de données utilisée reprend celle qui a été réalisée avant nous par François et Lucie. Il s'agit d'une base PostgreSQL 8.4.8 avec un module PostGIS qui permet de stocker des données de type SIG. Les modifications sur la base ont été faites avec le logiciel Pgadmin3 qui gère le PostGIS.

II – Le programme

Le programme a été codé en PHP 5 orienté objet, il génère un site web dynamique (html 5) et fonctionne sur un serveur Apache 2.2.

Aucun CMS ni IDE n'ont servi à la réalisation du programme.

Le logiciel qui a servi à faire tourner le programme en local était Wamp avec le serveur Apache cité.

Les versions ont été contrôlées avec un fichier de log.

Un module libre de gestion des authentification a été ajouté et adapté, il s'agit d'un module créé par Jpmaster77 et qu'il distribue librement. Nous l'avons modifié pour l'adapter à la base de donnée et au programme lui-même.

Partie 2 – Déroulement du logiciel

Admettons qu'une personne désireuse d'être enregistré comme étant un chercheur veuille s'inscrire. Il accède à un formulaire d'inscription :

Création de compte

Utilisateur : *	<input type="text" value="cherch"/>
Mot de passe : *	<input type="password" value="••••••"/>
e-mail : *	<input type="text" value="cherch@fdhsd.com"/>
Nom : *	<input type="text" value="nomduchercheur"/>
Prénom : *	<input type="text" value="prenomduchercheur"/>
Téléphone :	<input type="text" value="9999999"/>
Adresse :	<input type="text" value="adresseduchercheur"/>
Profession :	<input type="text"/>

* : Champs Obligatoires

[Retour à la page principale](#)

Certains champs sont obligatoires et il ne pourra pas continuer s'ils ne sont pas correctement remplis.

L'administrateur s'authentifie afin de donner son droit au chercheur

Authentification

Utilisateur :	<input type="text" value="admin"/>
Mot de passe :	<input type="password" value="•••••"/>
<input type="checkbox"/> Se souvenir de moi	<input type="button" value="Connexion"/>

Nouveau membre ? [S'enregistrer](#)

Simple visiteur ? [Entrez dans le site](#)

Copyright © 2011. Tous droits réservés.

L'administrateur peut donner des droits aux différents utilisateurs

Admin Center

..... Logged in as **admin**

Back to [\[Main Page\]](#)

Users Table Contents:

Username	Level	Email	Last Active
admin	9	admin@dsgs.com	1305295721
cherch	5	cherch@fdhsd.com	1305295655
moder	4	moder@gfshsd.com	1305295690
infor	3	infor@gdfhdf.com	1305295675

Update User Level

Username: Level:

Note : le libellé du niveau est indiqué dans le même formulaire.

Le chercheur peut alors s'authentifier

Authentification

Utilisateur :
Mot de passe :
 Se souvenir de moi

Nouveau membre ? [S'enregistrer](#)

Simple visiteur ? [Entrez dans le site](#)

Copyright © 2011. Tous droits réservés.

Son centre de contrôle lui permet d'ajouter, modifier, retirer et mettre en relation des données.

Ajouter

Objets	atelier chantier	buildings	great_roads	personne	rivers	small_roads		
Ressource documentaire	article	compterendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Retirer

Objets	atelier chantier	buildings	great_roads	personne	rivers	small_roads		
Ressource documentaire	article	compterendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Modifier

Objets	atelier chantier	buildings	great_roads	personne	rivers	small_roads		
Ressource documentaire	article	compterendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Mettre en relation

Objets	atelier chantier	buildings	great_roads	personne	rivers	small_roads		
Ressource documentaire	article	compterendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Par exemple, il va ajouter une rivière

nom_objet	<input type="text" value="riviere"/>
resume	<input type="text" value="résumé de la rivière"/>
historique	<input type="text" value="historique de la rivière"/>

Un tableau récapitulatif lui permet de visionner rapidement la base de donnée afin de voir si sa requête a été exécutée

nom_objet	resume	historique	description	adresse_postale
riviere	résumé de la rivière	historique de la rivière		

Elle l'a été.

Il veut maintenant modifier l'historique de sa rivière

Il va la chercher dans une liste (ici la liste contient uniquement la rivière qu'il a ajoutée, la base de donnée étant vide lors de la livraison)

rivers

Plusieurs attributs de l'objet sont proposés pour la modification. Il choisit l'historique

<u>nom objet</u>	<u>resume</u>	<u>historique</u>	<u>descriptio</u>
riviere	résumé de la rivière	historique de la rivière	

Un rappel sur l'historique précédent lui est fait, et il est recopié dans un champs d'édition au cas où la modification est minime. Il peut aussi bien tout écraser par un nouvel historique.

historique

historique de la rivière

lolol

Nouvel historique de la rivière

Le même récapitulatif lui permet de vérifier :

nom_objet	resume	historique	descriptio
riviere	résumé de la rivière	Nouvel historique de la rivière	

Pour finir, il met en relation sa rivière avec une image.

Mettre en relation

Objets	atelier chantier	buildings	great roads	personne	rivers	small roads		
Ressource documentaire	article	comptendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Il fait un clic sur carte et choisi la carte (précédemment créée) qu'il veut lier.

carte

carte1 ▼	envoyer
carte1	

Il accède à un formulaire dans lequel il peut décrire la relation qu'il crée.

envoyer	
CLASSER_RELATION	illustration ▼
DATE_DEBUT_RELATION	01/01/1901
DATE_DEBUT_RELATION	01/01/1909

Ici, on comprend que la carte qui représentait la rivière devait être valable de 1901 à 1909

Voici la requête qui est générée.

```
INSERT INTO relation(nom_relation,table_source,objet_source,table_dest,objet_dest) VALUES ('illustration','rivers',5,'carte',2)
```

C'est seulement l'affichage de la fiche de la rivière ou du document qui sera affecté par cette mise en relation.

L'informateur peut accéder à une page où il peut réaliser ses propositions

PAGE INFORMATEUR

Ajouter

Objets	atelier chantier	buildings	great_roads	personne	rivers	small_roads	
Ressource documentaire	article	compterendu	livre	memoire	photographie	rapport	revue these
Illustration	carte	image	photo	plan	video		

Retirer

Objets	atelier chantier	buildings	great_roads	personne	rivers	small_roads	
Ressource documentaire	article	compterendu	livre	memoire	photographie	rapport	revue these
Illustration	carte	image	photo	plan	video		

Modifier

Objets	atelier chantier	buildings	great_roads	personne	rivers	small_roads	
Ressource documentaire	article	compterendu	livre	memoire	photographie	rapport	revue these
Illustration	carte	image	photo	plan	video		

Copyright © 2011. Tous droits réservés.

Admettons que trois informateurs aient chacun leur version de la photo de l'usine LU à proposer. Voici ce qu'un des informateurs aura lors du téléchargement de photo :

photo

titre

description

date_debut_ressource

date_fin_ressource

archive_ressource

validation_ressource

mots_cles

utilisateur

objets

date_creation

date_maj

last_modified

reference_ressource

disponibilite

theme_ressource

auteur

editeur

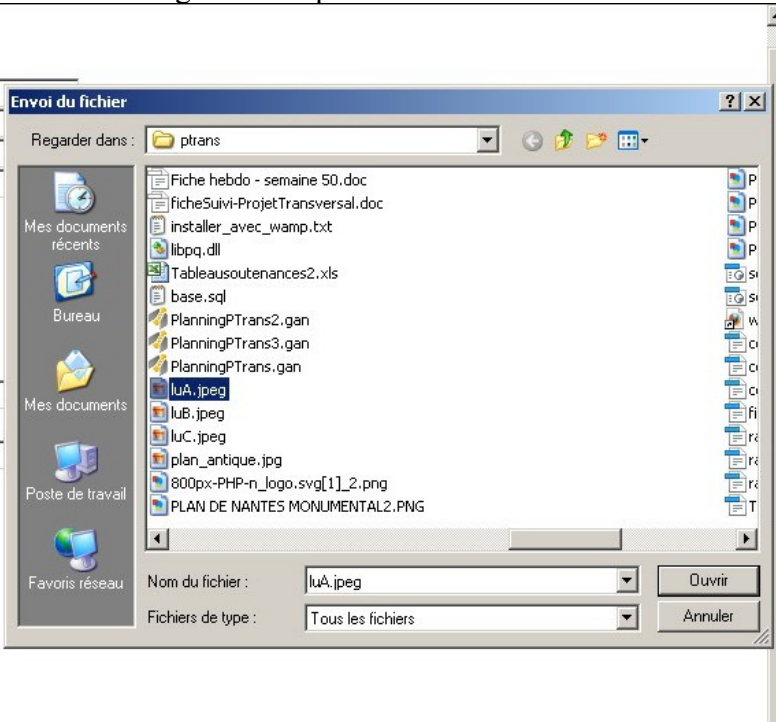
legende

couleur

type_support

Fichier :

titre



Le modérateur verra alors ces trois propositions d'ajout :

6	photo	titre_=lu+A&description=usine+LU+photo+A&date_debut_ressource=01%2F01%2F01&date_fin_ressource=01%2F01%2F01&archive_ressource=&validation_ressource=&mots_cles=&utilisateur=&objets=&date_creation=&date_maj=&last_modified=&reference_ressource=&disponibilite=&theme_ressource=&auteur=&editeur=&legende=&couleur=&type_support=&MAX_FILE_SIZE=100000000&titre=lu+A&enregistrer=ok	Details
7	photo	titre_=lu+B&description=usine+LU+photo+B&date_debut_ressource=01%2F01%2F01&date_fin_ressource=01%2F01%2F01&archive_ressource=&validation_ressource=&mots_cles=&utilisateur=&objets=&date_creation=&date_maj=&last_modified=&reference_ressource=&disponibilite=&theme_ressource=&auteur=&editeur=&legende=&couleur=&type_support=&MAX_FILE_SIZE=100000000&titre=lu+B&enregistrer=ok	Details
8	photo	titre_=lu+C&description=usine+LU+photo+C&date_debut_ressource=01%2F01%2F01&date_fin_ressource=01%2F01%2F01&archive_ressource=&validation_ressource=&mots_cles=&utilisateur=&objets=&date_creation=&date_maj=&last_modified=&reference_ressource=&disponibilite=&theme_ressource=&auteur=&editeur=&legende=&couleur=&type_support=&MAX_FILE_SIZE=100000000&titre=lu+C&enregistrer=ok	Details

Il va décider de mettre la photo A en attente, et de valider la photo C.

donnee a ajouter :

titre_	description	date_debut_ressource	date_fin_ressource	archive_ressource	validation_ressource
lu A	usine LU photo A	01/01/01	01/01/01		

[Valider](#) [Refuser](#) [Mettre en attente](#)

(ici il clique sur « mettre en attente »)

La proposition mise en attente est placée à la fin du tableau de propositions d'ajout et est grisée.

7	photo	titre_=lu+B&description=usine+LU+photo+B&date_debut_ressource=01%2F01%2F01&date_fin_ressource=01%2F01%2F01&archive_ressource=&validation_ressource=&mots_cles=&utilisateur=&objets=&date_creation=&date_maj=&last_modified=&reference_ressource=&disponibilite=&theme_ressource=&auteur=&editeur=&legende=&couleur=&type_support=&MAX_FILE_SIZE=100000000&titre=lu+B&enregistrer=ok	Details	
8	photo	titre_=lu+C&description=usine+LU+photo+C&date_debut_ressource=01%2F01%2F01&date_fin_ressource=01%2F01%2F01&archive_ressource=&validation_ressource=&mots_cles=&utilisateur=&objets=&date_creation=&date_maj=&last_modified=&reference_ressource=&disponibilite=&theme_ressource=&auteur=&editeur=&legende=&couleur=&type_support=&MAX_FILE_SIZE=100000000&titre=lu+C&enregistrer=ok	Details	
6	1	photo	titre_=lu+A&description=usine+LU+photo+A&date_debut_ressource=01%2F01%2F01&date_fin_ressource=01%2F01%2F01&archive_ressource=&validation_ressource=&mots_cles=&utilisateur=&objets=&date_creation=&date_maj=&last_modified=&reference_ressource=&disponibilite=&theme_ressource=&auteur=&editeur=&legende=&couleur=&type_support=&MAX_FILE_SIZE=100000000&titre=lu+A&enregistrer=ok	Details

Il valide ensuite la photo A

```
array
  'type' => string 'ajout' (length=5)
  'id' => string '8' (length=1)
  'nomtable' => string 'photo' (length=5)
  'donnees' => string 'titre_lu+C&description=usine+LU+photo+C&date_debut_ressource=01%2F01%2F01&date_fin_ressour
  'utilisateur' => string '' (length=0)
```

donnee a ajouter :

titre_	description	date_debut_ressource	date_fin_ressource	archive_ressource	validation_ressource
lu C	usine LU photo C	01/01/01	01/01/01		

[Valider](#) [Refuser](#) [Mettre en attente](#)

Enfin, quand un visiteur veut accéder à la photo, voici la fiche qui lui est présentée

Fiche numéro 1

Nom de l'illustration :

lu C

Description :

usine LU photo C

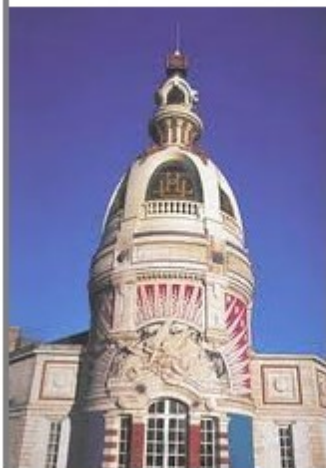
Date de création :

2001-01-01

Date de fin :

2001-01-01

Image :



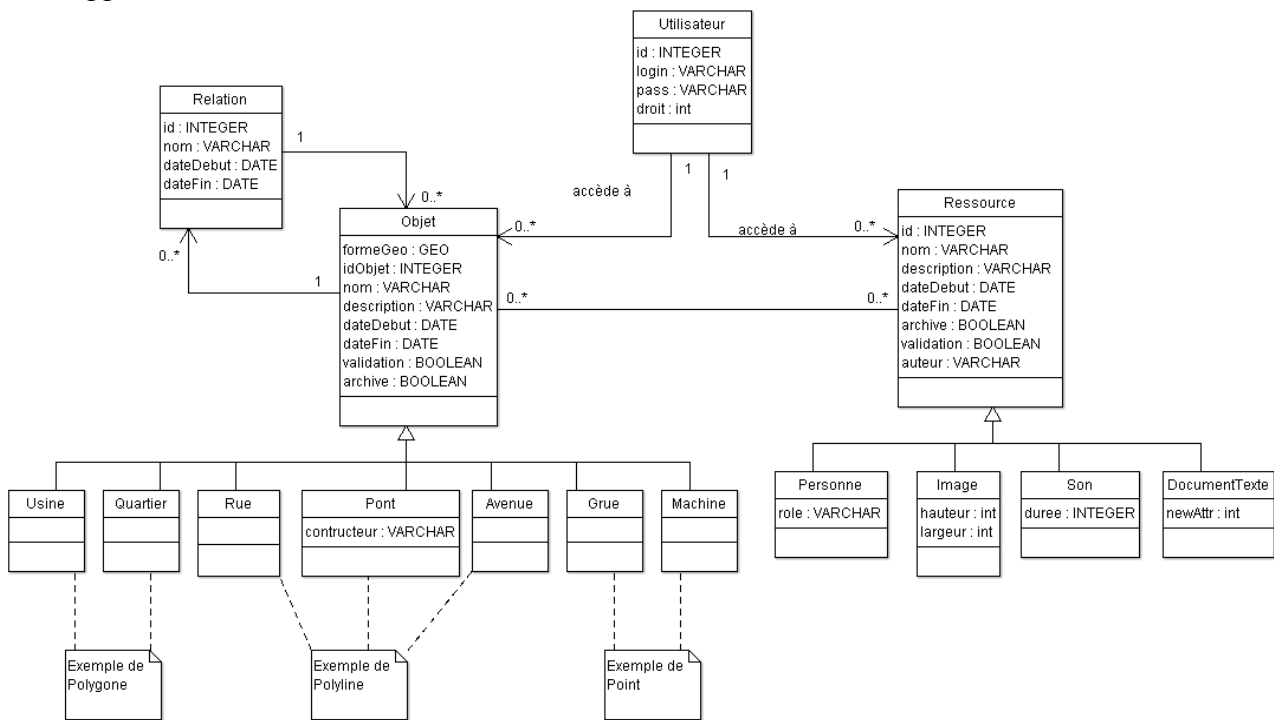
Partie 3 – Mise en œuvre

I – La base de données

La base de données nous a été fournie et nous avons créé des spécifications et une architecture en vue de l'exploiter. Pour que l'implémentation se fasse correctement, nous avons dû la retoucher, parfois de façon minime au sein des tables existantes, d'autres fois en créant de nouvelles tables. La structure de la base de données est fournie avec le projet réalisé.

a) Ajout de tables

Pour rappel, la base de données avait cette structure :



Nous y avons ajouté plusieurs tables.

En premier lieu, afin de gérer les utilisateurs, nous avons remplacé la table utilisateur par une nouvelle qui permet l'interfaçage avec le système d'authentification.

La table users garde les mêmes liaisons qu'avait la table utilisateurs.

active_users	
◊username	varchar
◊timestamp	int

Users	
◊username	varchar
◊password	varchar
◊userid	varchar
◊userlevel	int
◊email	varchar
◊timestamp	int
◊nom	text
◊prenom	text
◊adresse	text
◊telephone	varchar
◊profession	text

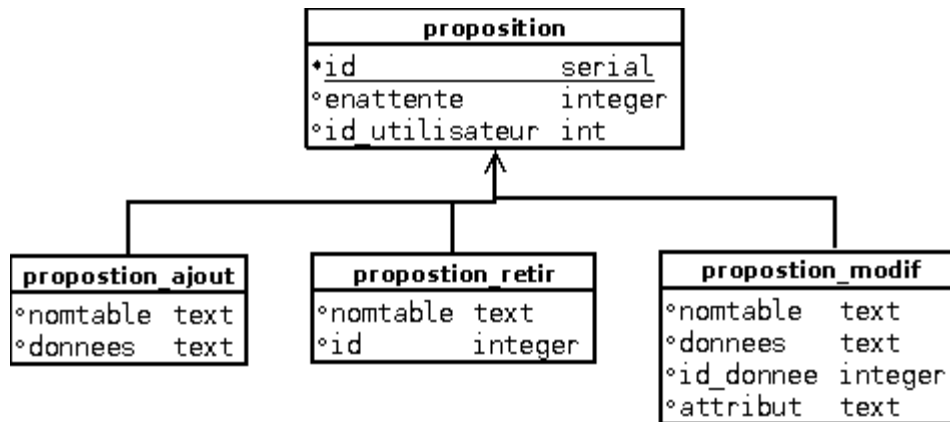
En second lieu, afin d'intégrer la conception des propositions (d'ajout, de modification, de retrait)

banned_users	
◊username	varchar
◊timestamp	int

active_guests	
◊ip	varchar
◊timestamp	int

des informateurs, de nouvelles tables ont été créées pour stocker ces propositions.

Vient alors s'ajouter à la base de données les tables suivantes :



Une proposition peut être mise en attente, c'est le comportement commun à toutes les propositions. Quand elle est supprimée, la proposition même est supprimée de la base de donnée, tout comme quand elle est validée (dans ce cas il y a bien sûr un traitement au niveau du programme en PHP qui se fait)

Une proposition peut-être une proposition d'ajout. Pour proposer un ajout, il suffit d'indiquer le nom de la table bénéficiaire de l'ajout et des données à ajouter qui peuvent être condensées en une grande chaîne. La table proposition_ajout stocke ces informations (l'attribut données est à la limite du respect de la contrainte d'atomicité des données pour la première forme normale, mais comme le programme php a besoin de la chaîne condensée telle qu'elle est, et qu'il n'est donc pas question que le serveur de base de donnée découpe les données inscrites dedans, on peut considérer que cette condensation est elle-même une unité atomique).

Pour proposer un retrait, il suffit d'un nom de table et d'un id. C'est ce qui est stocké dans proposition_retir.

Pour proposer une modification, il faut indiquer le nom de la table, l'id de la donnée à modifier, l'attribut qui sera modifié, et les données qui remplaceront les précédentes. C'est ce qui est stocké dans proposition_modif.

b) Modification de tables

Certains réglages ont dû être fait sur des tables de la base de données originale afin qu'elle s'intègre mieux à notre conception. Il s'agit pour la plupart des cas de la clef primaire que l'on a choisi d'auto-incréments. Ainsi, plus besoin de la gérer dans le programme PHP.

La table relation aussi a dû être modifiée. Elle mettait en relation deux id de deux tables (pas forcément deux tables différentes). Mais comment savoir de quelle table vient l'id stockée ? Pour résoudre ce problème, la table bénéficie simplement de deux nouveaux champs nom_table1 et

nom_table2 qui indiquent les noms des tables des ids « id1 » et « id2 ».

Encore une fois, toutes ces modifications seront faites avec le déploiement du programme sur les machines.

c) difficulté

La difficulté majeure de la manipulation de cette base de donnée est son orientation objet. La notion d'objet n'est pas complexe en soit, mais son utilisation, et donc la conception prévue par nos prédécesseurs n'était pas forcément évidente. Lors de la phase 1, nous avons longuement étudié cette structure de base de données, et c'est durant la réalisation que l'on s'est rendu compte des enjeux d'une telle approche : la factorisation du code en est une conséquence très intéressante. Le plus dur est d'exploiter toutes les subtilités conçues, et il a fallu de nombreuses version du programme avant d'avoir une architecture qui suive l'approche objet de la base et donc arriver au minimum de redondance de code.

La création de nouvelles tables nécessite toujours une retouche du code du programme PHP mais nous sommes arrivé à un stade où il est possible d'automatiser très simplement la création de nouvelles tables, à partir du moment où elle hérite d'une des deux classes mères de la base de données.

L'autre difficulté a été de se décider sur la façon de stocker les données de type multimédia (image, son, vidéo etc.).

La première approche a été de stocker les données en brut dans la base, afin que celle-ci devienne complètement un espace de stockage des données liées à la maquette. La seconde approche a été de stocker dans un dossier les données et de renseigner dans la base le nom de la donnée. C'est la seconde méthode qui a été retenue car le temps d'accès est bien plus court et la base est plus facilement exportable.

Enfin, une dernière difficulté aura été la gestion du SIG. C'est suite à plusieurs réunions qu'il a été décidé que les données géométriques et géographiques seraient stockées automatiquement dans la base de donnée par l'intermédiaire d'une autre base. Nous avons donc à reprendre les tables de notre base et modifier l'attribut du SIG comme un SIG le plus abstrait possible. L'approche objet permettant alors d'y stocker une ligne, un point, ou toute autre classe qui en hérite.

II – Le programme en PHP

Pourquoi avoir présenté la base de donnée avant le programme, qui représente pourtant le cœur du projet que nous avons à faire ?

C'est pour mettre en évidence la logique selon laquelle nous devons être certain de la structure finale de la base de donnée avant de programmer son interface. Là encore, on peut poser la question : pourquoi ?

Clemence et Pierre sont deux étudiants en master d'histoire qui travaillent en collaboration avec

nous. D'un coté nous devons leur fournir prématurément une interface afin qu'ils puissent commencer leur travail de stockage des données, et d'un autre coté ils devaient nous servir de testeurs. C'est pourquoi la base devait être finalisée d'emblée, afin que ce qu'ils ont rempli puisse être transporté à chaque mise à jour de la version « beta » du programme. La méthode agile « Scrum » se prêtait efficacement à cette pratique mais nous y reviendront ultérieurement dans ce dossier.

Nous avons déjà présenté les avantages et défauts du PHP, ainsi que son principe de fonctionnement dans le rapport de bibliographie. Nous avons également présenté l'architecture et les spécifications du programme dans le rapport de conception.

Il reste à présenter dans ce rapport de réalisation comment nous y sommes-nous pris pour proposer des prototypes fonctionnels à chaque version, quelles pratiques du php avons-nous adoptées et les difficultés rencontrées.

a) programme fonctionnel

Comme nous l'avons vu dans le rapport de conception, nous avons eu une approche originale de la question des utilisateurs : nous avons dressé un tableau présentant les actions qu'ils peuvent faire, et un arbre qui montrait les morceaux d'interface communs. Il fallait bien comprendre de cet arbre qu'il ne montrait pas les comportements communs, mais bien les blocs de programme qui pouvaient être réutilisés. Stratégiquement, nous devons donc créer intégralement l'interface du chercheur. Celle-ci avait alors des fonctions que l'on a pu réutiliser chez les autres utilisateurs : la recherche et la proposition de choix des données d'une table, l'affichage des données, les procédures d'ajout, de suppression et de modification de données (que l'on a pu découper selon les spécifications entre l'informateur et le modérateur), et bien sûr les fichiers liés à la structuration des objets utiles au programme.

La structuration des objets était importante car la base de donnée est orientée objet. Nous avons donc dû utiliser, comme prévu, l'approche objet du php (cf : spécifications en UML du rapport de conception). Nous avons pu alors factoriser les comportements communs dans la création de formulaires, dans la gestion d'ajout, de suppression et de modification, ou encore dans des fonctionnalités telles que l'affichage ou la recherche de données. Un exemple simple de cette structuration est montrée par le fragment de code suivant :

```
class ressource extends donnee
{
    function formulaire_ajout_remplir()
    {
        parent::formulaire_ajout_remplir();
        echo"

        titre <input type=text size=30 name=titre_><br>
        description <input type=text size=30 name=description><br>
        date_debut_ressource <input type=text size=30 name=date_debut_ressource
value='01/01/01'><br>
        date_fin_ressource <input type=text size=30 name=date_fin_ressource
```

```

value='01/01/01'><br>
    archive_ressource <input type=text size=30 name=archive_ressource><br>
    validation_ressource <input type=text size=30 name=validation_ressource><br>
    mots_cles <input type=text size=30 name=mots_cles><br>
    utilisateur <input type=text size=30 name=utilisateur><br>
    objets <input type=text size=30 name=objets><br>
    date_creation <input type=text size=30 name=date_creation><br>
    date_maj <input type=text size=30 name=date_maj><br>
    last_modified <input type=text size=30 name=last_modified><br>
    ";
}
(...)
class ressource_documentaire extends ressource
{
    function formulaire_ajout_remplir()
    {
        parent::formulaire_ajout_remplir();

        echo "reference_ressource<input type=text size=30 name=reference_ressource><br>
theme_ressource<input type=text size=30 name=theme_ressource><br>
auteur<input type=text size=30 name=auteur><br>
editeur<input type=text size=30 name=editeur><br>
disponibilite<input type=text size=30 name=disponibilite><br>
";
    }
}

```

Ce code montre que toutes les données de type ressource peuvent construire un formulaire d'ajout de données en proposant les champs communs à toutes les ressources. Mais une ressource documentaire proposer des champs en plus, alors la classe fille « ressource_documentaire » propose elle aussi de construire le formulaire, mais en réutilisant la méthode de sa classe mère, tout en proposant des détails supplémentaires.

La règle à suivre, selon les conclusions des réunions, était « d'abord fonctionnel ». Donc nous avons réalisé des interfaces graphiques très sommaires, avec un minimum de présentation afin que l'expérience d'utilisation ne soit pas dramatiquement détériorée, et des rapports sur les données étaient affichés à chaque page, pour que les rapports d'erreur soient les plus clairs, et pouvoir réagir le plus vite possible.

Tous ces points montrent comment le programme a pu être rapidement déployé sur les machines afin d'être tout de suite utilisé, alors que toutes les fonctionnalités prévues n'étaient pas présentes.

b) Les pratiques de programmation PHP

Comme nous l'avons vu dans le premier rapport, le PHP est un langage qui permet de nombreuses libertés. Mais cette trop grande liberté est elle-même une contrainte à la programmation, car la complexité grandissante du programme est la conséquence d'une augmentation drastique des lignes de code, s'il n'y a pas de règles et de marches à suivre précises dès le début. Les spécifications et l'architecture ne permettent pas de décrire clairement toutes ces pratiques, et nous avons dû les écrire « noir sur blanc » et les suivre à la lettre.

Nous avons ainsi suivi certaines pratiques de programmation telles que par exemple :

- Coder tous les textes affichés afin de les stocker dans des fichiers de traduction pour l'internationalisation (fr.php contient tous les textes du programme).
- Définir un en-tête commun, inclus dans toutes les pages, qui inclus lui-même les fichiers importants à inclure
- Utilisé un fichier contenant toutes les constantes du programme, servant surtout pour la connexion à la base de données et au système d'authentification
- Centraliser toute la partie concernant la communication avec la base de données (connexion, prise de requête, envoi des requêtes, retour des réponses) dans un fichier commun (database.php)
- Programmer à la fois sur windows et à la fois sur linux avec deux navigateurs différents, pour être certains que les étudiants en histoire puissent utiliser le programme (et nous verrons par la suite pourquoi le choix aura été judicieux, avec le point concernant le déploiement sur un serveur en ligne)
- Décider de fonctions de conversion des données lorsqu'elles ne sont pas atomiques (transformation d'un tableau en chaîne, et réciproquement) pour les envoyer d'une page à l'autre quand c'est nécessaire.

Outre ces quelques pratiques, il y a une architecture très fortement marquée par des fonctions. Un exemple est ce qu'il y a de mieux pour illustrer ce fait :

```
<?php

include("../include/session.php");
include("../include/dataclass.php");
include("../style/header.php");

/* la table concernee est passee en parametre.
On recupere son nom, et on cree l'instance de la classe dont le nom est celui donne en parametre.
on fait ensuite appel a la generation de formulaire propre à cet objet*/

echo $_GET['objet'];
$objet=$_GET['objet'];

// Affichage du formulaire d'ajout
$donnee = new $objet();
$donnee->formulaire_ajout();
?>
```

Ceci est l'intégralité du fichier concernant l'ajout de données. On peut voir les trois inclusions des

trois fichiers importants (dataclass.php et session.php n'ayant pas à être inclus dans toutes les pages, il faut préciser leur inclusion ici) et le contenu de la page est généré (cf les commentaires).

On notera une possibilité du php qu'est la création d'une instance dont le nom de la classe n'est pas connu (il est passé en paramètre de la page).

c) Les difficultés

La difficulté majeure dans la réalisation du programme PHP résidait dans le traçage des variables. En effet, nous avons travaillé avec une grande confiance en les spécifications initiales, mais le fonctionnement même du programme a été modifié en cours de projet, comme cela avait été prévu (par exemple avec l'adaptation du travail sans le module SolR). Nous devons donc réadapter certaines parties du code à ces changements et certaines variables ou structures ne suivaient plus. Nous avons donc dû faire preuve de rigueur afin de modifier le code de façon à ce qu'il se réadapte aux changements.

Le bon coté de cette difficulté est que nous avons acquis une certaine expérience dans la modification du code du programme. Quand M. Kerouanton nous a demandé qu'un modérateur puisse avoir la possibilité supplémentaire de mettre en attente une proposition, en plus de la valider ou la supprimer, nous avons pu rapidement implémenter cette fonctionnalité.

S'il faut soulever une autre difficulté, on peut noter la compatibilité du programme avec linux et windows en même temps, mise en péril par le stockage des fichiers multimédia. La solution qui a été retenue est d'utiliser une variable d'environnement PHP qui indique le chemin d'accès du dossier du site, qui fonctionne sur l'architecture des dossiers des deux systèmes d'exploitation.

Partie 4 – Tests

Nous avons décidé de réaliser une partie des tests de recette en présence des deux tuteurs avant la soutenance afin de décider des finalités à implémenter et être certain du fonctionnement global du programme.

Les tests de recette ponctuent la dimension organisationnelle originale de ce projet, puisque le cahier des charges est en constante évolution. Ainsi, de nombreux tests sont à revoir car leur comportement est devenu obsolète selon les nouveaux besoins.

D'autres tests de recette sont toujours d'actualité et ont été passés.

La réunion finale permettra de redéfinir ces tests obsolètes, voir les supprimer, et vérifier d'autres fonctionnalités non encore prévues à l'époque de la conception du programme.

Concernant les tests unitaires, la méthode Scrum que nous suivons nous a permis de réaliser les

tests unitaires en suivant la méthode Test Driven Development.

Pour chaque fonction, il se déroule en 5 étapes :

1 – définir le test

2 – tester. Le test échoue car rien n'est encore codé

3 – Écrire la fonction qui permet de passer le test

4 – tester. S'il ne réussit pas, revenir à l'étape 3

5 – complexifier la fonction, faire les factorisations (très important dans le cadre de notre programme) et revenir à l'étape 4.

Ce type de test est bien plus adapté à notre projet que ne l'est PHPUnit qui n'est pas forcément applicable à un projet en constante évolution.

Pour finir, le projet a été repri à tour de rôle par les membres de l'équipe, sur une seule machine à la fois, et l'intégration s'est faite "en continu". On utilise donc une méthode dite d'"Intégration continue"

Le détail de ces trois types de test sont détaillés dans le rapport de test.

Manuel d'utilisation

Ce manuel a été rédigé pour les chercheurs et modérateurs qui sont chargés de l'alimentation et de la gestion de la base de données du projet Nantes 1900.

S'authentifier

Lorsque vous vous connectez au site pour la première fois vous êtes dirigés sur cette page :

Authentification

Utilisateur :
Mot de passe :
 Se souvenir de moi

Nouveau membre ? [S'enregistrer](#)

Simple visiteur ? [Entrez dans le site](#)

Copyright © 2011. Tous droits réservés

Par défaut, il y a plusieurs utilisateurs qui existent : Chercheur, Informateur, Modérateur et admin. Le mot de passe de Chercheur/Informateur/Modérateur est nantes1900. Celui de admin est admin.

Chacun d'entre-eux correspond aux différents niveaux d'utilisateurs définis dans le cahier des charges.

Si vous souhaitez créer votre propre compte cliquez sur S'enregistrer.

Création de compte

Utilisateur : *
Mot de passe : *
e-mail : *
Nom : *
Prénom : *
Téléphone :
Adresse :
Profession :

* : Champs Obligatoires

[Retour à la page principale](#)

Copyright © 2011. Tous droits réservés.

Vous voyez qu'il faut obligatoirement remplir les champs suivants : « Utilisateur, Mot de passe, e-mail, Nom et Prénom ».

Une fois fini, vous devez cliquer sur « S'enregistrer ! », si il n'y aucune erreur, vous pouvez désormais vous authentifier avec le nom d'utilisateur que vous venez de choisir et le mot de passe.

Lorsque vous venez de vous enregistrer vous n'avez aucun droit par défaut. C'est-à-dire que vous ne pouvez faire à part consulter la base de données.

Lorsque l'administrateur vous aura mis vos droits vous aurez de nouvelles fonctionnalités qui apparaîtront sur votre page principale de connexion.

Administrer les utilisateurs et les droits

L'administrateur gère les droits des différents utilisateurs présents dans la base de données.

L'administrateur par défaut s'appelle « admin » et son mot de passe est « admin ». Il est fortement conseillé de supprimer ce compte une fois que vous avez donné des droits d'administrateur à une personne.

Voici la page d'administration :

Page d'Administration

Bonjour **admin**

[Aller à la page d'accueil](#)

Utilisateurs présents dans la base de données :

Nom d'utilisateur	Droit	Nom	Prénom	Email	Téléphone	Profession	Adresse
admin	Administrateur	Nom	Prénom	ee@ee.fr			
Chercheur	Chercheur	Nom	Prénom	ee@ee.fr			adresse
Moderateur	Moderateur	Nom	Prénom	ee@ee.fr			adresse
Informateur	Informateur	Nom	Prénom	ee@ee.fr			adresse
Test2	Visiteur	Nom	Prénom	ee@ee.fr			
Visiteur	Visiteur	Nom	Prénom	ee@ee.fr			adresse

Mettre à jour les droits

Nom d'utilisateur : Droit :

Supprimer un utilisateur

Nom d'utilisateur :

Vous pouvez y voir tous les utilisateurs existants ainsi que leur droit associé.

Vous pouvez mettre à jour les droits des utilisateurs en saisissant leur nom puis en choisissant le droit que vous voulez lui associer. Une fois fait cliquez sur Mettre à jour, et vous verrez directement le tableau se mettre à jour.

Pour supprimer un utilisateur, entrez le nom d'utilisateur que vous désirez supprimer et cliquez sur supprimer.

Ajouter des données

Pour pouvoir ajouter des données à la base de données il faut soit disposer des droits de Chercheur soit d'Informateur. Si vous n'êtes pas une de ces deux personnes dans ce cas vous ne pourrez pas accéder aux pages suivantes.

[retour](#)
Ajouter

Objets	atelier chantier	edifice	route principale	personne	riviere	route secondaire		
Ressource documentaire	article	comptereendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Retirer

Objets	atelier chantier	edifice	route principale	personne	riviere	petite route		
Ressource documentaire	article	comptereendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Modifier

Objets	atelier chantier	edifice	route principale	personne	riviere	route secondaire		
Ressource documentaire	article	comptereendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Pour ajouter une donnée il faut se référer au tableau « Ajouter » qui comporte la liste exhaustive des différentes données que l'on peut avoir dans la base de données. Prenons par exemple un objet atelier_chantier.

[retour](#)

nom_objet

resume

historique

description

adresse_postale

date_debut_objet

date_fin_objet

Copyright © 2011. Tous droits réservés.

Vous avez la liste des champs relatifs à un atelier chantier qu'il faut remplir. Il n'y a pas de vérifications des champs donc vous êtes libres de mettre ce que vous désirez. Une fois que ces champs sont remplis vous devez cliquer sur « envoyer ».

Une fois effectué, vous avez un tableau comprenant la totalité des objets du type que vous venez de créer qui s'affiche, et celui que vous venez d'enregistrer se trouve tout en bas de ce tableau. Si vous vous rendez compte que vous vous êtes trompés dans un des champs, vous pouvez faire en sorte de modifier votre donnée, pour ce faire, se référer à la section « Modifier des données ».

Attention : si vous êtes un informateur, votre donnée ne sera pas affichée dans le tableau car elle a besoin d'être acceptée par un modérateur avant cela.

Modifier des données

[retour](#)
Ajouter

Objets	atelier chantier	edifice	route principale	personne	riviere	route secondaire		
Ressource documentaire	article	comptereendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Retirer

Objets	atelier chantier	edifice	route principale	personne	riviere	petite route		
Ressource documentaire	article	comptereendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Modifier

Objets	atelier chantier	edifice	route principale	personne	riviere	route secondaire		
Ressource documentaire	article	comptereendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Pour modifier une donnée, référez vous au tableau « Modifier ». Par exemple on va modifier un atelier_chantier.

Voici la page sur laquelle vous arrivez, le menu déroulant comprend tous les différents atelier_chantiers présents dans la base de données. Lorsque vous avez fait votre choix cliquez sur « envoyer ».

[retour](#)
▼ envoyer

Vous avez le détail de la donnée, et si vous désirez modifier un champ, vous cliquez sur son nom. Par exemple si nous voulons modifier le nom :

[retour](#)

id	nom objet	resume	historique	description	adresse postale	date debut objet	date fin objet
10	Atelier_Test						

Par défaut le nom est réécrit dans le champ afin d'éviter d'envoyer un nom vide. Si vous désirez changer le nom, entrez un nouveau nom et cliquez sur « envoyer ».

[retour](#)

nom_objet
Atelier_Test

Retirer des données

[retour](#)
Ajouter

Objets	atelier chantier	edifice	route principale	personne	riviere	route secondaire		
Ressource documentaire	article	compterendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Retirer

Objets	atelier chantier	edifice	route principale	personne	riviere	petite route		
Ressource documentaire	article	compterendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Modifier

Objets	atelier chantier	edifice	route principale	personne	riviere	route secondaire		
Ressource documentaire	article	compterendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Comme pour modifier des données, vous tombez sur un menu déroulant avec les différents noms des objets présents dans la base. Pour en supprimer un, cliquez sur son nom et cliquez sur « envoyer ».

[retour](#)

Mettre en relation des données

Pourquoi mettre en relation des données ?

Lors de la visualisation d'une fiche d'une donnée, il est possible de naviguer directement vers les données relatives. Ces liens sont générés en fonction des relations créées.

Pour se faire, nous nous intéressons à la section « mettre en relation » du profile chercheur :

Mettre en relation

Objets	atelier chantier	edifice	route principale	personne	riviere	route secondaire		
Ressource documentaire	article	compterendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

Un clic gauche sur un type de donnée amène à une page de sélection de la donnée à mettre en relation. Exemple avec un édifice :



Un clic gauche sur le bouton « envoyer » ouvre une nouvelle page dans laquelle on choisit quel est le type de la donnée avec laquelle on veut mettre en relation notre sélection :

Mettre en relation

Objets	atelier chantier	edifice	route principale	personne	riviere	route secondaire		
Ressource documentaire	article	compterendu	livre	memoire	photographie	rapport	revue	these
Illustration	carte	image	photo	plan	video			

De la même façon que précédemment, un menu déroulant permet alors de choisir la donnée avec laquelle on fait la relation. Par exemple, ici, une donnée de type image qui se nomme « image_lu » :



Après un clic gauche sur le bouton « envoyer », une page apparaît permettant de spécifier la relation en choisissant son type et optionnellement sa durée par l'intermédiaire de deux dates.

CLASSER_RELATION

DATE_DEBUT_RELATION

DATE_DEBUT_RELATION

Le bouton « envoyer » valide la mise en relation. Cliquer alors sur retour pour revenir dans le menu.

Modérer des propositions

Vous noterez que l'ajout, le retrait et la modification de données peut être faite par un informateur, mais que les opérations n'aboutissent pas aux résultats d'un ajout, d'un retrait ou d'une modification. En effet, lorsqu'un informateur effectue l'un de ces traitements, celui-ci est en fait mis en suspens. C'est un modérateur qui a alors le pouvoir de valider ou non l'opération.

On parle alors de modérer des propositions.

Le modérateur dispose de trois pouvoirs : valider, refuser ou mettre en attente une proposition.

Voici comment se présente un tableau de proposition (ici des propositions d'ajout)

propositions d'ajout :

id	enattente	nomtable	donnees	utilisateur	
7		atelier_chantier	test5		Details
6	1	atelier_chantier	test5		Details
8	1	atelier_chantier	test5		Details

Chaque ligne est une proposition.

Les lignes grisées et qui sont mise en fin de tableau sont les propositions « en attente » (c'est à dire qu'elles sont considérées comme vues, mais aucune décision n'a encore été prise) Un clic sur le champs « details » permet de visualiser la proposition :

donnee a ajouter :

nom_objet	resume	historique	description	adresse_postale	date_debut_objet	date_fin_objet	enregistrer
test5							ok

[Valider](#) [Refuser](#) [Mettre en attente](#)

Un clic sur le lien « valider » permet de valider la proposition, l'opération s'effectue et la proposition est supprimée.

Un clic sur le lien « refuser » supprime la proposition

Un clic sur le lien « mettre en attente » permet de mettre en attente la proposition, l'opération n'est pas effectuée, et l'opération n'est pas supprimée.

Dossier de déploiement

Ce manuel a pour but de présenter comment faire fonctionner le site sur une machine possédant un système UNIX.

Tout d'abord il faut installer tous les paquets nécessaires :

Ouvrez votre terminal et tapez :

```
sudo apt-get install apache2 php5-pgsql postgresql-8.4 postgresql-8.4-postgis php5
```

Ensuite, il faut placer dans le répertoire `/var/www` tous les fichiers et dossiers du site.

Il faut changer les droits du répertoire « `fichiers/` », et faire en sorte que ce soit `www-data` qui possède ce dossier (sinon on ne peut pas déposer de fichiers) :

```
sudo chown -R www-data:www-data /var/www
```

Maintenant que le serveur `apache2` est prêt il faut configurer la base de données.

Pour ce faire il faut s'authentifier en tant qu'utilisateur `postgres`.

```
Sudo su postgres
```

Maintenant que l'on est connecté en tant qu'utilisateur `postgres` on peut faire des modifications sur la base de données avec la commande : `psql`

Une fois effectuée, il faut entrer ces commandes :

```
createdb nantes1900
createlang -nantes1900 plpgsql
psql -nantes1900 -f /usr/share/postgresql/8.4/contrib/postgis.sql
psql -nantes1900 -f
/usr/share/postgresql/8.4/contrib/spatial_ref_sys.sql
```

Cela crée la base de données appelée `nantes1900` avec le module `postGIS` inclut, qui permet de gérer les données SIG.

PV de Recette

Tests de recette (partie qui sera complétée lors de la réunion finale)

Les comportements observés des tests de recette ont été rédigés et réalisés par M. Kerouanton et vérifiés à nouveau par M. Hervy le Jeudi 26 Mai 2011 en présence de M. Catry et de M. Meunier. Les comportements validés concordent avec ceux des tests de recette internes.

Légende :

Validé = comportement identique à celui attendu

Validé + commentaire = comportement identique à celui attendu, mais certains détails sont à revoir

Obsolète = la fonction à tester a été otée en cours de projet.

Numéro du test	1
Fonction testée	Connexion visiteur
Cas de test	Tester si la page par défaut du site est l'accueil des visiteurs
Scenario	Entrer sur le site par la page principale
Comportement attendu	La page d'accueil des visiteurs est affichée
Comportement observé	Validé

Pour réaliser les tests 2, 3 et 4, il faut réaliser le test 5 puis le test 21, puis le test 4, et enfin les tests 22, 23 et 24.

Numéro du test	2
Fonction testée	Connexion chercheur
Cas de test	Tester si la connexion d'un chercheur fonctionner
Scenario	Depuis la page principale du site, entrer un identifiant et un mot de passe chercheur
Comportement attendu	La page d'accueil des chercheurs est affichée
Comportement observé	Validé

Numéro du test	4
Fonction testée	Connexion administrateur
Cas de test	Tester si la connexion d'un administrateur fonctionner
Scenario	Depuis la page principale du site, entrer un identifiant et un mot de passe administrateur
Comportement attendu	La page d'accueil des administrateur est affichée
Comportement observé	Validé

Numéro du test	5
Fonction testée	Connexion super utilisateur
Cas de test	Tester si la connexion d'un super utilisateur fonctionner
Scenario	Depuis la page principale du site, entrer un identifiant et un mot de passe super utilisateur
Comportement attendu	La page d'accueil des super utilisateur est affichée
Comportement observé	Obsolète

Pour tous les tests qui suivent, il faudra utiliser les comptes qui viennent d'être créé précédemment lorsque la connexion à un compte est demandée.

Numéro du test	6
Fonction testée	Récupérer des informations
Cas de test	Tester la fonction de requête et de récupération d'information pour l'usine LU
Scenario	Depuis la page principale du site, entrer le nom d'un bâtiment déjà existant dans la base de donnée (ex : L.U.) et valider la recherche
Comportement attendu	La page de description de l'élément correspondant (ex : L.U.) est affichée
Comportement observé	Validé

Numéro du test	7
Fonction testée	Obtenir plus d'informations
Cas de test	Tester la fonction d'extension sémantique
Scenario	Depuis une page de description, cliquer sur un lien au choix afin d'obtenir plus d'informations.
Comportement attendu	Plus d'informations doivent être affichées
Comportement observé	Validé

Numéro du test	8
Fonction testée	Visionner une video
Cas de test	tester le visionnage de video pour l'usine LU
Scenario	Se loguer en visiteur. Faire une recherche sur l'usine LU. Choisir une video et cliquer dessus.
Comportement attendu	La video se met en marche.
Comportement observé	Validé

Numéro du test	9
Fonction testée	Accéder à l'ajout d'information
Cas de test	tester l'ajout d'information sur l'usine LU
Scenario	Se loguer en informateur. Chercher l'usine LU, sur la page d'information liée à l'usine LU, cliquer sur le champ description.
Comportement attendu	La page proposant un éditeur de texte apparait.
Comportement observé	Validé

Numéro du test	10
Fonction testée	Proposer l'ajout d'information
Cas de test	tester l'ajout d'information sur l'usine LU (suite)
Scenario	Suite au test 9, inscrire « test 10 » dans le champs et cliquer sur valider.
Comportement attendu	la page d'information liée à l'usine LU apparaît avec un message indiquant que l'information a été proposée à l'ajout.
Comportement observé	Validé

Numéro du test	11
Fonction testée	Confirmer l'ajout d'information
Cas de test	tester l'ajout d'information sur l'usine LU (suite)
Scenario	Suite au test 10, se loguer en tant qu'administrateur.
Comportement attendu	Sur la page principale d'administrateur, dans le champs des requêtes d'ajout, une requete d'ajout pour l'attribut description de l'usine LU doit apparaitre.
Comportement observé	Validé mais l'idéal serait d'être prévenu des modifs proposées

Numéro du test	12
Fonction testée	Confirmer l'ajout d'information
Cas de test	tester l'ajout d'information sur l'usine LU (suite)
Scenario	Suite au test 11, cliquer sur la demande apparue
Comportement attendu	Le texte « test 10 » doit apparaître dans le champs lié à l'information en demande.
Comportement observé	Validé

Numéro du test	13
Fonction testée	Confirmer l'ajout d'information
Cas de test	tester l'ajout d'information sur l'usine LU (suite)
Scenario	Suite au test 12, cliquer sur valider.
Comportement attendu	La page principale d'administration doit apparaitre.
Comportement observé	Validé

Numéro du test	14
Fonction testée	multiples demande d'ajout d'information
Cas de test	tester deux ajouts d'information pour l'usine LU.
Scenario	Vider les données de test. Reprendre les tests 9 et 10 deux fois en proposant comme information « test 9 - 1 » et « test 9 - 2 ». Puis reprendre le test 11
Comportement attendu	Deux demandes d'ajout doivent etre affichées.
Comportement observé	Validé

Numéro du test	15
Fonction testée	multiples demande d'ajout d'information
Cas de test	tester deux ajouts d'information pour l'usine LU.
Scenario	Suite au test 14, reprendre les tests 12 et 13 pour l'une des deux demandes, sans prendre en compte le resultat attendu du test 13.
Comportement attendu	Il ne doit rester plus qu'une demande qui doit se trouver dans le champs des demandes de modification.
Comportement observé	Validé, avec décision de garder la possibilité de ne pas charger l'image.

Numéro du test	16
Fonction testée	demande de modification d'information
Cas de test	tester une demande de modification d'information pour l'usine LU.
Scenario	Sans vider les données de test, reprendre les tests 9 à 12 en choisissant la modification. (ecrire « test 16 » avant de soumettre la modification)
Comportement attendu	Une demande de modification supplémentaire est dans la liste de demande de modification.
Comportement observé	Validé

Numéro du test	17
Fonction testée	demande de modification d'information
Cas de test	tester une demande de modification d'information pour l'usine LU. (suite)
Scenario	Cliquer sur la nouvelle demande de modification
Comportement attendu	Une page avec un champs « avant » contenant « teste 9-1 » et un champs « apres » contenant « test 16 » doit apparaitre.
Comportement observé	Validé

Numéro du test	18
Fonction testée	demande de modification d'information
Cas de test	tester une demande de modification d'information pour l'usine LU. (suite)
Scenario	Suite au test 17, Cliquer sur valider
Comportement attendu	La page principale d'administration apparaît et l'ancienne nouvelle demande de modification n'y est plus. Il ne reste plus que la demande de modification apparue au test 15
Comportement observé	Validé

Numéro du test	19
Fonction testée	Consultation d'archive
Cas de test	tester la consultation de l'archive liée au test 18
Scenario	Suite au test 18, se connecter en tant que chercheur, et exécuter une recherche sur l'usine LU.
Comportement attendu	Dans le champs description, deux descriptions doivent apparaître : « test 16 » comme description et « test 9-1 » comme description de type archivée.
Comportement observé	Validé mais le modérateur devrait avoir se pouvoir également.

Numéro du test	20
Fonction testée	Datation des archivations
Cas de test	tester la datation des données archivées
Scenario	Reprendre le test 19 et tous ses antécédents après une purge des données et en réalisant la validation de la modification un jour après l'ajout de la donnée. (la donnée test 16 est validée à la date Y, et la donnée test 9-1 est validée à la date X)
Comportement attendu	Dans le champs description, deux descriptions doivent apparaître : « test 16 » comme description et « test 9-1 » comme description de type archivée. La date de création de « test 9-1 » et la date de fin de « test 9-1 » doivent être celles retenues X et Y
Comportement observé	Validé

Numéro du test	21
Fonction testée	Creation de compte
Cas de test	créer le compte Administrateur
Scenario	Après avoir fait le test 5, utiliser le compte super utilisateur pour créer un compte administrateur login : admin1, mdp : mdpadmin1.
Comportement attendu	Le test 4 en se connectant sous admin1 devrait fonctionner.
Comportement observé	Obsolète (admin créé par défaut)

Numéro du test	22
Fonction testée	Creation de compte
Cas de test	créer le compte Chercheur
Scenario	Après avoir fait le test 4, utiliser le compte administrateur sur lequel on est pour créer un compte chercheur login : chercheur1, mdp : mdpchercheur1.
Comportement attendu	Le test 2 en se connectant sous chercheur1 devrait fonctionner.
Comportement observé	Validé

Numéro du test	23
Fonction testée	Creation de compte
Cas de test	créer le compte modérateur
Scenario	Après avoir fait le test 4, utiliser le compte administrateur sur lequel on est pour créer un compte modérateur login : modérateur1, mdp : mdpmodérateur1.
Comportement attendu	Le test 3 en se connectant sous modérateur1 devrait fonctionner.
Comportement observé	Validé

Tests unitaires

La méthode Scrum que nous suivons nous a permis de réaliser les tests unitaires en suivant la méthode Test Driven Development.

Pour chaque fonction, il se déroule en 5 étapes :

- 1 – définir le test
- 2 – tester. Le test échoue car rien n'est encore codé
- 3 – Ecrire la fonction qui permet de passer le test
- 4 – tester. S'il ne réussit pas, revenir à l'étape 3
- 5 – complexifier la fonction, faire les factorisations (très important dans le cadre de notre programme) et revenir à l'étape 4.

On notera que ce test aura été écrit pour les fonctions suivantes :

ajout
retrait
modifications
recherche
affichage_donnée_modifiable
valider
refuser
mise_attente
afficher_fichier

les tests n'ont pas été écrits pour la gestion des utilisateurs car nous avons réutilisé un programme qui fonctionnait, son adaptation ne concernant que l'accès à diverses pages.

Voici le déroulement et les résultats de chacun des tests :

ajout

Faire une fonction qui ajoute un atelier chantier

clic gauche sur ajouter

page blanche

échec

la fonction crée une requête qui ajoute un atelier chantier sans attribut

réussi

Proposer des attributs

la fonction utilise les résultats d'un formulaire dédié

réussi

Générer le formulaire en fonction de l'atelier chantier

Le formulaire propose de remplir des attributs qu'il cherche dans la base de données

échec du test : gestion du SIG

Remplissage provisoire des données SIG

réussi

La fonction doit s'étendre aux autres données de type objet

La fonction utilise la classe Objet pour générer un formulaire et une requête qui permettent d'ajouter des objets (cf fonction recherche)

réussi

La fonction doit s'étendre aux autres données de type donnée

La fonction n'a pas eu besoin d'être adaptée. Elle doit fonctionner pour les données de type document

réussi

La fonction doit remplir elle-même les attributs qui doivent être remplis automatiquement (date d'ajout, utilisateur et archivage)

réussi

retrait

Faire une fonction qui retire un atelier chantier

clic gauche sur retirer

page blanche

échec

la fonction crée une requête qui retire un atelier chantier d'un id prédéfini

réussi

Proposer l'id

la fonction utilise les résultats d'un formulaire qui propose les identifiants des ateliers chantier (cf fonction recherche)

réussi

Afficher les noms des ateliers chantier

Le formulaire n'affiche plus les id mais les noms

réussi

La fonction doit s'étendre aux autres données

Toutes les données ont un champ id. La sélection de la table dans laquelle faire la suppression est passée en paramètre.

réussi

modifications

Faire une fonction qui permet de modifier le nom d'un atelier chantier

clic gauche sur modifier

page blanche

échec

Ajout d'un formulaire qui affiche la liste des ateliers chantiers (réutilisation du formulaire de retrait)

clic gauche sur l'atelier chantier

page blanche

échec

Ajout d'une page qui prend en paramètre l'id de l'atelier chantier et qui liste ses attributs avec un lien sur l'attribut nom

clic gauche sur le lien du nom

page blanche

échec

Ajout d'un formulaire de modification du nom : champs de texte éditable

Ajout de la requête qui modifie le nom
réussi

Proposer de modifier n'importe quel attribut

Tous les attributs ont un lien
clic gauche sur un autre attribut que le nom
proposition de modification du nom
échec

Modification du formulaire d'édition : il prend en paramètre l'attribut à modifier
réussi

Proposer de modifier n'importe quelle donnée

Rien à modifier : la liste des attributs est générée en fonction de la donnée, et la sélection est réutilisée (cf affichage-formulaire de retrait)
réussi

recherche

Proposer de chercher un atelier chantier dans la liste des ateliers chantier

clic gauche sur recherche

page blanche

échec

création d'un formulaire qui affiche la liste des ateliers chantiers

réussi

étendre la recherche à tout type de donnée

Passage en paramètre du type de donnée dans la fonction

réussi

affichage_donnée_modifiable

Permettre de modifier les attributs d'une donnée

clic gauche sur modifier

affichage d'un formulaire qui ajoute des ateliers chantier (ce test a été réalisé durant le test de la fonction de modification)

En fonction de la donnée, la fonction retourne la liste des attributs qui sont alors cliquables et ouvrent une page avec l'attribut en paramètre

réussi

Ne permettre de modifier que des attributs définis d'une donnée

En fonction de la donnée et de la liste des attributs modifiables définis dans la structure, la fonction génère les liens cliquables des attributs modifiables

réussi

valider

Permettre de valider une proposition

après clic sur "détail" d'une proposition d'ajout, clic sur valider

échec

création d'une fonction qui forme une requête d'ajout en fonction des données stockées dans la proposition

réussi

Supprimer la proposition

Création d'une requête qui supprime la proposition passée en paramètre
réussi

refuser

Permettre de refuser une proposition

après clic sur "détail" d'une proposition de retrait, clic sur valider
échec

Création d'une requête qui supprime la proposition passée en paramètre
réussi

mise_attente

Permettre de mettre en attente une proposition

après clic sur "détail" d'une proposition d'ajout, clic sur valider
échec

création d'une fonction qui forme une requête qui modifie l'attribut "en_attente" de la proposition
réussi

afficher_fichier

Afficher l'image d'un document

affichage d'un document : on ne voit que le nom de l'image
échec

modification de la fonction d'affichage pour qu'elle affiche une image
réussi

Afficher le media d'un document

adaptation de la fonction en prenant en paramètre le type de donnée qu'elle traite, elle crée les balises d'affichage de video, d'image ou de lecture du son en fonction de l'attribut correspondant
réussi.

Tests d'intégration

Le projet a été repri à tour de rôle par les membres de l'équipe, sur une seule machine à la fois, et l'intégration s'est faite "en continu". On utilise donc une méthode dite d'"Intégration continue"

Pour se faire, l'intégralité du programme a été transmise systématiquement à chaque tour de rôle, le donneur devenant le chef de projet provisoire.

Chaque version stable du projet (environ toutes les deux semaines) a été fournie à des collaborateurs qui ont un rôle à la fois d'utilisateur et des testeur.

Un fichier log permet de faire en sorte que chacun sache ce qui a été fait et ce qui va être fait.

Le clone du code s'est fait sur un serveur FTP afin de garantir des versions de secours.

Pas de compilation à exécuter car le PHP est un langage interprété. La vérification s'est faite sur les messages d'erreur et sur des jeux de tests d'ajout/modification/retrait/mise en relation/proposition de données "typiques", réalisés à chaque tour de rôle.

Manuel d'installation

Ce manuel a pour but de présenter comment faire fonctionner le site sur une machine possédant un système UNIX.

Tout d'abord il faut installer tous les paquets nécessaires :

Ouvrez votre terminal et tapez :

```
sudo apt-get install apache2 php5-pgsql postgresql-8.4 postgresql-8.4-postgis php5
```

Ensuite, il faut placer dans le répertoire `/var/www` tous les fichiers et dossiers du site.

Il faut changer les droits du répertoire « `fichiers/` », et faire en sorte que ce soit `www-data` qui possède ce dossier (sinon on ne peut pas déposer de fichiers) :

```
sudo chown -R www-data:www-data /var/www
```

Maintenant que le serveur `apache2` est prêt il faut configurer la base de données.

Pour ce faire il faut s'authentifier en tant qu'utilisateur `postgres`.

```
Sudo su postgres
```

Maintenant que l'on est connecté en tant qu'utilisateur `postgres` on peut faire des modifications sur la base de données avec la commande : `psql`

Une fois effectuée, il faut entrer ces commandes :

```
createdb nantes1900
createlang -nantes1900 plpgsql
psql -nantes1900 -f /usr/share/postgresql/8.4/contrib/postgis.sql
psql -nantes1900 -f
/usr/share/postgresql/8.4/contrib/spatial_ref_sys.sql
```

Cela crée la base de données appelée `nantes1900` avec le module `postGIS` inclus, qui permet de gérer les données SIG.

Dimension organisationnelle, commerciale ou environnementale.

Fiche :

Problématique : Pluridisciplinarité et empirisme : quels sont les avantages et inconvénients de ces concepts novateurs ?

Le projet Nantes 1900 a pour particularité d'être interdisciplinaires entre les sciences de l'ingénieur et les sciences humaines et sociales. En effet, d'une part il a pour but de valoriser la maquette du port de Nantes d'un point de vue historique, culturel et muséographique, et d'autre part, il doit faire appel aux techniques des sciences de l'ingénieur pour mettre techniquement en oeuvre cette valorisation. Ainsi, plusieurs équipes pluridisciplinaires issues de l'Ecole Polytechnique de l'Université de Nantes et de l'Ecole Centrale de Nantes s'organisent autour de ce projet.

Le projet renouvelle ainsi le lien entre l'histoire, la technique et la culture, en étant réalisé essentiellement par des étudiants, des stagiaires, leurs tuteurs, les enseignants-chercheurs et les permanents. Mais cela constitue aussi certaines difficultés originales.

Le projet est soumis à des contraintes temporelles. En effet, pour limiter les coûts du projet, il a été fait appel à des étudiants. Ceux-ci sont présents un temps limité et ne travaillent pas à temps plein, ce qui génère des problèmes de cohésion et de passation du travail entre les équipes. De plus, les écoles attendent des étudiants un travail différent de celui qui leur est demandé et ils doivent agencer entre elles les différentes attentes. Ce sont ainsi les contraintes associées à la pluridisciplinarité et qui ralentissent la réalisation du projet, mais qui en garantissent la créativité.

Cette créativité est également maintenue par l'aspect empirique du projet.

Le projet n'obéit pas à un cahier des charges spécifique et détaillé, mais à des lignes directrices qui sont revues tout au long de sa réalisation. En effet, l'équipe de direction a décrit la demande globale du client et fixe au fur et à mesure les objectifs généraux. De plus, le château des Ducs se veut à la pointe de la technologie muséographique et n'a donc pas imposé de cahier des charges au comité de direction du projet afin de laisser libre cours à la créativité et à l'utilisation des nouvelles technologies qui sont susceptibles d'apparaître durant sa réalisation. Enfin, il réside une certaine rédaction de cahier des charges qui s'effectue au sein des équipes responsables de la réalisation de sous-modules du projet, ce qui permet de garder une direction scientifique précise et rigoureuse. Cette description montre ainsi le côté empirique de la réalisation du projet qui garantit son côté original et créatif tout en restant lié à la rigueur scientifique.

Mais ce côté libre du projet admet certaines contraintes liées au changement d'objectif en cours de projet. En effet, en l'absence de cahier des charges, le rôle de chaque équipe change tout au long du projet et avec l'avancement des travaux, ce qui est source de réunions afin de rétablir la cohésion entre les équipes. Ainsi le projet peut parfois être ralenti et désorganisé, mais le comité de direction assure toujours le rassemblement des travaux et leur bon agencement.

En conclusion, le projet Nantes 1900 a pour originalité d'être pluridisciplinaires en rassemblant des étudiants de différents secteurs et en les faisant lier science de l'ingénieur et sciences humaines, et laisse libre cours à leur imagination et leur créativité pour réaliser et améliorer les objectifs globaux prévus. Cela a pourtant certaines limites comme les problèmes d'organisation entre les équipes et les contraintes temporelles liées au temps alloué aux étudiants pour travailler sur ce projet. Mais la centralisation du travail au sein de l'équipe de direction maintient le bon agencement du projet, qui se démarque particulièrement des projets d'entreprise qui est en bonne voie pour arriver au résultat espéré comme en témoigne l'état de l'avancée des travaux après ces deux dernières années.

Projet transversal
Dimension organisationnelle
Olivier Catry
Vincent Meunier

Enseignant : Jacques Moreau

I - Introduction

L'organisation globale du projet Nantes 1900 est particulière. C'est la remarque que l'on peut se faire lorsque l'on constate la construction de ses équipes, la gestion du temps et la communication interne : De nombreuses équipes pluridisciplinaires travaillent à temps partiel, en étroite collaboration, sur un projet dont elles ne peuvent pas voir à la fois le début et la fin et qui n'a pas d'objectif défini de façon concise. Ce rapport a pour but de montrer comment cette organisation originale basée sur la pluridisciplinarité et l'empirisme permet une modernisation et une coordination efficace, toutefois menacés par des difficultés et des effets de bord. Pour poser la problématique, interrogeons-nous donc sur la question suivante : Pluridisciplinarité et empirisme, quels sont les avantages et inconvénients de ces concepts novateurs ? Notons que le travail que nous avons fait est donné en exemple à plusieurs reprises dans ce rapport afin d'en illustrer ces propos qui présentent en premier la pluridisciplinarité du projet, les difficultés que celle-ci soulève et enfin ses bénéfices en terme de créativité.

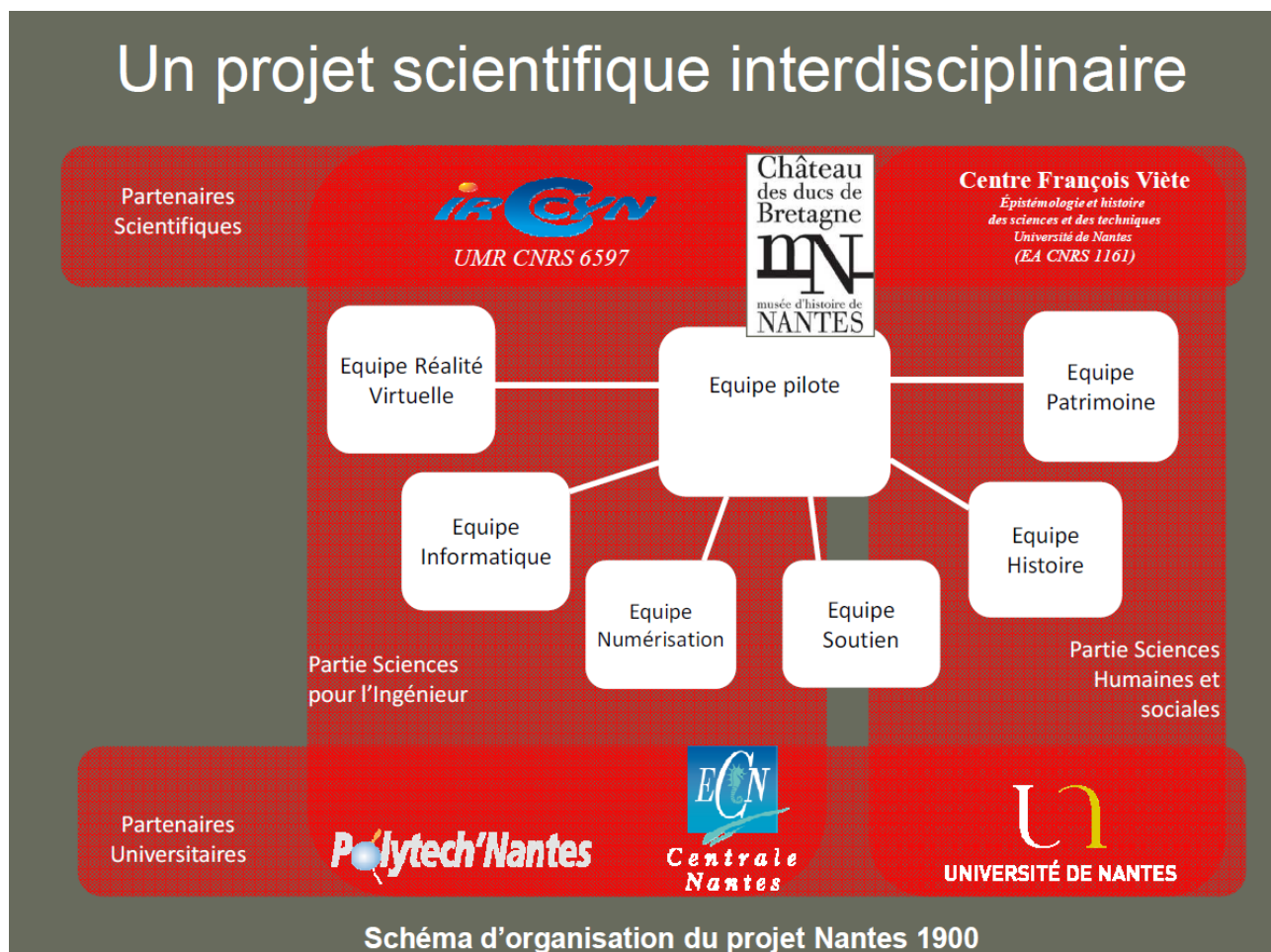
II - Pluridisciplinarité

Le projet Nantes 1900 a pour particularité d'être interdisciplinaires entre les sciences de l'ingénieur et les sciences humaines et sociales. En effet, d'une part il a pour but de valoriser la maquette du port de Nantes d'un point de vue historique, culturel et muséographique, et d'autre part, il doit faire appel aux techniques des sciences de l'ingénieur pour mettre techniquement en œuvre cette valorisation. Ainsi, plusieurs équipes pluridisciplinaires issues de l'École Polytechnique de l'Université de Nantes et de l'École Centrale de Nantes s'organisent autour de ce projet.

Il faut aussi ajouter à cela le Château des Ducs : la direction du projet a été confiée conjointement à Christophe Courtin (responsable du département multimédia du musée d'histoire de Nantes), Bertrand Guillet (directeur adjoint du musée d'histoire de Nantes), Jean-Louis Kerouanton (enseignant-chercheur au Centre François Viète, à l'université de Nantes et l'École Polytech'Nantes) et Florent Laroche (enseignant-chercheur à l'École Centrale de Nantes, Département IPSI et Laboratoire IRCCyN (Institut de Recherche en Communication et en Cybernétique de Nantes)). Cette équipe constitue le comité scientifique de pilotage et cautionne les choix et les orientations scientifiques du programme de recherche développé au sein de ce projet.

Nous pouvons donc nous rendre compte de la très forte interdisciplinarité entre les sciences pour l'ingénieur et les sciences humaines. L'intérêt d'un tel projet pluridisciplinaire et transversal est de fédérer de nombreuses compétences faisant appel aux écoles polytechnique et Centrale de Nantes, et de l'Université. Une autre spécificité de ce projet pédagogique réside dans sa composition. Les membres participant au projet sont principalement des étudiants-stagiaires, leurs enseignants-chercheurs et leurs tuteurs, tout comme quelques permanents (notamment pour la partie informatique, pour la réalisation et l'installation du prototype opérationnel, et pour la coordination générale du projet).

Les étudiants-stagiaires participants au projet sont principalement issus de l'université de Nantes et des Ecoles Centrale et Polytechnique de Nantes. Les membres sont répartis dans six domaines particuliers qui constituent les équipes mises en place dès le commencement du projet en 2008. On constate un partage du travail : les élèves issus des écoles d'ingénieurs se sont chargés de la maquette virtuelle (numérisation son pots-traitement tout comme la modélisation 3D) mais aussi de la partie informatique avec la conception de la base de données, des sites internet, et du dispositif de réalité virtuelle associé au prototype. Les étudiants des masters d'histoire et de valorisation du patrimoine se sont consacrés à l'étude de la maquette physique. Ils ont ainsi effectué des recherches documentaires et ont mené une réflexion muséographique. Ils travaillent en étroite collaboration avec les équipes scientifiques pour donner des retours sur l'utilisation des logiciels qui seront mis à disposition des chercheurs en histoire.



Comme le montre ce diagramme, l'organisation des équipes du projet repose donc sur six domaines très variés qui se distinguent en plus du comité de pilotage

Le projet renouvelle ainsi le lien entre l'histoire, la technique et la culture, en étant réalisé essentiellement par des étudiants, des stagiaires, leurs tuteurs, les enseignants-chercheurs et les permanents. Mais cela constitue aussi certaines difficultés.

III - Difficultés

Le projet est soumis à des contraintes temporelles. En effet, pour limiter les coûts du projet, il a été fait appel à des étudiants. Ceux-ci sont présents un temps limité et ne travaillent pas à temps plein, ce qui génère des problèmes de cohésion et de passation du travail entre les équipes. De plus, les écoles attendent des étudiants un travail différent de celui qui leur est demandé et ils doivent agencer entre elles les différentes attentes. Ce sont ainsi les contraintes associées à la pluridisciplinarité et qui ralentissent la réalisation du projet, mais qui en garantissent la créativité.

Ce projet au long cours présente également des contraintes temporelles. Des délais étaient à respecter, par exemple, une base de données doit être terminée pour Mai 2011 et remplie durant l'été 2011, mobilisant une équipe d'informaticiens et une équipe d'historiens. Il existe aussi des contraintes budgétaires car les dépenses sont limitées. Les principaux frais engagés sont liés à l'achat du matériel nécessaire au projet (tel que deux ordinateurs, la dalle tactile, le pointeur et le zoom, etc.). Le matériel acquis par le musée et par l'Ecole Centrale notamment sont également mis à la disposition du projet (comme par exemple les scanners 3D utilisés pour la numérisation etc.).

Pour limiter les coûts d'un tel projet mais aussi afin d'augmenter la créativité de l'équipe, le musée d'Histoire en accord avec les partenaires associés a choisi de faire appel à des étudiants plutôt qu'à une entreprise. Ainsi, plusieurs difficultés sont liées à la nature et au contexte de ce projet pluridisciplinaire et universitaire : Les acteurs ne travaillent pas tous à temps plein et leur durée de travail est limitée, donc il y a un manque de temps pour avoir une vue globale du projet. La solution est une rigueur dans la passation entre les équipes et une évolution des objectifs au cours du temps.

De plus, les attentes des écoles ou de l'université diffèrent parfois de celles du client (par exemple, au lancement du projet, l'équipe de réalité virtuelle a mené une réflexion sur les différentes interfaces homme/machine alors que leur Ecole attendait des résultats concrets, c'est-à-dire la présentation d'un prototype de programme informatique afin que l'interface soit utilisable à la fin du stage).

Les tâches à exécuter par toutes ces équipes sont diversifiées : comme nous l'avons dit, informatique, 3D, histoire, muséographie, etc. Il y a donc une forte hétérogénéité dans l'expertise mais aussi dans les méthodes de travail, et la communication entre les équipes ne peut se faire que par l'adaptation du vocabulaire et un nombre important de réunions et entrevues. Mais à cela s'ajoute la contrainte spatiale, puisque les étudiants viennent d'établissements différents et qu'aucun local n'est attribué de façon fixe. La difficulté réside alors dans cette communication entre les équipes, qui est pourtant nécessaire au bon déroulement du projet. La coordination et la cohésion est ainsi menacée et les acteurs du projet ont donc la volonté d'organiser de nombreuses réunions pour lutter contre cette menace.

Ces difficultés sont soulevées par la pluridisciplinarité des acteurs du projets, mais le bénéfice n'est pas moindre, puisque la conséquence de celle-ci est une créativité importante. Cette créativité est maintenue par l'aspect empirique du projet.

IV – Créativité

Le projet n'obéit pas à un cahier des charges spécifique et détaillé, mais à des lignes directrices qui sont revues tout au long de sa réalisation. En effet, l'équipe de direction a décrit la demande globale du client et fixe au fur et à mesure les objectifs généraux. De plus, le château des Ducs se veut à la pointe de la technologie muséographique et n'a donc pas imposé de cahier des charges au comité de direction du projet afin de laisser libre cours à la créativité et à l'utilisation des nouvelles technologies qui sont susceptibles d'apparaître durant sa réalisation. Enfin, il réside une certaine rédaction de cahier des charges qui s'effectue au sein des équipes responsables de la réalisation de sous-modules du projet, ce qui permet de garder une direction scientifique précise et rigoureuse. Cette description montre ainsi le côté empirique de la réalisation du projet qui garantit son côté original et créatif tout en restant lié à la rigueur scientifique.

Le château se veut à la pointe de la technologie muséographique et le projet a pour but d'être le plus innovant possible. Par exemple, l'interface créée se doit d'être facile à utiliser, attractive et pédagogique, car elle s'adresse à un public précis : les visiteurs du Château. Pour maintenir cet objectif d'innovation au cours des années, aucun cahier des charges précis n'a été écrit lors de la réunion initiale du comité de pilotage. Les équipes sont alors libres d'innover et imaginer des solutions originales, mais elles doivent par contre définir des cahiers des charges pour le travail qu'elles effectuent : planning, mission, objectifs. Elles doivent à la fin écrire un rapport afin d'assurer la passation de leur travail aux équipes suivantes qui rédigent à nouveau leur cahier des charges. On se rend compte ici d'une des difficultés soulevées, qui est celle du passage de relais, mais la réécriture d'un cahier des charges pour chaque sous-partie du projet amène un renouvellement de créativité et permet au projet de rester à la pointe de la technologie.

Comme nous l'avons vu, de nombreuses réunions sont organisées, et c'est au cours de celles-ci que les équipes ont l'occasion de donner leur point de vue sur la suite du projet. Elles peuvent alors faire part de leurs idées d'innovation tout en prenant en compte les contraintes des autres équipes. Par exemple, l'équipe chargée de l'interface de la base de données a pu faire part de son idée d'amélioration de la géolocalisation à l'équipe chargée de la muséographie et de l'histoire, qui a alors proposé une modification de cette amélioration en fonction d'une contrainte historique.

Nous constatons donc à quel point la pluridisciplinarité et l'empirisme ont permis une innovation au projet malgré une contrainte liée à la non-expertise qui a été surmontée par l'expertise des collaborateurs.

V – Conclusion

En conclusion, le projet Nantes 1900 a pour originalité d'être pluridisciplinaires en rassemblant des étudiants de différents secteurs et en les faisant lier science de l'ingénieur et sciences humaines, et laisse libre cours à leur imagination et leur créativité pour réaliser et améliorer les objectifs globaux prévus. Cela a pourtant certaines limites comme les problèmes d'organisation entre les équipes et les contraintes temporelles liées au temps alloué aux étudiants pour travailler sur ce projet. Mais la centralisation du travail au sein de l'équipe de direction maintient le bon agencement du projet, qui se démarque particulièrement des projets d'entreprise qui est en bonne voie pour arriver au résultat espéré comme en témoigne l'état de l'avancée des travaux après ces trois dernières années.

Bilan du projet

Ce projet qui proposait à la base un travail relativement classique de développement de site web faisant interface d'une base de donnée, s'est révélé être un projet plus original qu'on ne pouvait le penser, et qui nous a amené à une gestion inattendue sur plusieurs points.

La pluridisciplinarité marquant globalement le projet Nantes 1900 a eu des répercussions sur notre travail. Nous avons eu l'occasion d'étudier la maquette du port de Nantes d'un point de vue en rapport direct avec le module HES et non le module informatique. Ceci s'est porté sur une découverte à la fois d'une œuvre d'art mais aussi d'un morceau passionnant de notre Histoire. Il est à noter que nous avons pu survoler brièvement l'ingénierie fluviale et militaire, mais aussi l'économie, le commerce et la société de Nantes au cours de son histoire. Nous avons dû collaborer avec une équipe d'historiens avec lesquels nous avons appris à communiquer en employant un jargon commun alliant notre domaine et le leur, grâce à ce que nous avons appris de la phase de bibliographie.

L'empirisme, deuxième pilier de l'originalité du projet Nantes 1900, nous a entraîné à faire une gestion du projet souple. Nous nous sommes, pour cela, efforcé de suivre une méthode agile (Scrum) mais beaucoup de modifications et de changements dans la gestion du projet sont survenus au cours du grand nombre de réunions entre les équipes. L'objectif principale était de répondre aux besoins de tous en adaptant nos méthodes et notre programme à ces besoins. Le calendrier effectif prouve l'existence de ces changements, car on peut y voir que le super-utilisateur est devenu obsolète, que le développement du chercheur aura duré trois semaines de plus afin de l'adapter aux nouveaux besoins, et que le couple modérateur/informateur a été souvent repris et modifié. Nous avons heureusement prévu ce genre de changements lors de la phase de conception et l'application de la méthode agile et des tests de recette en direct ont permis de gérer tout ceci.

Le planning effectif montre les adaptations qui ont eu lieu durant la phase de réalisation qui était très dépendante des prises de décision au cours du projet, avec des distances temporelles importantes entre les fonctions des utilisateurs à réaliser.

Conclusion

La conclusion de ce livrable unique se décline en trois points : d'abord sur ce que nous retiendrons techniquement d'un tel projet, puis l'expérience liée à la gestion du projet, et enfin le futur du projet réalisé.

Nous avons acquis plusieurs éléments : premièrement, les connaissances sur le contexte du projet, à savoir l'histoire du port de Nantes, l'art de la maquette, mais aussi la façon de gérer un projet pluridisciplinaire et empirique. Deuxièmement, le choix des outils de développement nous a permis d'approfondir nos connaissances en PHP et de découvrir la conception et la gestion d'une base de données orientée objet. Nous avons également eu l'occasion de découvrir les données de type GIS.

Ce projet aura été une expérience du travail de groupe sur plusieurs dimensions. Tout d'abord sur une dimension plus internet, à savoir la collaboration entre les deux membres du binôme afin de réaliser à deux un unique produit. Ensuite bien la dimension externe consistant en une entente avec plusieurs équipes pluridisciplinaires qui auront tenu un rôle dans le développement de ce projet. Enfin nous aurons appris à répondre à la fois aux attentes des tuteurs industriels et des tuteurs coté école, qui sont parfois éloignées, et pour lesquelles il faut faire des compromis afin de satisfaire les deux partis.

Le futur de ce projet a été discuté lors de la réunion du 26 Mai 2011, durant laquelle le programme a été installée sur la machine du projet et a été testé par nos successeurs venant de l'école Centrale. Leur rôle sera d'intégrer ce projet à celui de la dalle tactile, et nous serons recontacté pour un "bonus", à savoir installer le programme sur un serveur en ligne, lorsque celui-ci sera mis en place.

Références

Partie bibliographique

- [1] Jules César. La guerre des Gaules. Ed. Folio Classique. Page 131
- [2] Visite guidée du Château des Ducs de Bretagne
- [3] Catherine Decours. Le Port de Nantes a 3000 ans. Ed Giotto
- [4] Christian Lyonnay. Le Château des Ducs de Bretagne à Nantes. ORTF Nantes Télé Pays de Loire. 13/04/1968 Archives INA <http://www.ina.fr/video/RXF04017686/le-chateau-des-ducs-de-bretagne-a-nantes.fr.html>
- [5] Fondation Belem. Présentation du Belem. http://www.fondationbelem.com/presentation_du_belem.htm
- [6] Myriam Figureau et Richard Leceux. Le Belem à Nantes. France Regions 3 Nantes. 27/05/1986. Archives INA <http://www.ina.fr/video/NAC86052707/le-belem-a-nantes.fr.html>
- [7] Laurent Claude Gloaguen. Histoire du Belem. <http://3mats.net/histoire-du-belem/>
- [8] Marc Flandrois. Aménagement du Port de Nantes. Télé Pays de Loire France régions 3 Nantes. 17/02/1979. Archives INA <http://www.ina.fr/video/RXC04007086/amenagement-du-port-de-nantes.fr.html>
- [9] Conservation du château des Ducs de Bretagne. Collections - la maquette du port de Nantes en 1900. http://www.chateau-nantes.fr/fr/collections/la_maquette_du_port_de_nantes_en_1900/
- [10] Fiche inventaire de la maquette au Château des Ducs http://aip2.iut-nantes.univ-nantes.fr/LotusQuickr/nantes1900/Main.nsf/h_463249FE0F57751EC12574E4004036B4/CBC00821666A8D16C12574E40040AF59/?OpenDocument
- [11] Conservation du château des Ducs de Bretagne. Qui sommes-nous ? http://www.chateau-nantes.fr/fr/qui_sommes_nous/conservation/
- [12] Centre François Viète. Composition de l'équipe. <http://www.sciences.univ-nantes.fr/cfv/organigramme.htm> . Dernière consultation : 14/11/2010
- [13] Centre François Viète. Thèses soutenues. <http://www.sciences.univ-nantes.fr/cfv/theses.html> . Dernière consultation : 14/11/2010.
- [14] Centre François Viète. Descriptif des UES et UEF année universitaire 2010. <http://www.sciences.univ-nantes.fr/cfv/cours/masterHST-programme-2009-10.pdf>
- [15] Centre François Viète. Thèmes de recherche. http://www.sciences.univ-nantes.fr/cfv/themes_recherche.htm

nantes.fr/cfv/themes_recherche.htm . Dernière consultation : 14/11/2010.

- [16] Sophie Jourdin. Projet de « Nantes 1900 » Synthèse générale à mi-parcours. 2010.
- [17] Gilles Vienvenu. Nantes Plans commentés. Plan Monumental p.73
- [18] Gilles Vienvenu. Nantes Plans commentés. Plan Delattre p.77
- [19] François Perez et Lucie Thourault. Rapport de Réalisation projet transversal Gestion documentaire et valorisation scientifique et muséographique de la maquette du port de Nantes. 2009-2010
- [20] Alain Lefebvre. L'architecture client-serveur, aspects techniques, enjeux stratégiques. Armand Colim. 1993
- [21] php.net. PHP Data Objects. <http://Php.net.manual.fr.book.pdo.php>
- [22] Olivier Heurtel. PHP 5 Développer un site web dynamique et interactif. Eni editions. 2004.
- [23] S. Sumathi, S. Esakkirajan. Fundamentals of relationnal database management systems. Springer. 2007.
- [24] Andreas Meier. Introduction pratique aux bases de données relationnelles, deuxième édition. Springer.
- [25] Talel Abdesalem. Cours : Base de données orientées objet. <http://www.infres.enst.fr/~talel/cours/bda/bdoo0.pdf>
- [26] Université de Californie. Object Definition Language. <http://www.ics.uci.edu/~dbclass/ics184/class-notes/odl.pdf>
- [27] National Committee for Information Technology Standards (NCITS). Sql3 Object Model. <http://www.objs.com/x3h7/sql3.htm>
- [28] Php.net. Manuel php. Fonctions PostgreSQL. <http://php.net/manual/fr/ref.pdo-pgsql.php>