

Document technique : Projet Athéna

ATHENA

Ulysse PERRET

Thanusan PARAMESWARAN

Version 1

Rendu le 26/02/2016

I. Introduction

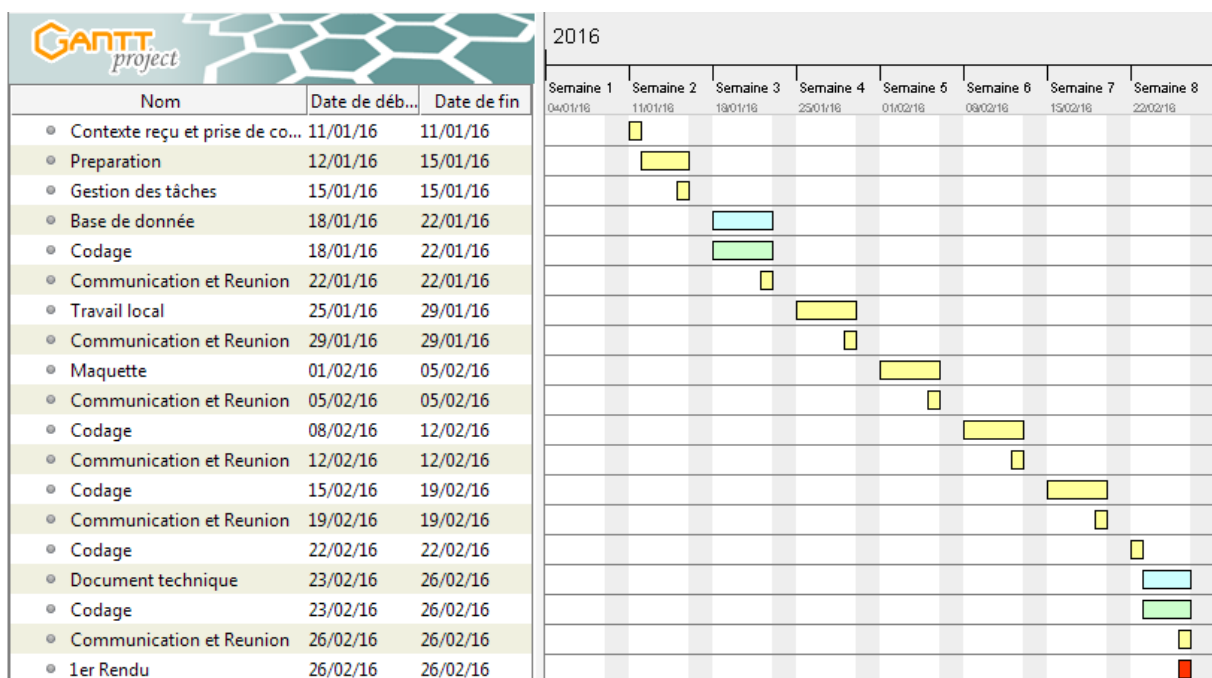
Ce document permet de présenter la structure du dossier, il contient tous les informations nécessaire pour lancer, comprendre et les solutions apportée au contexte.

Ce projet est à la fois un site web, donc un client léger, mais il est responsive et peut s'adapter à l'interface mobile.

On a décidé que tous les vendredis on faisait le point de notre travaille et qu'on se transmet les missions à faire.

On travaille en générale en commun

Voici notre planning de travail du projet à l'heure actuelle



Légende

En Jaune : commun

En Bleu : Ulysse

En Vert : Thanussan

En Rouge : Rendu

Voir annexe 2

II. Sommaire

I.	Introduction-----	page 2
II.	Sommaire-----	page 3
III.	Contexte et Expression des besoins-----	page 4-5
	A. Contexte-----	page 4
	B. Expression du besoin-----	page 4-5
IV.	Prérequis-----	page 6-5
V.	Présentation du dossier Athéna-----	page 7
VI.	Plan du site-----	page 8-9
VII.	Moyen d'accès-----	page 10-11
	A. Accès au site-----	page 10
	1. Sur internet-----	page 10
	2. Localement-----	page 10
	B. Accès aux documents-----	page 10-11
	1. Accès FTP-----	page 10
	2. Portfolio-----	page 11
	3. Hébergement par site web-----	page 11
VIII.	Réponse au besoin-----	page 12-17
IX.	Conclusion-----	page 18
X.	Annexes-----	page 19-25
	Annexes 1 : Structure des tables-----	page 19-20
	Annexe 2 : Gantt-----	page 21
	Annexe 3 : Maquette-----	page 22
	Annexe 4 : Code Source-----	page 23-25

III. Contexte et Expression des besoins

A. Contexte

La société Athéna effectue un appel d'offre auprès des différentes Ssii et web agency. Celui-ci porte sur la création d'un CMS (Contrat Management System) sur mesure

Notre société devra répondre à cet appel d'offre et proposer une solution complète et maquettée à la société Athéna

Une fois à la réponse l'appel d'offre validé par le client, chaque prestataire devra réaliser le CMS et le présenter au client en s'appuyant au dossier technique complet.

Pour ce projet, notre équipe composée de Thanussan et Ulysse, représentant une société de service, avec un chef de projet identifié qui coordonnera l'activité de son équipe et définira les axes de travail. Seul ce chef de projet pourra communiquer avec les formateurs qui représenteront la société ATHENA afin de pouvoir demander des compléments d'information nécessaire au projet

Un espace de stockage centralisé sera mis à disposition pour les équipes. La sécurisation de l'accès aux données est à leurs charges.

B. Expression du besoin

La société Athéna est une société spécialisée dans le développement de sites Internet et Intranet. Son activité dépend actuellement fortement des systèmes CMS existants, qui ne correspondent que partiellement aux besoins des clients.

Cela entraîne un temps de développement plus important, certains modules proposés par les CMS classiques devant être désactivés, remodelés voir même redéveloppés.

L'appel d'offre porte sur le développement d'un outil de gestion de contenu répondant aux besoins suivants. La liste exprime les fonctionnalités minimales devant apparaître, chaque prestataire est libre d'y ajouter des fonctionnalités additionnelles.

- L'objectif de ce CMS est de déployer rapidement des sites de ventes de produits particuliers.
- Les usagers doivent s'identifier par un couple login/mot de passe pour bénéficier des fonctionnalités d'achat et de vente. Un utilisateur non connecté ne peut que parcourir la liste des articles disponibles.
- Les articles proposés à la vente sont stockés dans une base de données. Les usagers authentifiés peuvent ajouter de nouveaux articles à cette base de données.

- A chaque article correspond une fiche contenant les informations détaillées de celui-ci.
- Lorsqu'un article est ajouté à la base de données, il est directement disponible à la vente. Les administrateurs vérifient la fiche descriptive et la valide. Une fois cette fiche validée, les informations renseignées ne sont plus modifiables.
- Lorsqu'un usager désire mettre un article en vente, il le sélectionne dans la base de données et indique qu'il possède un exemplaire à vendre ainsi que son prix de vente.
- Un usager acheteur doit se voir la liste des vendeurs pour un article et les prix demandés par chacun.
- Une fonction de recherche doit permettre aux usagers de localiser un article par son nom ou par des tags associés.
- Un système de back-office doit permettre aux sociétés clientes la gestion des comptes utilisateurs, des articles présents en base de données et de catégories d'articles.
- Un système de panier doit permettre aux usagers d'acheter plusieurs articles simultanément.
- Un système de contact doit permettre l'envoi de mails de la part des usagers aux administrateurs du site.
- L'application doit être accessible aussi bien sur pc que sur mobile

Ce système étant destiné à des clients travaillant sur des secteurs d'activité distincts, votre solution devra présenter une modularité suffisante pour s'adapter à n'importe quel type de contenu.

IV. Prérequis

Pour lancer le projet nous avons utilisé chacun un ordinateur pour le développement du site web.

Le projet a été réalisé sur **NeatBeas IDE 7.4** pour le code PHP. En effet notre équipe maitriser cette interface et pour permettre d'être le plus efficace possible le fait d'être sur la même méthode de travail et mieux répartir les missions à faire. Eventuellement on utilise **Notepad +** pour des modifications spécifiques, sur certaines pages.

Pour la base de donnée on utilise **PhpMyAdmin** sur lequel on a travaillé sur la base de donnée **athena.sql**. La connexion sur la base de donnée utilise des requête PDO via la commande :

```
$bdd = new PDO ('mysql: host=localhost: 3306; dbname=athena','root',")
```

Voir le fichier : connection.php pour plus d'information.

Pour travaille avec PhpMyAdmin et accéder au projet localement par un localhost, on a utilisé **WAMP**, étant donné qu'on travaille en local et qu'on les membres de l'équipe travaille sur pc sous exploitation Windows.

Le projet Athéna est un site web responsive. Il s'agit donc d'un client léger qui est capable de fonctionner sur mobile, et donc aussi de répondre à la problématique de fonctionnement sur mobile.

Pour que le site soit responsive, en plus d'ajouter du style, on a décidé de télécharger le fichier **Bootstrap**. Pour que le site soit considéré comme client léger, le bootstrap est intégrer dans le ftp du site, et donc au ressources disponible et localement on la donner dans le dossier Athena :
Athena\bootstrap-3.3.6-dist

Pour gérer la connexion au serveur, et mettre notre projet sur le web, nous avons décidé de prendre un hébergement sur le site **1 & 1** et de transférer nos donnée avec FTP via **Filezilla**.

Pour réaliser le Gantt on a utilisé **GanttProjet**.

v. Présentation du dossier Athéna

Le dossier a 6dossiers, et tous les pages web, codé en PHP pour que vous puissiez regarder localement toutes les pages. Le détail de la hiérarchisation des pages se trouve à la page suivante avec le plan du site

1. Base de donnée : ce dossier à la base de donnée athena.sql, ainsi que les tables qui la compose
2. Bootstrap-3.3.6-dist : ce dossier est pour le bootstrap.

En effet pour que le site soit responsive et puisse être accessible sur une interface mobile on a utilisé bootstrap.

En son sein il a tous les caractéristiques qu'on a besoin pour gérer les différents cas de plateforme, selon l'écran de l'utilisateur

On utilise notamment le bootstrap.min.css pour minimiser le code pour qu'il respecte l'interface mobile.

Celui-ci se trouve à Athena\Athena\bootstrap-3.3.6-dist\css

3. Document : il a le contexte Athéna et tous les documents utiles lié

Contexte Athena : le contexte de base donnée initialement

Contexte situation 2 : le contexte modifié pour être applicable, avec les réponses au besoin

Document technique : le fichier que vous regardez, qui donne tous les informations utiles

Le dossier Annexe est tous les éventuelle annexes et repris dans ce document technique

4. Css : tous les styles utilisés. Il intègre aussi le bootstrap.
5. Nbproject : le projet et ses propriétés, liées du fait que le projet a été codé en Neat Beas
6. Plan: ce dossier reprend tous les pages web avec une structuration selon le plan du site web fait, détaille à la page suivante

VI. Plan du site

Le plan du site est le page : plan.php

Démarrage

Accueil : Accueil.php

Athena : Accueil.php

Afficher les produits : affichep.php

Inscription : inscription.php

Connexion : connexion.php

Traitement de la Connexion : connection.php

Barre de navigation : menu1.php

Site

Accueil : Accueil2.php

Athena : Accueil2.php

Afficher les produits : affichep.php

Ajouter un produit : vendre.php

Afficher les produits que vous vendez : affichepu.php

Vendre un produit : vendrep.php

Compte : compte.php

Afficher votre compte : afficheut.php

Modifier votre compte : modifuco.php

Rechercher : rechercher.php

Contact : contact.php

Panier : panier.php

Ajouter un panier : ajoutpanier.php

Consulter un panier : consulpanier.php

Barre de navigation : menu2.php

Déconnexion : deconnexion.php

pour Accéder au tableau de bord : admin.php (caché sur le site)

Tableau de bord

Athena: Acceuil3.php

Accueil : Acceuil3.php

Gestion des comptes : gcompte.php

Afficher les comptes : afficheu.php

Modifier les comptes : modifu.php

Ajouter les comptes : ajoutu.php

Fiches : fiche.php

Afficher les fiches : affichef.php

Valider les fiches : validerf.php

Articles : carticle.php

Afficher les articles : affichec.php

Catégories : mcategorie.php

Afficher les catégories : affichec.php

Ajouter les catégories : ajoutc.php

Barre de navigation : menu3.php

Deconnexion : deconnexion.php

VII. Moyen d'accès

A. Accès au site

1. Sur internet

Le site web est accessible sur internet, il est en ligne à l'adresse suivante :

www.ulysseperretathena.com

Hébergement du site est fait par 1 & 1.

Pour se connecter en administrateur voici le compte de l'administrateur par défaut

Login : admin

Mot de passe : admin

Un utilisateur standard :

Login : test

Mot de passe : test

2. Localement

Pour installer le site localement il vous faudra installer le WAMP et le lancer, puis mettre tout le dossier Athena, décompressé avant dans C:\wamp\www

Le site web sera accessible localement en lançant wamp

Il faudra implémenter la base de données athena.sql présent dans le dossier nbprojet.

Si vous n'hésitez pas à taper dans la barre d'adresse <http://localhost/Athena/Acceuil.php>

B. Accès aux documents

On propose différentes solutions pour pouvoir accéder au document

1. Accès FTP

Voici ce que vous avez besoin pour accéder au ftp

Hôte : ulysseperretathena.com

Identifiant : [u84137149-ulysse](#)

Mot de passe : baleine

2. Portfolio

Vous pouvez aussi retrouver tous les documents et document sur les portfolio suivant

Ulysse Perret : www.ulyseperret.com

Il transfère ses documents via Dropbox dans le dossier Athena

Lien du dossier du projet:

<https://www.dropbox.com/sh/8xsrmt6ouh80uq5/AAAGNcNdilDtKjEOB4Ms0kBaa?dl=0>

3. Hébergement par site web

L'école fournira un hébergement des documents en plus par un site web (non mise en place)

VIII. Réponse au besoin

Dans cette partie nous traiterons de répondre à l'expression des besoins défini par le contexte.

- L'objectif de ce CMS est de déployer rapidement des sites de ventes de produits particuliers

Nous avons décidé de faire un site de commerces, où les utilisateurs peuvent ajouter, acheter ou vendre le produit avec un système de panier. Cependant les fonctionnalités de livraison, d'achat avec carte bancaire n'ont pas été mises en place pour que le site soit le plus simple mais permette la vente et l'achat de biens. Les utilisateurs savent qui est le vendeur éventuel

- Les usagers doivent s'identifier par un couple login/mot de passe pour bénéficier des fonctionnalités d'achat et de vente. Un utilisateur non connecté ne peut que parcourir la liste des articles disponibles

Le site est décomposé en trois stades différents. Le visiteur pourra qu'accéder au page Accueil, connexion et inscription avec la demande du login et du mot de passe. Le visiteur pourra voir les produits en vente sur la page accueil. On stocke le couple login et mot de passe dans la base de données.

La requête SQL utilisée sont :

- Sur la page connexion.php on sélectionne la table utilisateur déjà rentrer avec : `SELECT * FROM utilisateur WHERE login = : login AND motdepasse = : motdepasse`
- Sur la page inscription on insère un utilisateur dans la base de données avec : `"INSERT INTO utilisateur (nom, prenom, mail, login, motdepasse, groupe, rue, codepostal, ville) VALUES ('".$nom."', '".$prenom."', '".$email."', '".$email."', '".$password."', 4, '".$rue."', '".$codepostal."', '".$ville."")"`

Il est à noter que sur les pages du site, pour éviter que le visiteur accède au page avec la barre de navigation, on a écrit :

```
$login = $_SESSION['login'];  
  
$motdepasse = $_SESSION['motdepasse'];  
  
if(($login=="") && ($motdepasse=="") )  
  
{ header('Location: deconnexion.php'); }
```

Avec deconnexion.php qui déconnectera automatiquement

Voir inscription.php ; connexion.php pour le code lié à la fonctionnalité login/motdepasse

Tablier utilisateur

utilisateur
Clé primaire : <u>id(int)</u>
Nom (varchar) Prenom(vvarchar) mail(vvarchar) login(vvarchar) motdepasse (varchar) groupe (<u>int</u>) rue (varchar) codepostal(<u>int</u>) ville (varchar)

- Les articles proposés à la vente sont stockés dans une base de données. Les usagers authentifiés peuvent ajouter de nouveaux articles à cette base de donné.

Pour stocker les annonces, les produits différents, on créer une base de donné produit.

Table produit

Produit
Clé primaire : <u>id_produit (int)</u>
Titre (varchar) Image(vvarchar) Prix (<u>int</u>) Fiche(vvarchar) Stock (varchar) Validation (bool) Id_vendeur (<u>int</u>) Id_catégorie(<u>int</u>) Date (date) Tag (varchar)

Pour l'ajout d'un article à la base de donné la requête SQL sera : "INSERT INTO `athena`.`produit` (`id_produit`, `titre`, `image`, `prix`, `fiche`, `stock`, `validation`, `id_vendeur`, `id_categorie`, `date`, `tag`) VALUES (NULL, '\$nom', '\$prix', '\$fiche', '1', '0', '\$id', NOW(), '\$tag');" "

La page qui traite de l'ajout du produit est vendre.php

- A chaque article correspond une fiche contenant les informations détaillées de celui-ci.

L'affichage de la table produit en annonce se fait avec la requête SQL : "SELECT `produit`.`titre`, `produit`.`date`, `produit`.`fiche`, `produit`.`prix`, `utilisateur`.`nom` FROM `produit`, `utilisateur` WHERE `utilisateur`.`id` = `produit`.`id_vendeur` Order By `date` desc limit 10 "

Voir affichep.php

- Lorsqu'un article est ajouté à la base de données, il est directement disponible à la vente.

Les administrateurs vérifient la fiche descriptive et la valide. Une fois cette fiche validée, les informations renseignées ne sont plus modifiables

Pour la validation des fiches on utilisera la fonction

```
"UPDATE `athena`.`produit` SET `validation` = '1' WHERE `produit`.`id_produit` = '". $id. "'";
```

Le booléen validation permet de ne pas voir les fiche validée, et donc de ne sont pas modifiable après.

Voir la page : validerf.php

- Lorsqu'un usager désire mettre un article en vente, il le sélectionne dans la base de données et indique qu'il possède un exemplaire à vendre ainsi que son prix de vente.

Même si on n'a pas mis de sélection sur la base de données, l'utilisateur peut ajouter un produit et vendre son bien avec la page ajouter un produit.

- Un usager acheteur doit se voir la liste des vendeurs pour un article et les prix demandés par chacun

L'affichage actuel des annonces des produits permet l'affichage du vendeur et du prix qu'il demande

Voir : affichagep.php

- Une fonction de recherche doit permettre aux usagers de localiser un article par son nom ou par des tags associés.

On a fait une page Recherche sur le site qui prend en compte le nom, la description du produit mais aussi des tags éventuels ajoutés.

La commande SQL est : "SELECT * FROM produit where ('nom' LIKE '\$recherchec') OR ('fiche' LIKE '%\$recherchec%')" OR ('tag' LIKE '%\$recherchec%')" ORDER BY 'nom' ;"

Voir rechercher.php

- Un système de back-office doit permettre aux sociétés clientes la gestion des comptes utilisateurs, des articles présents en base de données et de catégories d'articles.

On a créé un page admin.php, caché sur le site pour que l'administrateur accède, avec un login/mot de passe pour aller sur le tableau de bord et notamment sur la gestion des comptes

La commande SQL : `SELECT * FROM utilisateur WHERE login = :login AND motdepasse = :motdepasse AND groupe = '3'`

Pour la gestion des comptes,

On a créé la page gcompte.php qui intègre lui-même différent page

Pour afficher les comptes on utilise afficheu.php

Commande SQL : `"SELECT * FROM utilisateur"` ;

Pour modifier les comptes on utilise modifu.php

Commande SQL : `"UPDATE athena.utilisateur SET nom = ".$nom.", prenom = ".$prenom.", groupe = ".$groupe.", mail = ".$email.", login = ".$email.", motdepasse = ".$password.", rue= ".$rue.", codepostal = ".$codepostal.", ville = ".$ville." WHERE `utilisateur`.`id` = ".$id." ;"`

Pour ajouter les comptes on utilise ajoutu.php :

`"INSERT INTO utilisateur(nom, prenom, groupe, mail, login, motdepasse, rue, codepostal, ville) VALUES (".$nom.", ".$prenom.", ".$groupe.", ".$email.", ".$email.", ".$password.", ".$rue.", ".$codepostal.", ".$ville.")"`

L'utilisateur peut gérer son compte sur la page compte.php avec les pages afficheut.php qui affiche son compte et modifuco.php pour qu'il puisse modifier celui-ci

Pour afficher la commande SQL : `"SELECT * FROM utilisateur where `login` = ".$login." AND `motdepasse` = ".$motdepasse." ;"`

Pour modifier la commande SQL : `"SELECT * FROM utilisateur where `login` = ".$login." AND `motdepasse` = ".$motdepasse." ;"`

Voir admin.php pour l'accès au tableau de bord ; Voir gcompte.php, afficheu.php, ajoutu.php, modifu.php pour la gestion des comptes en tant qu'admin. Pour l'utilisateur la gestion de son compte se fera avec : compte.php, afficheut.php et modifuco.php

Des articles présents en base de données

[Voir carticle.php](#)

Catégories d'articles.

La page du tableau de bord qui regroupe les fichiers traitant des catégories d'article est mcategorie.php.

Pour afficher les catégories on utilise la page affichec.php

Commande sql: "SELECT * FROM `categorie` "

Pour ajouter une catégorie on utilise la page ajoutc.php

Commande SQL : "INSERT INTO categorie (libelle_categorie) VALUES('".\$libelle."")"

Table Catégorie

Catégorie
Clé primaire : <u>id_categorie (int)</u>
Libelle_categorie (varchar)

Voir mcatégorie.php, affichec.php et ajoutc.php ;

- Un système de panier doit permettre aux usagers d'acheter plusieurs articles simultanément

On a fait un système de panier qui permet d'ajouter un produit à la fois. Pour des besoins de temps et de fonctionnalité on n'a pas géré la possibilité d'acheter plusieurs articles en simultané.

Le formulaire actuel permet juste l'ajout d'un panier

La page du panier est : panier.php

Pour l'ajout du panier on utilise le fichier : ajoutpanier.php .

Commande SQL : " INSERT INTO `athena`.`commande` (`id_commande`, `id_utilisateur`, `id_produit`, `stock`, `prix`, `nom`)"

. "VALUES (NULL, ".\$idu.", ".\$idp.", ".\$stock.", ".\$prix.", ".\$nom."); "

Pour la consultation du panier : consulpanier.php .

Commande sql : "SELECT `prix`, `stock`, `id_produit`, `nom` FROM `commande` WHERE 1 ";

Table Commande

Commande
Clé primaire : <u>id_commande (int)</u>
Id_utilisateur (int)
Id_produit (int)
Stock (int)
Prix (int)
Nom (varchar)

Table Panier

Commande
Clé primaire : <u>id_panier (int)</u>
Id_utilisateur (int)
Id_commande (int)
Date_panier (date)

A voir : panier.php ; ajoutpanier.php ; consulpanier.php.

- Un système de contact doit permettre l'envoi de mails de la part des usagers aux administrateurs du site.

On utilise la fonction mail. La page pour envoyer un mail est contact.php

Fonction : mail(\$destinataire, \$sujet, \$message);

- L'application doit être accessible aussi bien sur pc que sur mobile

Le site est responsive. Pour cela on a utilisé bootstrap.

Pour intégrer bootstrap sur toutes les pages on a fait :

```
<link rel="stylesheet" href="bootstrap-3.3.6-dist/css/bootstrap.min.css">
```

```
<link rel="stylesheet" href="css/bootstrap.css">
```

Voir annexe 1 pour les table.

Voir annexe 4 pour les codes.

IX. Conclusion

Ce projet répond à la problématique de l'examen, du fait que c'est à la fois un client léger, qui est consultable sur internet ou localement mais ne requière pas a fortiori de base un téléchargement.

De plus le critère de la mobilité et de la responsivité du site a été mis en avant, pour faire en sorte que le projet respecte deux critères de sélection demandé par l'examen.

Le projet n'est pas encore terminer, et le site n'est pas entièrement en ligne à l'heure actuelle, même si les fichiers et pages sont présent, le FTP et la base de donné établie, la connexion ne l'ai pas encore actuellement du fait qu'il faut refaire les pages pour permettre la connexion des page à la nouvelle base de donné.

X. Annexes

Annexes 1: Structure des tables

Voici les tables utilisé

Table Catégorie

Catégorie
Clé primaire : <u>id_categorie (int)</u>
Libelle_categorie (varchar)

Table Produit

Produit
Clé primaire : <u>id_produit (int)</u>
Titre (varchar) Image(vvarchar) Prix (<u>int</u>) Fiche(vvarchar) Stock (varchar) Validation (bool) Id_vendeur (<u>int</u>) Id_catégorie(<u>int</u>) Date (date) Tag (varchar)

Table utilisateur

utilisateur
Clé primaire : <u>id(int)</u>
Nom (varchar) Prenom(vvarchar) mail(vvarchar) login(vvarchar) motdepasse (varchar) groupe (<u>int</u>) rue (varchar) codepostal(<u>int</u>) ville (varchar)

Table Commande

Commande
Clé primaire : <u>id_commande (int)</u>
Id_utilisateur (int)
Id_produit (int)
Stock (int)
Prix (int)
Nom (varchar)

Table Panier

Commande
Clé primaire : <u>id_panier (int)</u>
Id_utilisateur (int)
Id_commande (int)
Date_panier (date)

Table Groupe

Commande
Clé primaire : <u>id_panier (int)</u>
Id_utilisateur (int)
Id_commande (int)
Date_panier (date)

Relation des tables

L'id-produit de la table Produit est relié en clé étrangère à l'id_produit de la table commande

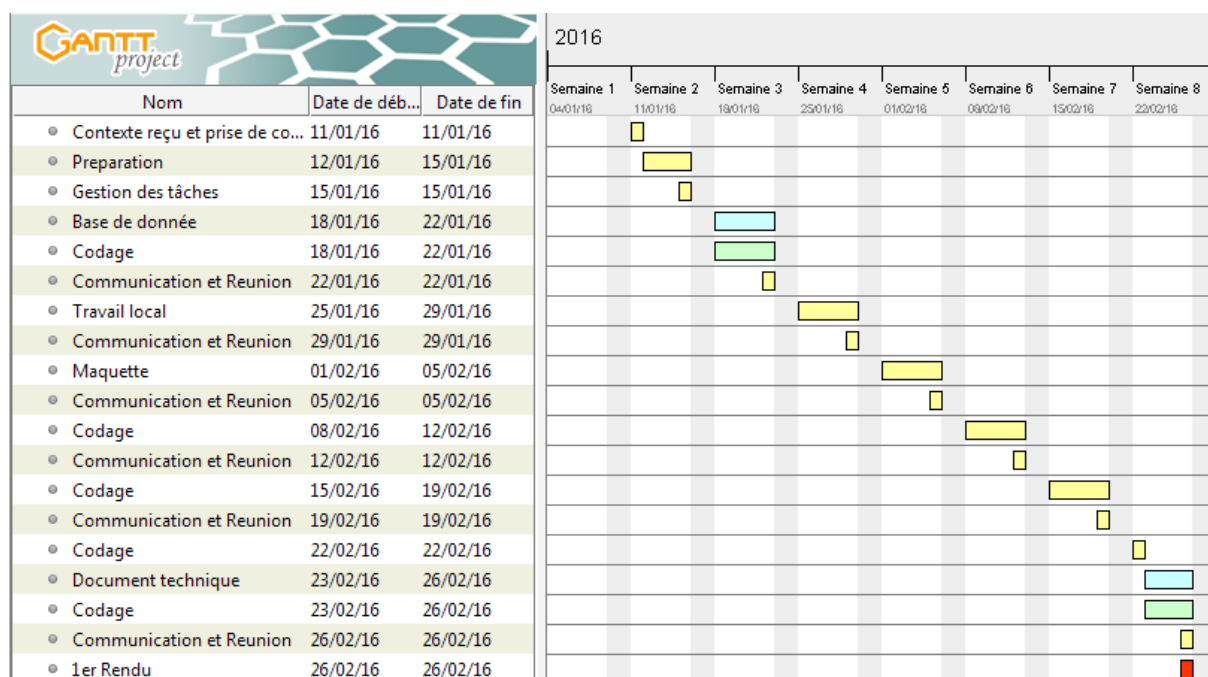
L'id_utilisateur de la table utilisateur est relié à l'id_utilisateur de la table Groupe

L'id_utilisateur de la table utilisateur est relié à l'Id_vendeur de la table Vendeur

L'id_utilisateur de la table utilisateur est relié à l'Id_vendeur de la table Commande

L'id _categorie de la table catégorie est relié à l'id _categorie de la table produit

Annexe 2 : Gantt



Légende : En Jaune : commun/ En Bleu : Ulysse / En Vert : Thanussan/En Rouge : Rendu

Date de début: 11 janvier 2016/ Date de 1^{er} rendu : 26 Février 2016

Tâches

Contexte reçu et prise de connaissance : Reçu du projet

Préparation : étape de réflexion et d'orientation du projet, cherché à cibler le projet

Base de donn   : cr  ation de la base de donn   sur MySQL

Codage : t  che ou on cr  e des pages web.

Travail local : centralisation des postes, et fixation du contexte en local. Mise de l'accord du travail en commun.

Document technique :   criture du document technique et pr  paration aux donn  es

Maquette : travaille sur le style et sur l'aspect responsive du site

Communication et R  union : mise en accord de l'  volution du projet

Travail sur internet : travail sur le site en ligne.

Rendu : tous les rendus du projet. Le rendu Final est le rendu tel qu'il est pr  sent   actuellement

Annexe 3 : Maquette

La page Accueil.php

Athena

Accueil

Inscription

Se connecter

Athena

Site de vente entre particulier

Athena vous propose de vendre et d'acheter vos produits :

- * Proposez vos produits.
- * Rechercher un produits.
- * Acheter vos produits.

Consultation des produits

Voici toutes les dix derniers produits ajoutés

Titre	Date	Description	Prix	Vendeur
test	2016-02-24	test	0	test
PC	2016-02-23	Pc vendu par admin	10	test
Meuble Ikea	2016-02-23	meuble ikea	200	Balou
trousse	2016-02-23	la trousse de ma grand-mere	50	Perret
t-shirt logo	2016-02-23	T-shirt avec un logo bleu	12	test
test	2016-02-23	test	0	test
Ecran	2016-02-23	un ecran plat Dimension : 1000*800 Etat d occasion marque Asus.	127	test
clavier	2016-02-23	clavier azert	20	test

Annexe 4 : Code Source

Ici son répertoriée toutes les commandes citées.

Pour la connexion, Sur la page connexion.php on utilise : `SELECT * FROM utilisateur WHERE login = : login AND motdepasse = : motdepasse`

Pour insérer un utilisateur Sur la page inscription : `«INSERT INTO utilisateur (nom, prenom, mail, login, motdepasse, groupe, rue, codepostal, ville) VALUES ("$.nom." ', "$.prenom.", "$.email.", "$.email.", "$.password.", 4, "$.rue.", "$.codepostal.", "$.ville.")»`

Il est à noter que sur les pages du site, pour éviter que le visiteur accède au page avec la barre de navigation, on a écrit :

```
$login = $_SESSION ['login'] ;  
  
$motdepasse = $_SESSION ['motdepasse'] ;  
  
If (($login=="") && ($motdepasse==""))  
  
{ header ('Location: deconnexion.php') ;}
```

Voir inscription.php ; connection.php pour le code lié à la fonctionnalité login/motdepasse

Pour l'ajout d'un article à la base de donné la requête SQL sera : `"INSERT INTO `athena`.`produit` (`id_produit`, `titre`, `image`, `prix`, `fiche`, `stock`, `validation`, `id_vendeur`, `id_categorie`, `date`, `tag`) VALUES (NULL,"$.nom.", "$.prix.", " ", "$.fiche.", " '1', '0', "$.id.", " ", NOW(), "$.tag."); "`

La page qui traite de l'ajout du produit est vendre.php

L'affichage de la table produit en annonce se fait avec la requête SQL :

```
"SELECT `produit`.`titre` , `produit`.`date`, `produit`.`fiche`, `produit`.`prix`, `utilisateur`.`nom`  
FROM `produit`, `utilisateur` WHERE `utilisateur`.`id` = `produit`.`id_vendeur` Order By `date` desc  
limit 10 "
```

Voir affichage.php

Pour la validation des fiches on utilisera la fonction : "UPDATE `athena`.`produit` SET `validation` = '1'
WHERE `produit`.`id_produit` = ".\$id."";

Voir la page : validerf.php

Fonction recherché

La commande SQL est : "SELECT * FROM produit where ('nom' LIKE '\$recherchec') OR ('fiche' LIKE '%\$recherchec%')" OR ('tag' LIKE '%\$recherchec%') ORDER BY 'nom' ;"

Voir rechercher.php

La commande SQL : SELECT * FROM utilisateur WHERE login = : login AND motdepasse = :
motdepasse AND groupe = '3'

Permet de vérifier que l'utilisateur est bien administrateur

Pour afficher les comptes on utilise afficheu.php

Commande SQL : "SELECT * FROM utilisateur" ;

Pour modifier les comptes on utilise modifu.php

Commande SQL : "UPDATE athena.utilisateur SET nom = ".\$nom.", prenom = ".\$prenom.", groupe =
".\$groupe.", mail = ".\$email.", login = ".\$email.", motdepasse = ".\$password.", rue = ".\$rue.",
codepostal = ".\$codepostal.", ville = ".\$ville." WHERE `utilisateur`.`id` = ".\$id." ;"

Pour ajouter les comptes on utilise ajoutu.php : "INSERT INTO utilisateur(nom, prenom, groupe,
mail, login, motdepasse, rue, codepostal, ville) VALUES (".\$nom.", ".\$prenom.", ".\$groupe.",
".\$email.", ".\$email.", ".\$password.", ".\$rue.", ".\$codepostal.", ".\$ville.")"

L'utilisateur peut gérer son compte sur la page compte.php avec les pages afficheut.php qui affiche
son compte et modifuco.php pour qu'il puisse modifier celui-ci

Pour afficher la commande SQL : "SELECT * FROM utilisateur where `login` = ".\$login." AND
`motdepasse` = ".\$motdepasse." ;"

Pour modifier la commande SQL : "SELECT * FROM utilisateur where `login` = ".\$login." AND
`motdepasse` = ".\$motdepasse." ;"

Voir admin.php pour l'accès au tableau de bord ; Voir gcompte.php, afficheu.php, ajoutu.php, modifu.php pour la gestion des comptes en tant qu'admin. pour l'utilisateur la gestion de son compte se fera avec : compte.php, afficheut.php et modifuco.php

Pour afficher les catégories on utilise la page affichec.php

Commande sql: "SELECT * FROM `categorie` "

Pour ajouter une catégorie on utilise la page ajoutc.php

Commande SQL : "INSERT INTO categorie (libelle_categorie) VALUES ('".\$libelle."')"

Voir mcategorie.php, affichec.php et ajoutc.php ;

Pour l'ajout du panier on utilise le fichier : ajoutpanier.php.

Commande SQL : " INSERT INTO `athena`.`commande` (`id_commande`, `id_utilisateur`, `id_produit`, `stock`, `prix`, `nom`) VALUES (NULL, ".\$idu.", ".\$idp.", ".\$stock.", ".\$prix.", ".\$nom."); "

Pour la consultation du panier : consulpanier.php.

Commande sql : "SELECT `prix`, `stock`, `id_produit`, `nom` FROM `commande` WHERE 1 ";

A voir : panier.php ; ajoutpanier.php ; consulpanier.php.

On utilise la fonction mail. La page pour envoyer un mail est contact.php

Fonction : mail (\$destinataire, \$sujet, \$message);

Pour intégrer Bootstrap sur toutes les pages on a fait :

<link rel="stylesheet" href="bootstrap-3.3.6-dist/css/bootstrap.min.css">

<link rel="stylesheet" href="css/bootstrap.css">