**Introduction**

Le processus de vente aujourd’hui n’est plus comme auparavant, suite à l’apparition de la notion d’enchère électronique, ou encore appelée enchère en ligne ou enchère virtuelle, les sociétés commerciales réalisent des bénéfices incontournables et visent plus de clientèle en ligne.

Dans ce contexte la société X fait appel à développer une plateforme de vente aux enchères électroniques. Cette plateforme doit être configurable et maintenable et procure un service d’enchère souple et facile à ses utilisateurs.

1. **Spécification des besoins**

Cette section a comme principal objectif de spécifier les différents besoins de la plateforme des enchères électronique souvent nommée PVEE. Un prototype est ensuite généré pour valiser les besoins auprès des utilisateurs.

1. **Etude des besoins**
2. Les besoins fonctionnels

Les besoins fonctionnels viennent définir les fonctionnalités offertes par le système au profil de son utilisateur.

Dans notre cas, 6 principales fonctionnalités existent à savoir :

**Fonctionnalité 1** **: s’inscrire à la plateforme.**

Cette fonctionnalité comprend la gestion des informations relatives aux membres (informations personnelles, transactions réalisées, évaluation (réputation), etc.)

**Fonctionnalité 2 : s’authentifier auprès de la plateforme.**

Fonctionnalité permettant d’identifier si l’utilisateur en cous est un membre.

**Fonctionnalité 3 : consulter la plateforme.**

Cette fonctionnalité doit permettre une consultation facile et conviviale des différentes rubriques des articles, des articles à vendre, des évaluations des membres, des historiques des offres et autres documents d’information.

**Fonctionnalité 4** **: enchérir sur un article.**

La future plateforme PVEE doit offrir la possibilité de la saisie et la validation des enchères, la mise à jour des historiques d’enchères des participants, etc.

**Fonctionnalité 5** **: vendre un article suite à une enchère.**

La future PVEE permet l’inscription des articles à vendre aux enchères et le contrôle du processus de négociation par enchérissement (à l’anglaise ou à l’hollandaise).

**Fonctionnalité 6** **: configurer la plateforme.**

Fonctionnalités permettant au client de configurer les différents paramètres du système (bannières, rubriques, règlements, montant des commissions sur les ventes, etc.)

1. Les besoins non fonctionnels

Nous décrivons dans cette partie les contraintes d’utilisation imposées à la plateforme PVEE.

* La plateforme doit assurer une manipulation facile et claire pour ses utilisateurs.
* La plateforme doit présenter un espace sécurisé pour ses utilisateurs en assurant un processus d’authentification rigoureux.
* La plateforme doit avoir l’aptitude d’être maintenable et configurable.

1. **Modèle de cas d’utilisation**
2. Identification des acteurs

Un acteur est une entité externe au système et qui agit sur ce dernier en consultant et/ou modifiant son état. Dans notre application nous distinguons les acteurs suivants :

|  |  |
| --- | --- |
| **Acteur** | **Rôle** |
| Utilisateur | Est toute personne accédant à la plateforme PVEE sans inscription et donc sans authentification. |
| Membre | C’est toute personne inscrite et pouvant faire une authentification correcte à la plateforme. |
| Technicien | C’est une personne responsable de la configuration et la maintenance de la plateforme PVEE. |

1. Identification des cas d’utilisation

Les cas d’utilisation décrivent sous la forme d’actions et de réactions le comportement d’un système du point de vue d’un utilisateur. Un cas d’utilisation est une manière spécifique d’utiliser un système. C’est l’image d’une fonctionnalité déclenchée en réponse à la simulation d’un acteur externe.

Pour mieux détailler les cas d’utilisation, UML définit trois types de relations standards

* La relation d’inclusion **(include)** a un caractère obligatoire, le cas d’utilisation source nécessite la réalisation du cas d’utilisation cible.
* La relation d’extension **(extend)** signifie que le comportement du cas d’utilisation source peut compléter le comportement du cas d’utilisation cible d’où le cas d’utilisation source n’est pas toujours nécessaire mais qu’il peut l’être dans certaines situations et dans ce cas de figure nous devons préciser une condition que l’on appelle point d’extension.
* La relation de généralisation/spécification où les cas d’utilisation descendants héritent de la description de leurs parents communs.

Les principaux cas d’utilisation de notre système PVEE mis en évidence par l’expression de besoins préliminaire sont :

* S’inscrire dans la PVEE
* S’authentifier
* Consulter les pages d’articles et les conditions d’utilisation
* Consulter l’historique des offres
* Enchérir sur un article
* Vendre un article
* Configurer la PVEE

1. Diagramme de cas d’utilisation système

La figure 1 suivante décrit les différents acteurs avec les principales fonctionnalités.

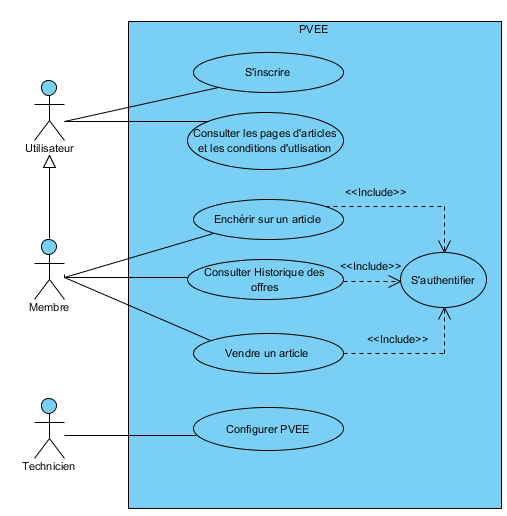


Figure 1 : diagramme de cas d’utilisation système.

1. Description des cas d’utilisation

**Description textuelle**

La fiche de description textuelle d’un cas d’utilisation n’est pas normalisée par UML nous procédons de suivre la structuration proposée par Pascal Roques.

* **S’inscrire**

1. **Sommaire d’identification**

* ***Titre :*** s’inscrire
* ***Résumé :*** ce cas d’utilisation permet à un utilisateur de s’inscrire auprès de la plateforme et devenir un membre ; donc être autorisé à enchérir, à vendre à travers la plateforme et à consulter l’historique des offres.

1. **Préconditions :**

* L’utilisateur doit disposer d’un navigateur
* Une connexion à internet est établie

1. **Scénario nominal**

|  |
| --- |
| 1. L’utilisateur saisit les renseignements nécessaires y compris un identifiant et un mot de passe. 2. Le système vérifie les données saisies et affiche un message de confirmation 3. Le système envoie un e-mail à l’utilisateur. 4. L’utilisateur confirme en cliquant sur l’hyperlien correspondant. 5. Le système enregistre les données de l’utilisateur et crée un compte membre |

1. **Enchaînements alternatifs :**

Nous distinguons ici les enchaînements alternatifs notés (Ax) qui représentent une étape du scénario nominal des enchainements d’erreur notés (Ey) qui terminent brutalement le cas d’utilisation en échec. L’objectif de l’acteur principal est donc atteint par les scénarios alternatifs et nominaux mais pas par ceux d’erreur.

A1 : Les données saisies par l’utilisateur sont erronées

L’enchainement A1 démarre au point 1 du scénario nominal et continue comme suit :

2- Le système vérifie les données saisies et signale à

l’utilisateur les éventuelles erreurs.

3- L’utilisateur ressaisit les champs signalés comme

Erronés.

Le scénario nominal reprend au point 2.

A2 : L’utilisateur choisit l’annulation de l’inscription ou délai de confirmation d’inscription est expiré.

L’enchainement A2 démarre au point 3 du scénario nominal et continue comme suit :

4- L’utilisateur annule en cliquant sur l’hyperlien

d’annulation.

5- Le système annule l’inscription

A3 : Le délai de confirmation d’inscription est expiré.

L’enchainement A3 démarre au point 3 du scénario nominal et continue comme suit :

4- L’utilisateur ne clique aucun hyperlien ni de

confirmation ni d’annulation

5- Le système annule l’inscription

1. **Post conditions :** Un compte membre est crée
2. **Besoins d’IHM :**

Afin de mener à bien le cas d’utilisation s’inscrire il nous faut une interface d’inscription où :

* La disposition des données à saisir soit le plus claire et le plus ergonomique que possible.
* Le contrôle de saisie est bien étudié.
* **S’authentifier**

1. **Sommaire d’identification**

* ***Titre :*** s’authentifier
* ***Résumé :*** ce cas d’utilisation permet à un membre déjà inscrit de s’authentifier auprès de la plateforme PVEE.

1. **Préconditions :**

* Le membre doit disposer d’un login et d’un mot de passe enregistrés dans la base de la PVEE.

1. **Scénario nominal**

|  |
| --- |
| 1. Le membre saisit son login et son mot de passe. 2. Le système vérifie les paramètres saisis et valide l’authentification du membre. |

1. **Enchaînements alternatifs :**

A1 : Les données saisies par le membre sont erronées

L’enchainement A1 démarre au point 1 du scénario nominal et continue comme suit :

2- Le système vérifie les données saisies et signale au

membre une erreur dans le login ou/et le mot de passe.

3- Le membre ressaisit les champs signalés comme

Erronés.

Le scénario nominal reprend au point 2.

1. **Post conditions :** Un membre est identifié et autorisé de vendre et d’enchérie sur la plateforme PVEE.
2. **Besoins d’IHM :**

L’interface d’authentification doit être apparente et concise avec un contrôle rigoureux et capable de se recharger après chaque tentative échouée.

**Description par le diagramme de séquence système**

Un diagramme de séquence système représente graphiquement la chronologie des interactions entre les acteurs et le système vu comme une boîte noire, dans le cadre du scénario nominal.

* **Enchérir sur un article**

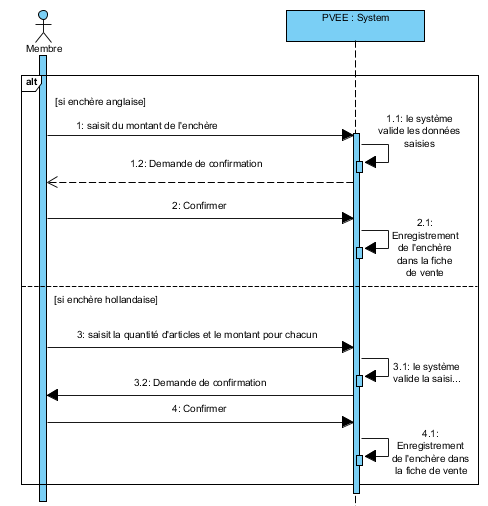
****

Figure 2 : Diagramme de séquence système du scénario nominal Enchérir sur un article

* **Vendre un article**

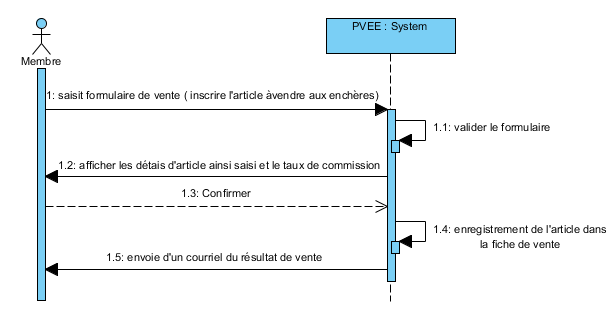
****

Figure 3 : Diagramme de séquence système du scénario nominal Vendre un article

1. **Prototype**

Un prototype est un modèle intermédiaire du livrable logiciel. Il s’agit d’une version simplifiée d’une partie de l’application finale. La méthode de prototypage consiste à effectuer un test pour obtenir rapidement et à moindre cout une idée du bon ou mauvais fonctionnement de l’application

En outre en adoptant UP comme méthode de conception l’artéfact prototype nous servons pour diminuer le risque majeur de l’inutilité celui de développer des fonctionnalités non adéquates avec les besoins.

A travers un prototype nous pouvons ainsi juger l’ergonomie de l’application.

Nous procédons, alors, d’élaborer un prototype de la page principale de la plateforme où nous trouvons les fonctionnalités principales exposées au profil de l’utilisateur.

Nous avons fait appel à Lumzy ; un outil en ligne gratuit de prototypage d’où l’interface suivante :

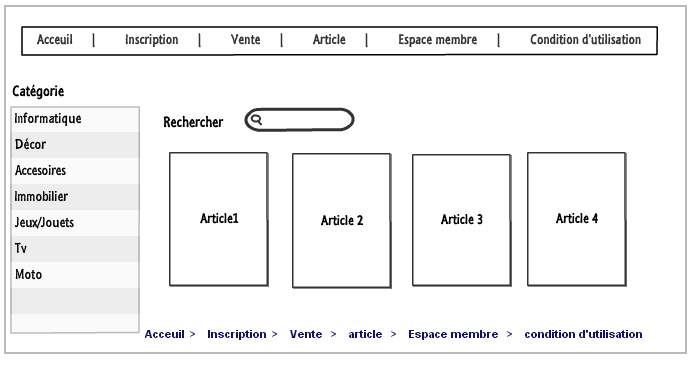


Figure 4 : Interface d’accueil

La phase de spécification des besoins exprime des besoins affinés de l’utilisateur afin de passer à l’étape de conception. Elle peut être envisagée comme une première ébauche du modèle de conception et constitue ainsi une base aidant à la phase de construction où la conception devient plus raffinée.

1. **Conception**

En partant des modèles de la spécification des besoins nous procédons dans cette deuxième partie à identifier les classes participantes, à déterminer l’architecture logique et physique de notre système PVEE et à décrire le dynamisme de certaines fonctionnalités que nous les jugeons intéressantes.

1. **Conception préliminaire**

L’objectif de la conception préliminaire est d’identifier les classes participantes du système et leurs attribuer des responsabilités selon leurs comportements.

1. Diagramme de classe de domaine

Le modèle de domaine est la première version du diagramme de classe sans attributions des responsabilités pour chaque classe. Il décrit les entités du monde réel et leurs relations. Il englobe les entités les plus importants dans le contexte métier du système.

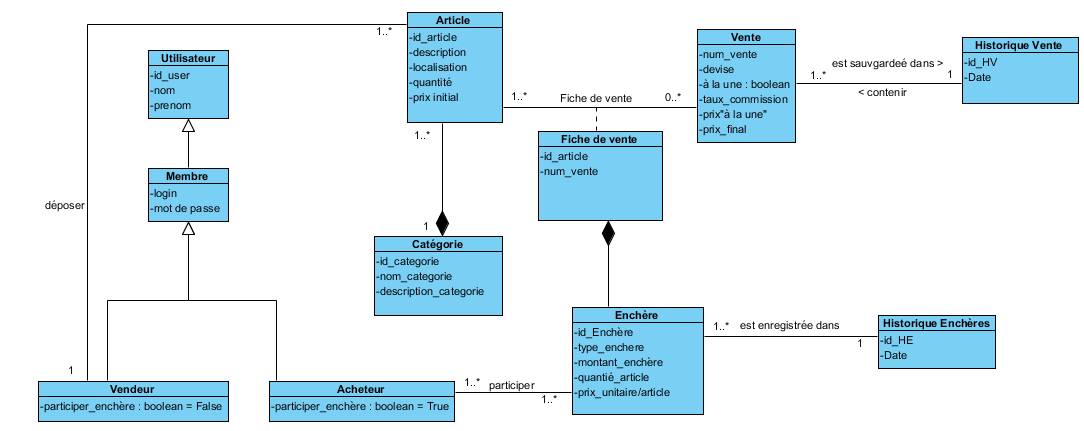


Figure 5 : Diagramme de classe de domaine

Dans ce type de diagramme de classe nous représentons seulement les attributs comme détails dans la classe.

1. Diagramme de séquence

Les diagrammes de séquence sont particulièrement utiles au concepteur pour représenter graphiquement ses décisions d’allocations de responsabilités. Chaque diagramme représente un ensemble d’objets de classes différentes collaborant dans le cadre d’un scénario d’exécution du système. Dans ce genre de diagramme, les objets communiquent en s’envoyant des messages sui invoquent des opérations sur les objets récepteurs.

Par rapport aux diagrammes de séquence système, nous remplaçons le système vu comme une boîte noire par un ensemble d’objets communiquant afin d’enrichir le diagramme de classe.

Notre démarche étant de procéder par cas d’utilisation pour dégager les différents types d’objets qui peuvent figurer dans un diagramme de séquence à savoir :

* **Les interfaces** qui représentent les écrans de dialogue ou d’interaction avec le système (IHM).
* **Les contrôles** qui contiennent la logique applicative
* **Les entités** qui sont les objets métier provenant du modèle de classe de domaine.

1. **Diagramme de séquence du cas d’utilisation s’inscrire**

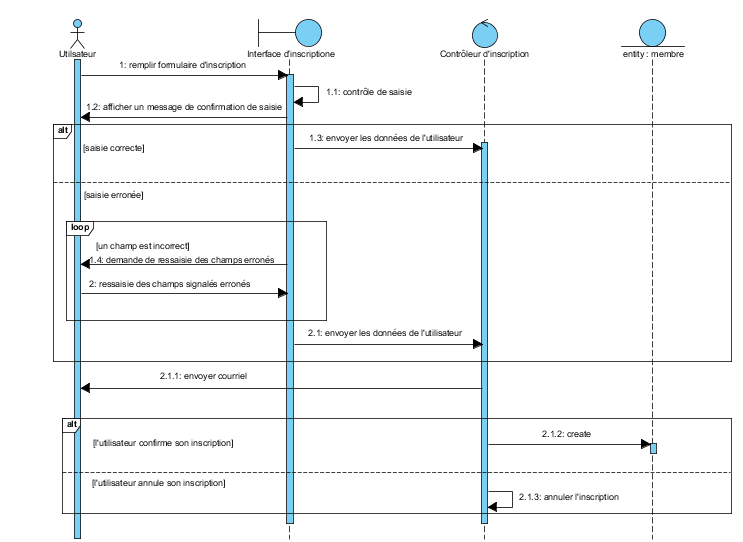
****

Figure 6 : Diagramme de séquence de cas d’utilisation s’inscrire

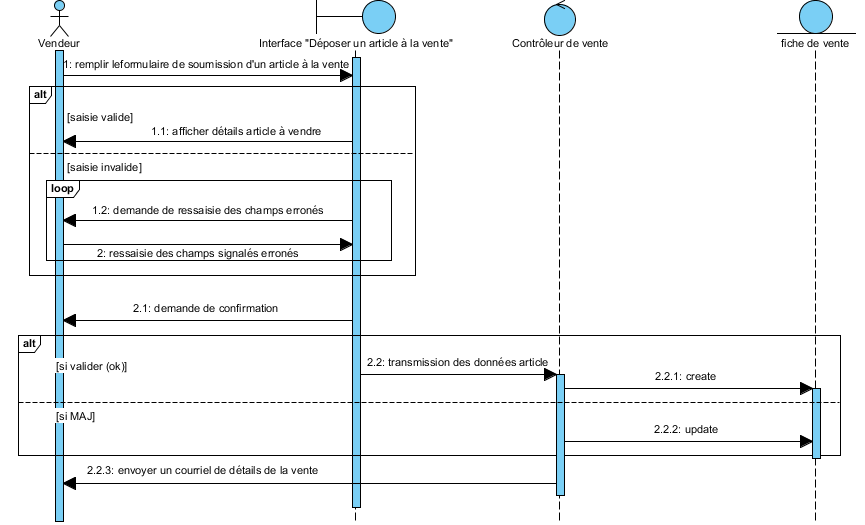


Figure 7 : Diagramme de séquence de cas d’utilisation vendre un article

1. Diagramme de déploiement

Le diagramme de déploiement représente l’agencement physique d’un système, montrant sur quels composants matériels les différents composants logiciels s’exécutent. Les ressources matérielles sont représentées sous forme de nœuds et les connexions ou support de communication sous forme de lien entre les nœuds.

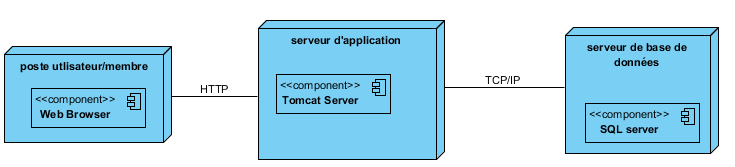


Figure8 : Diagramme de déploiement

1. Architecture logique de l’application

L’architecture logique du système développé est donnée par les trois couches suivantes :

* La couche présentation contient l’interface homme-machine de la plateforme PVEE.
* La couche applicative constituée des classes contrôleurs.
* La couche de données correspondant aux classes entités ou classes de domaine.

1. Diagramme de classe de l’application

En partant du modèle de classe de domaine, nous allons affiner et compléter le diagramme de classes participantes. Pour cela nous utilisons les diagrammes de séquence que nous venons de réaliser pour :

* Ajouter ou préciser les opérations dans les classes (un message ne peut être reçu par un objet que si sa classe a déclaré l’opération publique.
* Ajouter des types aux attributs et aux paramètres et retours des opérations.
* Affiner les relations entre classes : association avec indications de navigabilité, de rôle, généralisations ou dépendances.

Nous avons ajouté :

* les classes interface tel que : interface Inscription, interface d’authentification, interface ErreurAuthentification, interface VenteEnchère, interfaceEnchère, interface HistoriqueEnchère, interface HistoriqueVente, interface consultationArticle, interfaceConfiguration.
* Les classes contrôleurs tel que : ContrôleurEnchère, ContrôleurVente, ContrôleurAuthentification, ContrôleurInscription, ContrôleurConfiguration.
* Les classes entités sont ceux de diagramme de classe de domaine.

**A FAIRE APRES LES DS**

1. Regroupement par package

Le diagramme de packages permet de décomposer le système en parties plus facilement observables, appelés « packages ».  Il permet d’obtenir une image sur les dépendances d’où de les maitriser.

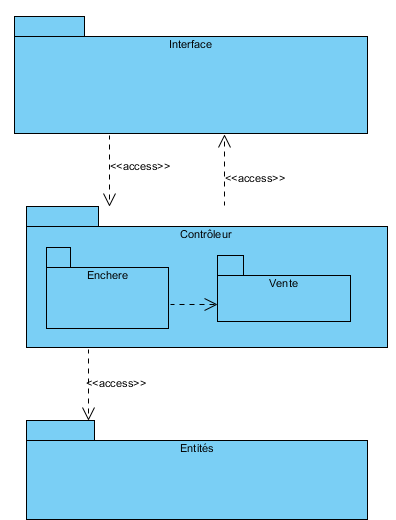


Figure 10 : diagramme de package

1. **Conception détaillée**
2. Description des cas d’utilisation

Dans cette section nous allons décrire la cinématique des cas d’utilisation à l’aide des diagrammes d’activités

1. Conception des classes