# 

**NF17 :**

**Projet Nimpstore**

Groupe : TD mercredi 16h30-19h30

* Chef de projet : Thomas Pelletier
* Responsable Modélisation : Irène Rong
* Responsable Technique : Etienne Chognard
* Responsable Qualité : Emilie Confais

**Table des matières**

# 

# **Lettre de mission………………………………………………………………..…3**

# **Note de Clarification………………………………………………………...…… 5**

# **Organigramme Produit…………..……………………………………….....…. 10**

1. **Modèle Conceptuel de données………………………………………..…..….14**
2. **Modèle relationnel……………....………………………………………..…..….15**
3. **Normalisation…………………….……………………………………….……….16**
4. **Conclusion………………………..………………………………………....…….18**

# **Lettre de Mission**

## ORGANISME CLIENT

### Identification

Nous allons réaliser le projet pour l’Université de Technologie de Compiègne localisée rue Roger Couttolenc, 60200 Compiègne.

### Interlocuteurs

Nous aurons deux interlocuteurs au cours de la réalisation de ce projet : Stéphane Crozat dirigera la maîtrise d’ouvrage et Kévin Carpentier sera notre principal contact au sein de l’organisme client.

Les adresses email pour les contacter sont :

* pour Stéphane Crozat : stephane.crozat@utc.fr
* pour Kévin Carpentier : [kevin.carpentier@hds.utc.fr](mailto:kevin.carpentier@hds.utc.fr)

## NATURE DU PROJET

### Type de projet

Le projet demande la réalisation d’une plate-forme de téléchargement d’applications et de ressources. Pour permettre l’utilisation de la plate-forme, il est nécessaire d’implémenter un site web en langage HTML/CSS et PHP et une base de données permettant de gérer les téléchargements des utilisateurs ainsi que faire des analyses sur les informations en utilisant PostgreSQL. Par ailleurs, l’utilisation de la plate-forme doit être facile et ludique pour l’utilisateur et le site doit présenter une ergonomie agréable.

### Origine et objectif du projet

Les outils tels que les smartphones étant de plus en plus performants, ils permettent aujourd’hui d’installer une très grande variété d’applications. Pour répondre aux volontés des utilisateurs, il est nécessaire pour les sociétés vendant ces applications de proposer des plates-formes de téléchargement de leurs produits. C’est pourquoi ce projet nous demande de réaliser un tel produit. Celui-ci doit répondre à tous les critères de l’organisme client et doit proposer une interface agréable et de bonne qualité pour les utilisateurs. Le but de l’organisme client étant d’augmenter le nombre de leurs clients grâce à la mise en place de cette nouvelle plate-forme.

### Délais

Le produit final est attendu pour le Mercredi 17 juin 2015.

## MOYEN MIS A DISPOSITION

### Budget

Aucun

Réserve budgétaire pour aléas

Aucune

### Intervenants affectés au projet

Chef de projet : Thomas Pelletier

Responsable modélisation : Irène Rong

Responsable technique : Etienne Chognard

Responsable qualité : Emilie Confais

### Moyens matériels

L’Université de Technologie de Compiègne nous met à disposition leurs locaux qui disposent de tout le matériel nécessaire ainsi que des logiciels utiles au développement du projet.

## RESPONSABILITE – AUTONOMIE

Le chef de projet dépend de Kévin Carpentier, notre contact dans l’organisme client. Il travaille en autonomie pour gérer l’organisation de l’équipe d’un point de vue technique et temporel. En revanche, il doit contacter le référent si nous souhaitons apporter des modifications au produit attendu ou pour d’éventuels problèmes de conception, de technique, de budget ou d’organisation. Il n’y a pas de rencontres fixées entre le référent et le chef de projet, à part pour le rendu des livrables, ils se contactent quand cela est nécessaire. Le chef de permet de coordonner les membres de l’équipe pour permettre un travail efficace et rigoureux.

## AUTHENTIFICATION

Kévin Carpentier Thomas PELLETIER

# **Note De Clarification**

## CONTEXTE

L’organisme client de ce projet est l’Université de Technologie de Compiègne représentée par M. Stéphane Crozat, responsable de l’UV NF17.

Dans le cadre de ce cours, notre équipe formée de 4 étudiants a été mobilisée pour concevoir et implémenter une base de données pour la gestion d’un nouveau produit.

Il s’agit du Nimpstore, qui est un service d'achat en ligne d'applications mobiles. L’organisme souhaite ainsi proposer à ses clients une plate-forme de téléchargement de différentes applications et ressources.

1. **DÉFINITION DU PROJET**

### Problématique

En plus d’offrir aux clients la possibilité d'acheter des applications mobiles en ligne, la réalisation du projet doit permettre à l’organisme client de pouvoir suivre toutes les actions effectuées sur la plate-forme afin de pouvoir recueillir différentes informations et chiffres clés relatifs au fonctionnement du Nimpstore.

### Objectifs visés

La base de données devra répondre aux besoins de trois types de personnes : l’utilisateur, l’administrateur et l’analyste. Les objectifs sont donc de proposer :

A l’utilisateur :

· Une liste des applications compatibles avec ses appareils

· Un historique de ses achats et installations

· Une fonctionnalité pour gérer ses terminaux, ajouter ou retirer des applications

A l’administrateur :

· Une fonctionnalité pour ajouter des applications et des ressources

· Une fonctionnalité pour ajouter des éditeurs

· Une fonctionnalité pour accorder des cartes prépayées à des clients

A l’analyste :

· Une liste des applications les plus rentables

· Une liste des éditeurs qui réalisent le plus de profit et le plus de vente (en quantité)

· La quantité d'installations faites par application ou ressource

· Le profit réalisé par le service distributeur (sachant qu'il prend une part de 30% du prix affiché)

· Le profit réalisé par le service éditeur (sachant qu'il prend une part de 70% du prix affiché)

· Une liste des utilisateurs les plus actifs (en termes de nombre d'avis rendus)

### Début du projet

Le projet a officiellement débuté le Mercredi 25 février 2015, lors du premier TD du semestre de printemps 2015.

### Fin du projet

Le livrable doit être rendu avant la fin du semestre de printemps 2015. Autrement dit, le Mercredi 17 juin 2015.

### Budget du projet

La charge horaire totale du projet est de 80 heures réparties sur 4 personnes, soit une charge de 20h de travail par personne sur le projet.

Des machines fonctionnant sous Linux (Fedora) avec un serveur Unix sont mises à notre disposition par l’organisme client et n’ont donc pas de coût.

Le budget total est nul car le projet à accepté dans le but d’obtenir une récompense intellectuelle, en leur permettant de mettre en pratique et de développer leurs connaissances et compétences.

1. **ACTEURS DU PROJET**

### Maître d’ouvrage

La maîtrise d’ouvrage est dirigée par Stéphane Crozat.

Le contact pour le projet au sein de l’organisme client est Kévin Carpentier.

### Maître d’œuvre

Le maître d’œuvre est le chef de projet. Il s’agit de Thomas Pelletier. Son rôle est d’attribuer les tâches à chaque membre du projet et de définir les dates limites pour leur réalisation. Il assure également le contact avec l’organisme client.

### Compétences nécessaires

|  |  |  |  |
| --- | --- | --- | --- |
| COMPÉTENCES | SERVICE | RESPONSABLE | REMARQUE |
| Gestion de projet |  | Thomas Pelletier | Objectifs, tâches, plannification, contact client |
| Modélisation | Responsable Modélisation | Han Rong | Standard UML |
| Implémentation | Responsable Technique | Etienne Chognard | PostgreSQL et PHP |
| Qualification | Responsable Qualité | Emilie Confais | Tous tests possibles pour déceler toutes les erreurs |

1. **PRODUIT DU PROJET**

Le produit du projet est un site et une base de données qui vont permettre :

· aux clients :

o de télécharger en ligne des applications et des ressources

o de donner leur avis sur les produits

o d’avoir accès aux informations les concernant qui sont stockées dans la base de données

· à l’entreprise :

o de gérer les produits en vente

o de gérer les modes de paiement

o d’analyser les activités effectuées sur la plate-forme

La base de données contiendra donc les informations entrées par l’entreprise et les données obtenues suite à l’utilisation de la plate-forme.

L’interface web permettra d’accéder à la base de données : on pourra la consulter, faire des insertions, des modifications, des suppressions, des mises à jour et récupérer des données.

Les fonctionnalités disponibles pour chaque personne dépendront de son statut :

· Un utilisateur n’aura accès qu’aux informations qui le concernent. Il pourra faire des ajouts et des suppressions au niveau de ses terminaux.

· Un administrateur, comme son nom l’indique, assurera la maintenance de la base de données et aura donc globalement accès à l’ensemble des fonctionnalités.

· Un analyste ne pourra a priori que consulter la base de données mais aura accès à une grande étendue de données.

1. **RISQUES**

Les risques encourus pour le produit final sont :

· Un site pas totalement ne reflétant pas exactement les volontés du client dû à un mauvais choix de modélisation.

· Un site ne contenant pas certaines fonctionnalités que le client espérait implicitement.

· Un modèle conceptuel mal optimisé et donc lourd.

· Un manque d’expérience de l’équipe entraînant des problèmes dans les choix de modélisation et d’implémentation, et donc du retard dans le développement.

1. **CONTRAINTES A RESPECTER**

### Contraintes de délais

Les livrables du projet sont :

· Le MCD, ainsi que la note de clarification et l’organigramme produit, forment le rendu intermédiaire prévu le Mercredi 18 mars 2015.

· Le rapport d’analyse complet, contenant en plus le MLD, doit être livré le Mercredi 8 avril 2015.

Le produit du projet doit être livré le Mercredi 17 juin 2015.

### Contraintes de coûts

Aucune contrainte de coût.

### Contraintes de performances

Le livrable de mi-semestre doit être complet et propre pour permettre à une équipe d’implémenter la solution.

Le produit final doit fonctionner sur les machines des salles informatiques de l’Université de Technologie de Compiègne. Les technologies utilisées doivent être PostgreSQL et PHP.

La solution doit illustrer les principales fonctionnalités demandées par l’organisme client mais la liste des fonctions implémentées ne doit pas nécessairement être exhaustive.

Enfin le système développé se doit d’être performant et d’optimiser la base de données implémentée en veillant à sa sécurité.

# **Organigramme Produit**

## ENVIRONNANTS DU PRODUIT DU PROJET

Les utilisateurs de ce site sont les acheteurs immédiats d’applications et de ressources ou par abonnement, qui utilisent des cartes prépayées ou une carte bancaire, ainsi que les administrateurs et les analystes.

## FONCTIONS

### Utilisateur :

* Connaître les applications compatibles avec ses appareils
* Avoir un historique des achats et des installations
* Gérer ses terminaux (ajout/retrait)

### Administrateur :

* valider des ajouts des applications et des ressources
* gérer des application et des ressources
* Accorder des cartes prépayées à des clients

### Analyste :

* Connaître les applications les plus rentables
* Connaître les éditeurs qui réalisent le plus de profit et le plus de vente (en quantité).
* Connaître le nombre d'installations d'une application/ressources
* Évaluer les profits réalisés par le service distributeur, sachant qu'il prend une part de 30% du prix affiché.
* Évaluer les profits réalisés par le service éditeur, (70% du prix).
* Connaître les utilisateurs les plus actifs (en termes de nombre d'avis rendus)

## SOLUTIONS

|  |  |
| --- | --- |
| Fonction | Solution |
| Utilisateur : | |
| S’identifier | Une page pour entrer son identifiant et son mot de passe et accéder au Nimpstore |
| S’inscrire | Une page pour entrer ses données personnelles pour la création d’un compte utilisateur |
| Se déconnecter | Un bouton pour se déconnecter |
| Modifier un profil | Une page pour mettre à jour son profil ou d’autres profils si on est administrateur |
| Supprimer | Un bouton pour supprimer un compte |
| Sélectionner | Une page pour connaître les applications/ressources compatibles avec au moins un de ses supports |
| Rechercher | Une case de recherche thématique pour proposer toutes les applications/ressources en relation avec les mots de la recherche |
| S’informer | Une page pour voir les informations sur une application ou une ressource |
| Ajouter | Un bouton pour sélectionner une application/ressource et l’ajouter à son compte |
| Consulter | Une page pour voir tous ses achats |
| Retirer | Un bouton pour supprimer une application de son compte |
| Administrateur : | |
| Ajouter une appli/ressource ou ressource | Une page pour ajouter une application et ses informations au site |
| Modifier appli/ressource | Un bouton pour chaque application ou ressource pour accéder à la page de leur définition et la modifier |
| Supprimer | Un bouton pour chaque application pour la supprimer |
| Ajout / Suppression modèle ou OS | Des boutons pour chaque actions : Ajouter un OS, Ajouter un modèle, supprimer un OS, supprimer un modèle |
| Modération compte | Un bouton pour modifier les informations malvenues sur un profil |
| Modération avis | Un bouton pour Modifier ou même supprimer des avis malvenus (racistes, offensants, …) |
| Ajouter des cartes prepayées | Un bouton pour créer une carte prépayée et renseigner ses infos (date de validité / montant) |
| Analyste : | |
| Rechercher sur les applications | Une fonctionnalité de tri par rentabilité |
| Rechercher sur les éditeurs | Une fonctionnalité de tri par profit et par quantité |
| S’informer sur les applications | Un compteur s’incrémentant automatiquement pour chaque application et indiquant le nombre d’installations |
| Évaluer le service distributeur | Un chiffre indiquant le profit du service distributeur |
| Évaluer le service éditeur | Un chiffre indiquant le profit du service éditeur |
| Rechercher sur les utilisateurs | Une fonctionnalité de tri par taux d’activité |

## CLASSEMENT DES SOLUTIONS

### Gestion des profils :

#### *Utilisateur :*

* Une page pour entrer son identifiant et son mot de passe et accéder au Nimpstore
* Une page pour entrer ses données personnelles pour la création d’un compte utilisateur
* Une page pour mettre à jour son profil ou d’autres profils si on est administrateur
* Un bouton pour supprimer un compte

### Gestion des stocks :

#### *Utilisateur :*

* Une page pour connaître les applications compatibles avec ses supports
* Une page pour voir les informations sur une application ou une ressource

#### *Administrateur :*

* Une page pour renseigner une nouvelle application ou ressource
* Une page pour renseigner un nouvel éditeur
* Un bouton pour chaque application ou ressource, et éditeur, pour accéder à la page de leur définition et la modifier
* Un bouton pour chaque application ou ressource pour la supprimer

#### *Analyste :*

* Une fonctionnalité de tri par rentabilité
* Une fonctionnalité de tri par profit et par quantité
* Un chiffre s’incrémentant automatiquement pour chaque application et indiquant le nombre d’installations

Gestion des ventes ou achats :

#### *Utilisateur :*

* Une page pour voir son historique
* Un bouton pour sélectionner une application ou une ressource et l’ajouter à l’un de ses supports
* Un bouton pour supprimer une application de l’un de ses supports

#### *Administrateur :*

* Une page récapitulant toutes les demandes de cartes prépayées des utilisateurs et un bouton en face de chacune d’elles pour l’accepter

### Gestion de l’efficacité de la plate-forme :

#### *Analyste :*

* Un chiffre indiquant le profit du service distributeur
* Un chiffre indiquant le profit du service éditeur
* Une fonctionnalité de tri par taux d’activité

**ORGANIGRAMME PRODUIT**





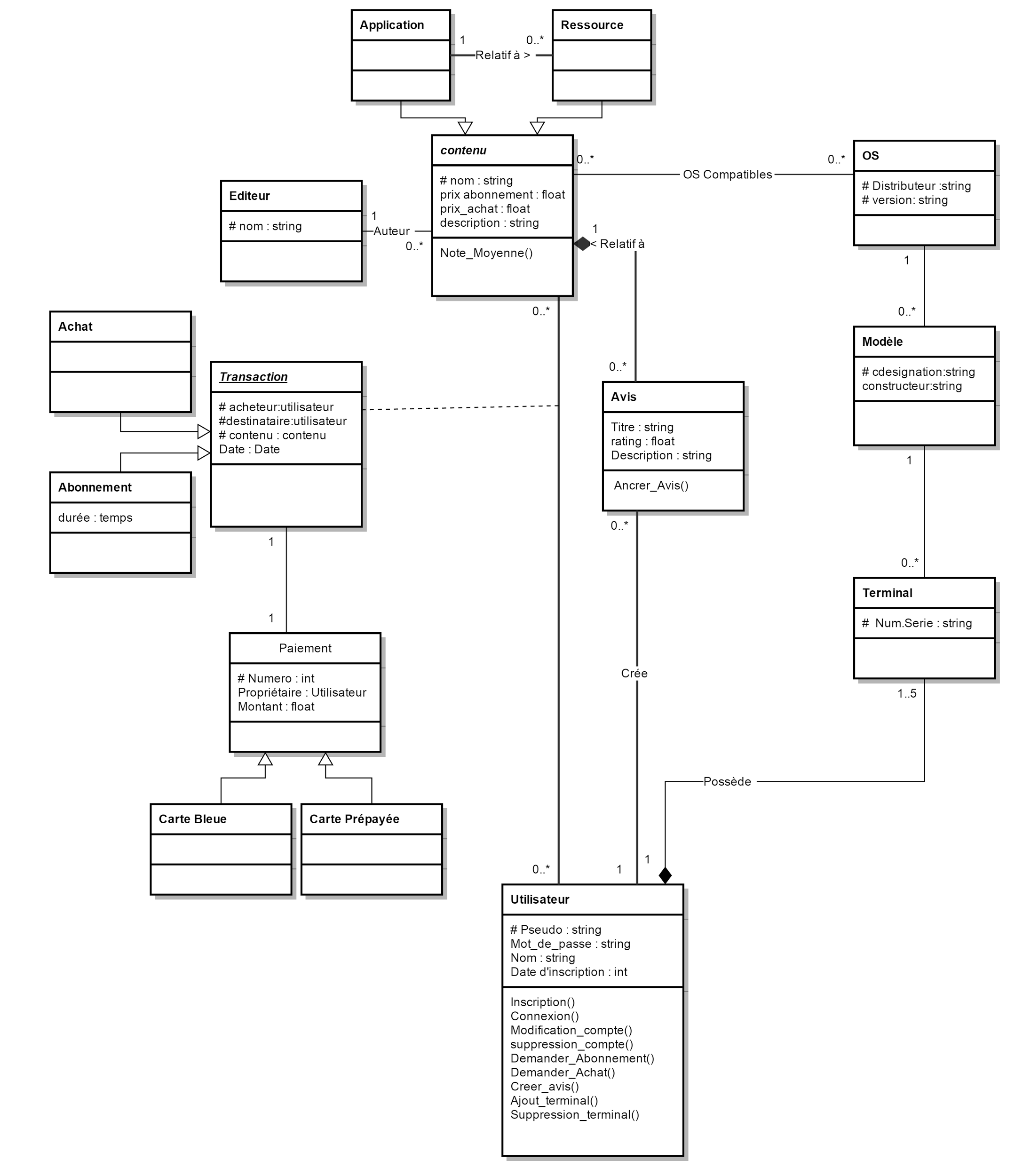


# **Modèle conceptuel de donnée**

Pour plus de lisibilité :

http://bit.ly/1E0ASdQ

ou : https://drive.google.com/file/d/0B-yo7w0fqdL8cUt5SFFCb0NlVkE/view?usp=sharing



**Modèle relationnel**

* Utilisateur(#Pseudo:string, Nom:string, Date\_Inscription:date)
* Terminal(#Num\_Serie:string, Modele=>Modele, utilisateur=>utilisateur)
* Modèle(#Code\_Designation:string,distributeurOS=>OS.distributeur,versionOS=>OS.version,Constructeur:string)
* OS(#Distributeur:string, #version:string)
* Avis\_ressource(#utilisateur=>utilisateur,#ressource=>ressource.nom, Titre:string, Rating:float, Description:string)
* Avis\_application(#utilisateur=>utilisateur,#application=>application.nom, Titre:string, Rating:float, Description:string)
* ModePaiement(#numero:int, type{CB,Prépayée}, montant:float, proprietaire=>Utilisateur)

Tous les attributs non clés des relations précédentes sont NOT NULL

* Ressource(#nom:string, prix abonnement:float, prix\_achat:float, description:string, Appli=>Application,editeur=>editeur)
* Application(#nom:string, prix abonnement:float, prix\_achat:float, description:string, editeur=>editeur)

On peut supposer qu’une application peut etre disponible en achat uniquement ou en abonnement uniquement, les prix peuvent donc etre NULL, ce qui rajoute des contraintes lors de l’achat.

* OS\_Ressource(#distributeurOs=>OS.distributeur,#versionOS=>OS.version, #Ressource=>ressource) OS\_Application(#distributeurOs=>OS.distributeur,#versionOS=>OS.version, #Application=>Application)
* Editeur(#nom:string)

Les relations précédentes étant “toutes clés” elles ne possèdent pas de contraintes particulières (autres que sur les clés)

* Transaction\_Application(#acheteur=>Utilisateur, #contenu=>Application, #destinataire=>Utilisateur, Date:Date, paiement=>ModePaiement,Type={Achat, Abonnement, Duree:int)
* Transaction\_Ressource(#acheteur=>Utilisateur, #contenu=>Ressource, #destinataire=>Utilisateur, Date:Date, paiement=>ModePaiement,Type={Achat, Abonnement, Duree:int)

Pour chaque transaction il faut vérifier que le prix d’achat abonnement est bien renseigné (CF contraintes sur ressource/transaction). Les autres attributs sont NOT NULL, sauf Duree, pour laquelle on choisira une valeur par defaut en cas “d’achat”.

* vCB = Projection(Restriction(ModePaiement, type = 'CB'), numero, montant, proprietaire)
* vCartePrepaye = Projection(Restriction(ModePaiement, type = 'Prépayée'), numero, montant, proprietaire)
* vAbonnement = Projection(Restriction(Jointure(Transaction\_Application, Transaction\_Ressource), type =’Abonnement’),acheteur, contenu, destinataire, date, paiement, Duree))
* vAchat : Projection(Restriction(Jointure(Transaction\_Application, Transaction\_Ressource), type =’Achat’), acheteur, contenu, destinataire, date, paiement))

**Normalisation**

CM : Couverture Minimale

F+ : Fermeture Transitive

* Utilisateur(#Pseudo:string, Nom:string, Date\_Inscription:date)

CM = F+ = {Pseudo -> nom,Date, argent}

Clé:Pseudo

1NF: une clé existe, les attributs sont attributs atomiques

2NF: la clé est atomique, il n’y a donc pas de dépendances partielles

3NF: les attributs non clés sont tous indépendants

* Terminal(#Num\_Serie:string, Modele=>Modele.Code-Designation, utilisateur=>utilisateur.Pseudo)

CM = F+ = {Num\_serie -> Modele, Utilisateur}

1NF: une clé existe, les attributs sont attributs atomiques

2NF: la clé est atomique, il n’y a donc pas de dépendances partielles

3NF: Si un Utilisateur possède deux terminaux du même modèle, il y aura redondance

* Modèle(#Code\_Designation:string, distributeurOS=>OS.distributeur, versionOS=>OS.version, Constructeur:string)

CM = F+ = {Code -> distributeurOS, versionOS, Constructeur}

Clé : Code\_designation

1NF : Hypothèse déduite de l’énoncé : le même modèle aura le même système d’exploitation

Chaque attribut est donc atomique

2NF : Clé atomique

3NF: Aucune relation externe à la clé

* OS(#Distributeur:string, #version:string)

relation toutes clés -> 3NF

* OS\_Ressource(#distributeurOs=>OS.distributeur,#versionOS=>OS.version, #Ressource=>ressource)
* OS\_Application(#distributeurOs=>OS.distributeur,#versionOS=>OS.version, #Application=>Application)

relations toutes clé : 3NF

* Ressource(#nom:string, prix abonnement:float, prix\_achat:float, description:string, Appli=>Application,editeur=>editeur)
* Application(#nom:string, prix abonnement:float, prix\_achat:float, description:string, editeur=>editeur)

Ressource : CM = F+ = {nom->prix\_achat, prix\_abonnement, description, editeur, Application}

Application : CM = F+ = {nom->prix\_achat, prix\_abonnement, description, editeur}

Clé : {nom}

1NF: On suppose qu’une ressource ne peut pas être distribuée sur plusieurs applications.

Donc attributs atomiques, de plus une clé existe.

2NF: La clé est atomique

3NF: On suppose que le prix de l’abonnement est donné pour un mois, et qu’il est indépendant du prix de l’application. Il n’y a donc pas de relation entre les attributs non clé

* Avis\_ressource(#utilisateur=>utilisateur,#ressource=>ressource, Titre:string, Rating:float, Description:string)
* Avis\_application(#utilisateur=>utilisateur,#application=>application, Titre:string, Rating:float, Description:string)

CM = F+ = {utilisateur, ressource} -> titre, rating, description}

1NF: attributs atomiques, clé existante

2NF: les attributs non clés sont défini par la clé en entier.

3NF: il y a pas de relation entre titre/rating/description

* Transaction\_Application(#acheteur=>Utilisateur, #contenu=>Application, #destinataire=>Utilisateur, Date:Date, paiement=>ModePaiement,Type={Achat, Abonnement, Duree:int) Transaction\_Ressource(#acheteur=>Utilisateur, #contenu=>Ressource, #destinataire=>Utilisateur, Date:Date, paiement=>ModePaiement,Type={Achat, Abonnement, Duree:int)

CM = F+ = {utilisateur, contenu, date} ->type, duree}

1NF: clé existante, attributs atomiques (on ne peut pas avoir simultanément un achat et un abonnement, et une transaction ne fait référence qu’à un paiement, même si le paiement peut concerner plusieurs applications d'où transactions)

2NF: les attributs non clés dépendent bien de la clé complète : il faut bien un acheteur, un destinataire et une application concernée pour chaque transaction

3NF : Il y a une relation entre Type et Date : si le type est “achat” alors il faut choisir une valeur d’initialisation de Duree, même si elle n’est évidemment pas utilisée. On peut cependant choisir de l’ignorer dans notre modèle.

* ModePaiement(#numero:int, type{CB,Prépayée}, montant:float, proprietaire=>Utilisateur)

CM = F+ = numero ->type, montant, propriétaire

1NF : pour plus de lisibilité et afin d’éviter la répétitions d’attributs déjà présents dans les tables “transaction”, nous décidons d’utiliser une clef artificielle.

Les attributs présents dans la relation sont de plus atomiques

2NF : la clef est atomique, donc pas de relation non-complète

3NF: il n’y a aucun lien entre les attributs non-clef.

**Conclusion**

Ce projet nous semble ambitieux, en effet, en plus de la gestion des transactions, relativement complexe, il y a de nombreuses méthodes à implémenter, en particulier au niveau de l’administration, qui n’est pas représentée dans notre modélisation.

Cette modélisation nous semble la plus pertinente pour parer à tous les problèmes mis en avant. Le temps dédié à la réalisation du projet nous fait penser que certaines parties ne pourront pas complètement être implémentée. Nous souhaitons bonne chance à l’équipe qui s’occupera de la suite de ce projet.