

# INSEA

## RAPPORT DU PROJET Modélisation Orientée Objet

**Sujet :** Service de l'Agence Nationale de la Conservation Foncière, du Cadastre et de la Cartographie

— DOUIH Zakaria  
— BELMELLAH Mohammed  
— NAIT-EL-HAJ Abderrahmane  
— FAHIMI Yassir

**Filière :** DSE  
**Encadré par :** Mme. SAIDI Rajaa

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>1 Etude préliminaire</b>	<b>4</b>
1.1 Scénario nominal d'interaction avec l'ANCFCC . . . . .	4
1.2 Identification des modules préliminaires . . . . .	5
1.3 Modélisation du contexte . . . . .	5
1.3.1 Identification des acteurs . . . . .	5
1.3.2 Messages et modèle du contexte dynamique . . . . .	6
<b>2 Capture des besoins fonctionnels</b>	<b>8</b>
2.1 Identification des cas d'utilisation . . . . .	8
2.2 Identification des classes candidates . . . . .	9
<b>3 Capture des besoins techniques</b>	<b>10</b>
3.1 Architecture matérielle . . . . .	10
3.2 Organisation logicielle . . . . .	10
3.3 Cas d'utilisation techniques . . . . .	11
<b>4 Analyse</b>	<b>12</b>
4.1 Découpage en catégories . . . . .	12
4.2 Modèle statique . . . . .	12
4.3 Modèle dynamique . . . . .	13
<b>5 Conception générique</b>	<b>15</b>
<b>6 Conception préliminaire</b>	<b>17</b>
<b>7 Conception détaillée</b>	<b>20</b>
<b>Conclusion</b>	<b>22</b>

# List des abbréviations

**2TUP** 2 Track Unified Process. 3

**ANCFCC** Agence Nationale de la Conservation Foncière, du Cadastre et de la Cartographie. 1, 3–6, 17, 22

**MLD** Modèle Logique de Données. 20, 21

**UML** Unified Modeling Language. 3

# Introduction

L'Agence Nationale de la Conservation Foncière, du Cadastre et de la Cartographie (ANCFCC) est un établissement public marocain chargé de la gestion, de la sécurisation et de la mise à jour des informations foncières, cadastrales et cartographiques du Royaume.

Le site web de l'Agence Nationale de la Conservation Foncière, du Cadastre et de la Cartographie [ancfcc.gov.ma](http://ancfcc.gov.ma) propose une large gamme de services électroniques destinés à simplifier les démarches foncières pour les citoyens et les professionnels.

Pour une meilleure organisation de notre travail, nous commençons par une étude préliminaire du besoin en faisant des parallèles avec le contenu du site web. Après nous allons entamer notre modélisation du système en suivant le processus 2TUP[1] et en se basant sur le langage UML 2.0.

# Chapitre 1

## Etude préliminaire

Dans cette partie nous voulons travailler sur l'élaboration d'un cahier de charges préliminaire, et sur la modélisation du contexte. Il s'agit d'une détermination des frontières du système au sein du métier. De ce que nous allons avoir, nous préparons une liste préliminaire des acteurs qui interagissent avec le système.

### 1.1 Scénario nominal d'interaction avec l'ANCFCC

Puisque l'application existe déjà, nous allons juste analyser le rôle qu'elle joue au milieu des interactions entre l'agence et les bénéficiaires (retro-ingénierie).

Lors de l'achat d'un bien immobilier, plusieurs étapes doivent être suivies avec précision pour garantir la sécurité juridique et éviter des problèmes financiers. Voici une explication détaillée et complète pour chaque étape, intégrant la vérification des documents et les démarches administratives :

- **Recherche initiale du bien** : la personne concernée visite le bien qu'elle veut acheter (maison, appartement, terrain, local commercial, etc) et elle le vérifie pour évaluer son état et son emplacement. Elle compare son prix avec les données du **référentiel commun des prix immobiliers** pour vérifier s'il est raisonnable. Le système doit donc fournir ce référentiel ainsi qu'un service d'achat des **cartes topographiques**.
- **Vérification des documents de propriété** : elle demande par la suite un **extrait de propriété** soit au propriétaire actuel ou à l'ANCFCC.  
Cet extrait inclut le nom du propriétaire, le numéro du titre foncier et la situation juridique du bien (hypothèque, litiges, etc.)
- **Vérification du plan cadastral** : elle demande par la suite le **plan cadastral** auprès de l'agence.
- **Préparation de l'accord de vente** : elle signe un compromis de vente avec l'acheteur (compris le prix de vente, les modalités de paiement et la date de transfert du bien).
- **Publicité foncière** : elle enregistre le compromis de vente au registre foncier.
- **Paiement et vérification des fonds** : elle verse un acompte pour le vendeur lors de la signature du compromis de vente. Elle effectue les paiements via des moyens sécurisés (chèque certifié, virement bancaire).
- **Préparation de l'acte de vente définitif** : Un notaire rédige l'acte de vente final après avoir vérifié tous les documents.  
Le notaire :
  - Vérifie la légalité des documents.
  - Enregistre la transaction au registre foncier.
  - S'assure que toutes les taxes (enregistrement, conservation foncière) sont payées

- **Transfert officiel de propriété** : une fois l'acte final signé, le bien est inscrit à son nom au registre foncier.  
Elle reçoit un nouvel extrait de propriété attestant qu'elle est le propriétaire légal.
- **Vérification de la superficie et des caractéristiques du bien** : après l'acquisition, il est recommandé de recourir à un ingénieur topographique pour mesurer la superficie et vérifier que le bien correspond exactement à ce qui est indiqué dans les documents.

## 1.2 Identification des modules préliminaires

Dans cette section, nous identifions les modules fonctionnels préliminaires du site web de l'ANCFCC. Ces modules sont organisés en fonction des services qu'ils proposent pour répondre aux besoins des différents utilisateurs.

- **Consultation des informations foncières** : Ce module permet aux utilisateurs d'accéder aux données relatives aux propriétés foncières. Les services incluent :
  - **Cartographie** : Fournit une visualisation interactive des propriétés foncières sur une carte numérique.
  - **Publicité foncière** : Offre la possibilité de consulter les annonces et informations légales sur les biens fonciers.
  - **Référentiel commun des prix** : Permet d'accéder aux prix de référence des terrains et propriétés dans différentes zones géographiques.
- **Dépôt et suivi des documents** : Ce module facilite le dépôt en ligne et le suivi des documents soumis pour différents services.
  - **Plan cadastral** : Permet de soumettre et de consulter les plans cadastraux.
  - **Certificat de propriété** : Offre la possibilité de demander et de suivre l'état des certificats de propriété.
  - **Calcul de contenance** : Intègre un outil interactif pour le calcul des superficies des terrains.
- **Paiement des frais** : Ce module est dédié au paiement en ligne des frais associés à différents services, comme la demande de duplicata ou le certificat de propriété.
- **Authenticité des documents** : Fournit un outil pour vérifier en ligne l'authenticité des documents émis par l'ANCFCC.
- **Services pour les professionnels** : Propose des outils spécialisés pour les professionnels, tels que les notaires et les topographes.
- **Mohafadati** : Plateforme personnalisée permettant aux citoyens de suivre leurs biens fonciers et d'accéder à des services dédiés.
- **Notifications et assistance** :
  - **Notifications** : Envoi d'alertes aux utilisateurs pour le suivi de leurs demandes ou pour les informer des étapes suivantes.
  - **Assistance en ligne** : Fournit une aide interactive ou via chat pour répondre aux questions des utilisateurs.
- **Services divers** : Ce module regroupe d'autres services relatifs à la gestion foncière, notamment :
  - **Demande de duplicata du titre foncier.**
  - **Demande de copie d'un document.**
  - **Demande de bornage ou reprise de bornage.**

## 1.3 Modélisation du contexte

### 1.3.1 Identification des acteurs

Cette section identifie les principaux acteurs du système ainsi que leurs rôles dans l'utilisation des modules définis.

- **Client** :
  - Consulter les informations foncières.
  - Soumettre des demandes (duplicata, certificat de propriété, bornage).
  - Vérifier l'authenticité des documents.
  - Payer les frais en ligne pour certains services.
- **Professionnel** :

- **Notaire** : Consultation des titres fonciers, vérification d'authenticité, et soumission de documents légaux.
  - **Topographe** : Accès aux plans cadastraux, calcul de contenance, et demandes liées au bornage.
  - **Autorité de validation de documents/Employé** :
    - Validation des documents soumis par les utilisateurs.
    - Gestion des services en ligne et des bases de données.
    - Administration du système pour garantir son bon fonctionnement.
- N.B.** Nous n'avons pas intégré l'acteur secondaire "système bancaire" pour une simplicité du modèle. Nous n'avons considéré que le système serveur/client du site web.

### 1.3.2 Messages et modèle du contexte dynamique

#### Messages échangés

Les messages décrivent les interactions entre les différents acteurs (utilisateurs et employés) et le système ANCFCC. Ils incluent les flux d'information et les actions déclenchées.

- **Demande de consultation** :
  - *Acteurs concernés* : Citoyen, Professionnel.
  - *Description* : Les utilisateurs envoient une requête pour consulter des informations foncières, telles que des données cartographiques, des publicités foncières ou des références de prix.
  - *Résultat attendu* : Réponse du système avec les données demandées.
- **Soumission de documents** :
  - *Acteurs concernés* : Citoyen, Professionnel.
  - *Description* : Envoi de documents pour effectuer des demandes spécifiques (duplicata, certificat de propriété, bornage).
  - *Résultat attendu* : Confirmation de réception par le système et suivi de l'état de traitement.
- **Notification d'état** :
  - *Acteurs concernés* : Tous les utilisateurs.
  - *Description* : Le système informe les utilisateurs des changements d'état de leurs demandes (par exemple : "En cours", "Validé", "Rejeté").
  - *Résultat attendu* : Message reçu par l'utilisateur via email ou interface du site.
- **Vérification de l'authenticité** :
  - *Acteurs concernés* : Citoyen, Professionnel.
  - *Description* : Les utilisateurs soumettent un document pour en vérifier l'authenticité.
  - *Résultat attendu* : Réponse immédiate du système indiquant si le document est authentique ou non.
- **Paiement en ligne** :
  - *Acteurs concernés* : Tous les utilisateurs.
  - *Description* : Les utilisateurs effectuent des paiements pour finaliser leurs demandes ou accéder à certains services.
  - *Résultat attendu* : Confirmation du paiement et enregistrement dans le système.
- **Demande de support** :
  - *Acteurs concernés* : Tous les utilisateurs.
  - *Description* : Envoi d'une requête au module d'assistance pour obtenir des informations ou résoudre des problèmes.
  - *Résultat attendu* : Réponse du support utilisateur ou redirection vers des ressources utiles.
- **Récupération des demandes en attente** :
  - *Description* : L'employé consulte la liste des demandes soumises par les utilisateurs qui nécessitent une validation ou une action.
  - *Acteurs concernés* : Employé → Système.
  - *Résultat attendu* : L'employé obtient la liste des demandes en attente de validation ou d'action.
- **Validation ou rejet des documents soumis** :
  - *Description* : L'employé valide ou rejette un document après vérification des informations fournies par l'utilisateur.
  - *Acteurs concernés* : Employé → Système.
  - *Résultat attendu* : Le système prend en compte la validation ou le rejet du document par l'employé.
- **Mise à jour du statut des demandes** :

- Description : L'employé modifie l'état d'une demande (exemple : "En cours de traitement", "Validée", ou "Rejetée").
- Acteurs concernés : Employé → Système.
- Résultat attendu : Le statut de la demande est mis à jour dans le système.
- **Notification aux utilisateurs :**
  - Description : Le système envoie une notification automatique à l'utilisateur lorsque l'employé change le statut d'une demande ou ajoute des commentaires.
  - Acteurs concernés : Système → Utilisateur (indirectement initié par l'employé).
  - Résultat attendu : L'utilisateur reçoit une notification automatique sur le changement de statut ou de commentaire.
- **Consultation de l'historique des actions :**
  - Description : L'employé consulte les logs des actions effectuées sur les demandes pour assurer un suivi administratif ou résoudre des problèmes.
  - Acteurs concernés : Employé → Système.
  - Résultat attendu : L'employé peut visualiser l'historique des actions effectuées sur les demandes.
- **Gestion des erreurs ou incohérences :**
  - Description : L'employé signale ou corrige des incohérences dans les documents soumis ou dans le système.
  - Acteurs concernés : Employé → Système.
  - Résultat attendu : Les incohérences sont corrigées ou signalées dans le système.

### Modèle du contexte dynamique

Pour une meilleur organisation acteurs/messages, voici une présentation de **quelques** messages en utilisant un diagramme de communication :

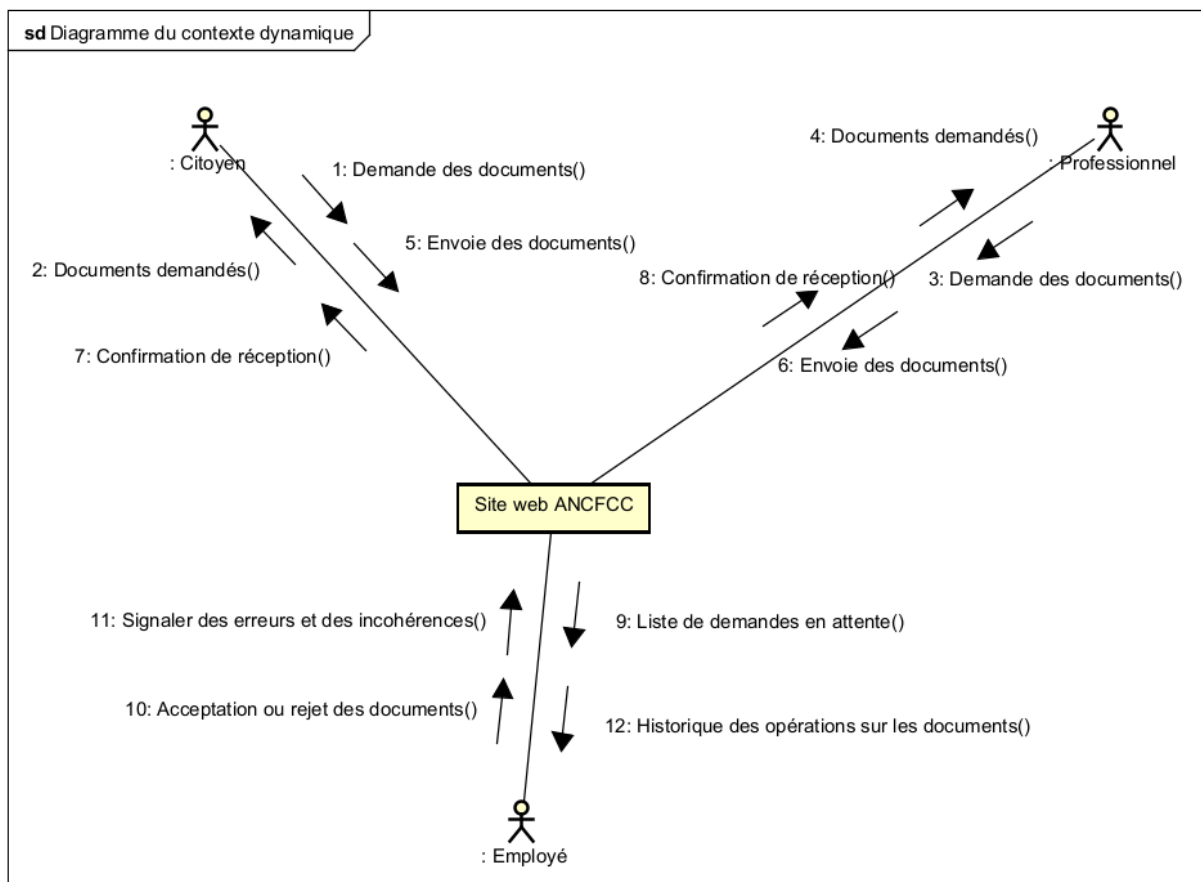


FIGURE 1.1 – Quelques messages entre acteurs et système (non exhaustif)



## Chapitre 2

# Capture des besoins fonctionnels

### 2.1 Identification des cas d'utilisation

Voici un aperçus des cas d'utilisations les plus pertinentes :

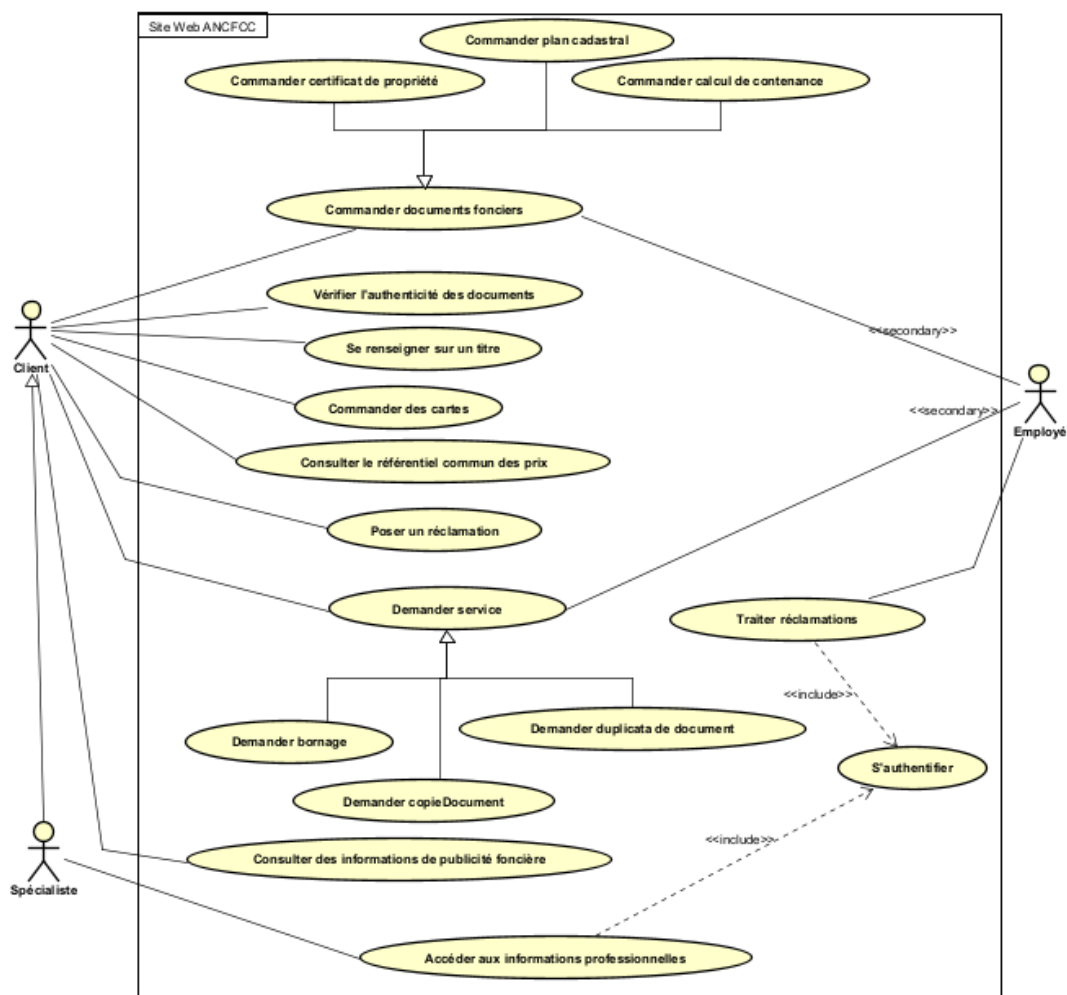


FIGURE 2.1 – Cas d'utilisation préliminaire

## 2.2 Identification des classes candidates

Voici un tableau des classes identifiées à lors de la phase préliminaire avec les principales responsabilités pour chaque une :

Classe	Responsabilité
<i>Client</i>	<ul style="list-style-type: none"><li>— Elle doit savoir son nom complet, son adresse, son courrier, etc.</li><li>— Elle doit savoir les immobiliers dont elle est propriétaire</li></ul>
<i>Immobilier</i>	<ul style="list-style-type: none"><li>— Elle doit savoir son propriétaire</li><li>— Elle doit savoir le type de son titre</li></ul>
<i>Certificat de propriété</i>	<ul style="list-style-type: none"><li>— Elle doit savoir à quel client et à quelle immobilisation elle y est associée</li></ul>
<i>Calcul Contenance</i>	<ul style="list-style-type: none"><li>— Elle doit savoir à quel immobilier elle y est associée</li></ul>
<i>Plan Cadastral</i>	<ul style="list-style-type: none"><li>— Elle doit savoir à quel immobilier elle y est associée</li></ul>
<i>Cartographie</i>	<ul style="list-style-type: none"><li>— Elle doit savoir son échelle et le nom de la carte y elle est associée</li></ul>
<i>Commande</i>	<ul style="list-style-type: none"><li>— Elle doit savoir à quel client et à quel produit elle concerne</li></ul>
<i>Produit</i>	<ul style="list-style-type: none"><li>— Elle doit savoir son nom, quantité, prix et la description</li><li>— Elle doit être capable de calculer le prix total</li></ul>
<i>Réclamation</i>	<ul style="list-style-type: none"><li>— Elle doit savoir l'immobilier elle y est associée</li></ul>
<i>Bornage</i>	<ul style="list-style-type: none"><li>— Elle doit savoir le client elle y est associée</li><li>— Elle doit savoir l'employé qui en est chargé</li></ul>
<i>CopieDocument</i>	<ul style="list-style-type: none"><li>— Les mêmes responsabilités</li></ul>
<i>DuplicataDocument</i>	<ul style="list-style-type: none"><li>— Les mêmes responsabilités</li></ul>

TABLE 2.1 – Classes candidates et leurs responsabilités

## Chapitre 3

# Capture des besoins techniques

### 3.1 Architecture matérielle

Il s'agit d'une architecture 3-tiers (Client, serveur, Base de données). Voici le diagramme de déploiement proposé :

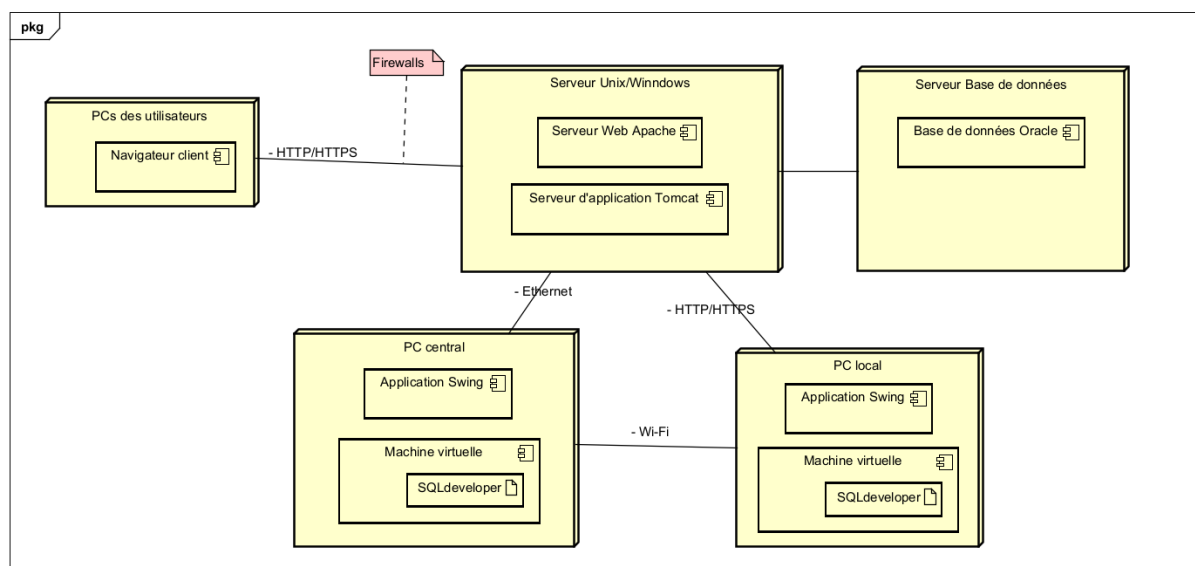


FIGURE 3.1 – Configuration matérielle proposée pour le système

### 3.2 Organisation logicielle

L'architecture proposée du logiciel s'agit d'une architecture en niveaux tel que chaque composant d'un niveau délègue un composant dans un autre niveau pour réaliser une fonctionnalité technique. Voici l'architecture en niveaux proposée :

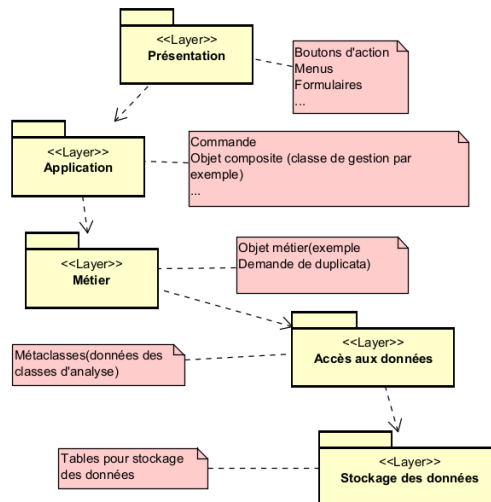


FIGURE 3.2 – Organisation logicielle

### 3.3 Cas d'utilisation techniques

Le cas d'utilisation général peut être décomposé en sous diagrammes de cas d'utilisation mais en niveau d'encapsulation plus profond. Nous ne voulons pas rentrer dans les détails de la délégations des fonctionnalités entre couches logicielles. Voici un aperçu général du cas d'utilisation technique :

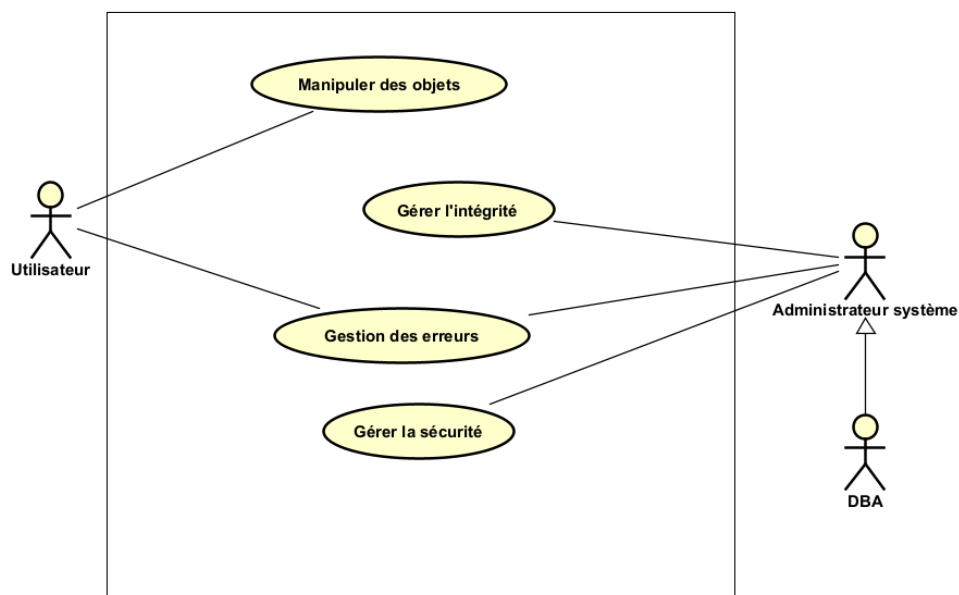


FIGURE 3.3 – Diagramme des cas d'utilisation techniques

# Chapitre 4

## Analyse

### 4.1 Découpage en catégories

Dans cette partie nous proposons de diviser les classes candidates en catégories selon deux critères :

- Similarité des fonctionnalités et interactions avec les autres classes (par exemple catégories Services et Commandes)
- Cohérence entre les classes de la même catégorie (par exemple catégorie Clients)

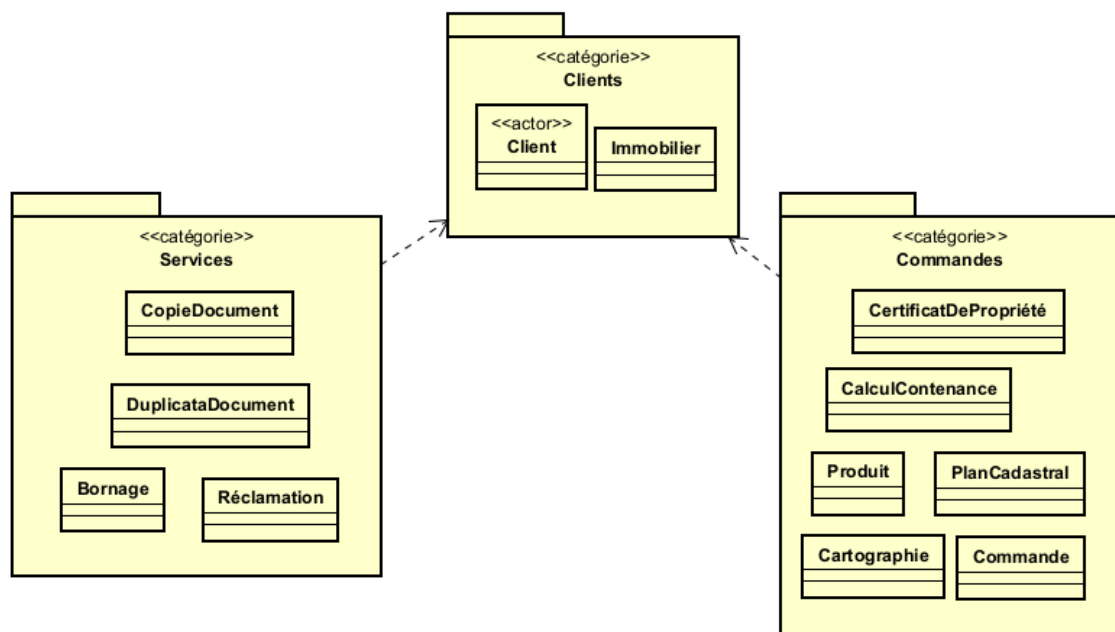


FIGURE 4.1 – Découpage en catégories

### 4.2 Modèle statique

Nous proposons ici un diagramme de classes préliminaire :

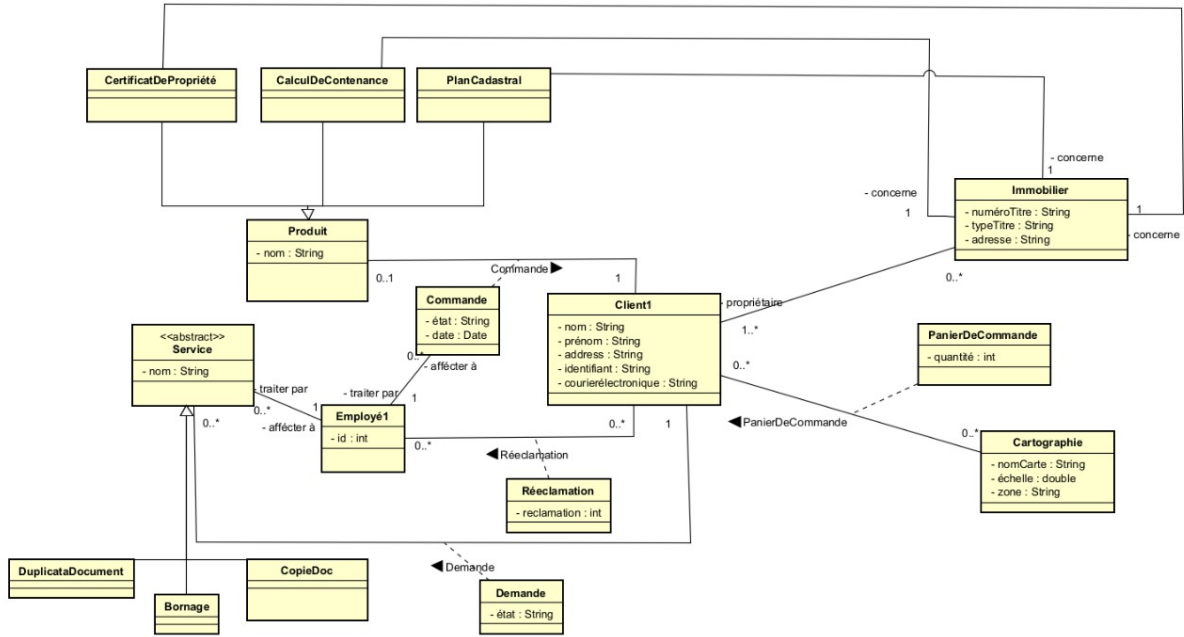


FIGURE 4.2 – Diagramme de classes préliminaire

Les multiplicités sont à changer par la suite. Les attributs et les opérations sont à ajouter et des classes toutes entières sont à soit ajouter ou à éliminer après le développement du modèle dynamique.

### 4.3 Modèle dynamique

Nous proposons dans cette partie quelques scénarios clés dans l'application. Commander un produit (à transformer en classe abstraite) suit généralement la séquence suivante :

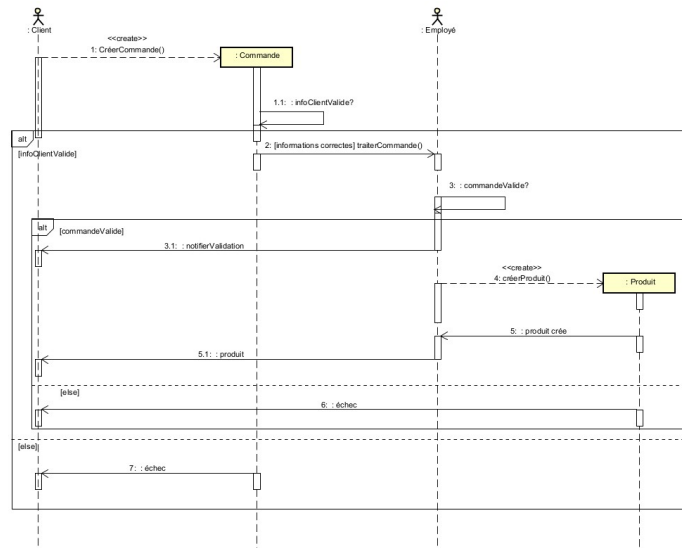


FIGURE 4.3 – Diagramme de séquence de cas commander un produit

Commander des cartes peut passer par une séquence générale comme suit :

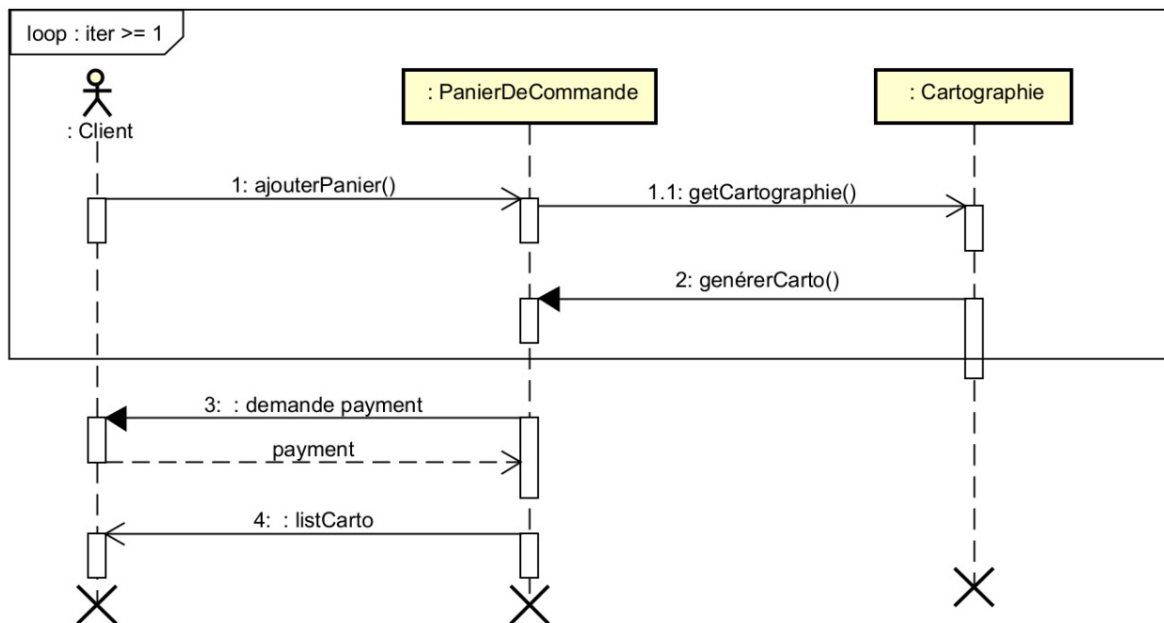


FIGURE 4.4 – Diagramme de séquence de cas commander des cartes

Un immobilier passe par des états lors de processus d'immatriculation. Le savoir de cette information peut servir lors d'une étape de développement ultérieure :



FIGURE 4.5 – Diagramme d'état-transition de classe Immobilier

# Chapitre 5

## Conception générique

La conception générique développe une solution conforme aux besoins techniques, indépendante des aspects fonctionnels, et aboutit à un prototype validé par les utilisateurs techniques. Elle nécessite un framework robuste et flexible pour un système diversifié et une base de données complexe. Une sélection de frameworks est proposée pour répondre aux exigences du projet.

### 1. Backend (Côté Serveur)

#### Spring Boot

##### Pourquoi choisir Spring Boot ?

- Framework Java puissant et flexible, idéal pour développer des applications backend complexes.
- Configuration simplifiée pour un démarrage rapide, tout en permettant une personnalisation avancée.
- Intégration native de modules comme Spring Security pour la gestion de la sécurité et Spring Data JPA pour interagir avec PostgreSQL.

##### Modules clés :

- **Spring Security** : Pour implémenter des mécanismes d'authentification et d'autorisation robustes.
- **Spring Data JPA** : Simplifie les interactions avec les bases de données relationnelles grâce à un ORM intégré.

### 2. Frontend (Côté Client)

#### AngularJS

##### Pourquoi choisir AngularJS ?

- Framework JavaScript qui permet de construire des interfaces utilisateur dynamiques et réactives.
- Prise en charge du *data binding* bidirectionnel, facilitant la synchronisation entre les données du modèle et la vue.
- Intégration facile avec des APIs RESTful sécurisées via JWT.

##### Avantages :

- Architecture modulaire, facilitant la réutilisation des composants.
- Large écosystème avec des bibliothèques et des extensions disponibles, comme Angular Material pour des interfaces modernes.

### 3. Base de Données

#### PostgreSQL

##### Pourquoi choisir PostgreSQL ?

- SGBD relationnel open source, reconnu pour sa stabilité et ses performances.
- Prend en charge des types de données avancés comme JSON, idéal pour stocker des données semi-structurées.



- Compatible avec Spring Boot via Spring Data JPA, simplifiant les opérations CRUD et les requêtes complexes.

**Cas d'utilisation :**

- Gestion des utilisateurs et des rôles.
- Stockage des données des sessions et des logs d'activité.

## 4. Sécurité et Authentification

### JWT (JSON Web Token)

**Pourquoi choisir JWT ?**

- Standard léger et sécurisé pour l'authentification sans état (*stateless*).
- Convient parfaitement aux architectures RESTful en sécurisant les communications entre le frontend et le backend.

### Spring Security

**Pourquoi choisir Spring Security ?**

- Module robuste pour gérer les mécanismes de sécurité dans les applications Spring Boot.
- Supporte l'intégration avec JWT pour protéger les endpoints RESTful.
- Permet la gestion fine des rôles et des permissions utilisateur.

### Architecture Recommandée :

Composant	Framework/Technologie	Rôle
Backend	Spring Boot	API principale, logique métier, gestion des données.
Frontend	AngularJS	Interface utilisateur réactive et interactive.
Base de Données	PostgreSQL	Stockage relationnel des données utilisateur.
Sécurité	JWT + Spring Security	Authentification, autorisation, et protection des APIs.

### Résumé des Avantages :

- Spring Boot offre un environnement backend modulaire et puissant, adapté aux systèmes complexes.
- AngularJS fournit une interface utilisateur moderne, optimisée pour les interactions dynamiques.
- PostgreSQL garantit une gestion robuste et fiable des données relationnelles.
- JWT et Spring Security renforcent la sécurité et permettent une gestion avancée des accès utilisateur.

## Chapitre 6

# Conception préliminaire

L'application ANCFCC fournit une interface répondant aux besoins des utilisateurs en simplifiant l'accès à des services concernant un immobilier, un document juridique ou un produit. Voici l'interface d'accueil : Pour chaque une des services, on donne l'accès à un choix entre déposer une commande de l'ob-



FIGURE 6.1 – Interface accueil

tention du service ou de suivre une commande : Cette même logique s'applique sur les autres produits à savoir le plan cadastral et calcul de contenance.

Le système offre le service de vérification de l'authenticité d'un document juridique liée à un immobilier foncier :

Les actualités concernant un titre se poste dans la partie **Publicité foncière** :

Au plus de ce qui précède, les visiteurs professionnel peuvent accéder à des informations propres à eux via une interface de l'authentification :





FIGURE 6.4 – Publicité foncière

**BIENVENUE SUR L'ESPACE CADASTRE ET CARTOGRAPHIE**

Connexion à votre compte :

Login :

Mot de passe :

Code de vérification :

2484

IDENTIFICATION

FIGURE 6.5 – Authentification pour les professionnels

## Chapitre 7

# Conception détaillée

Dans cette dernière étape de conception, nous allons essayer de tracer le diagramme des classes détaillé pour en partir construire par la suite un MLD pour le stockage des données.

Voici le diagramme de classes détaillé :

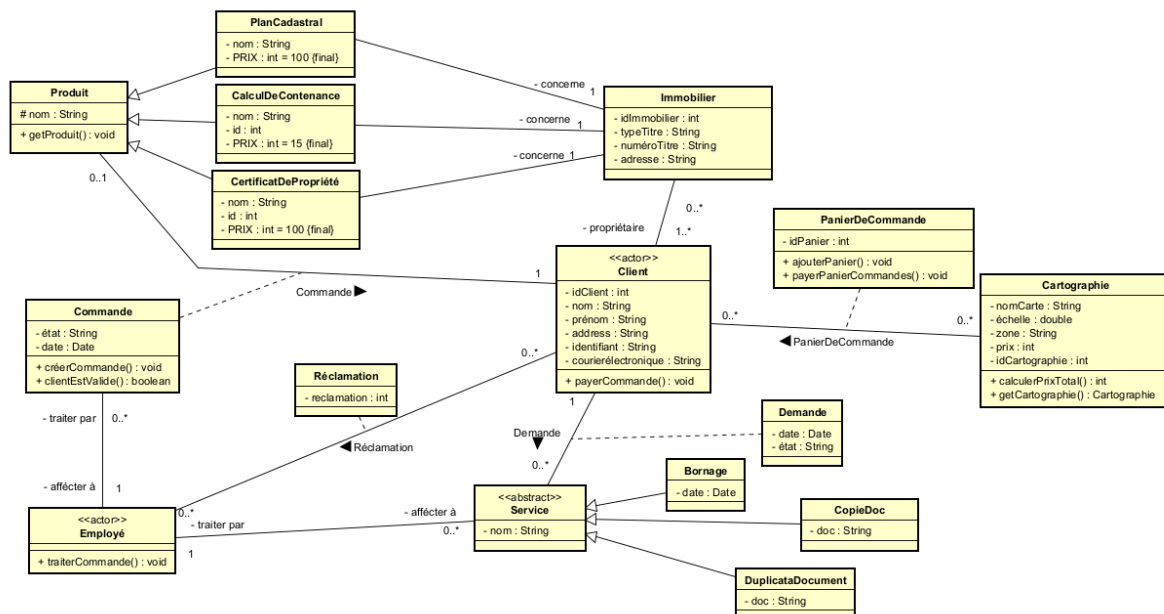


FIGURE 7.1 – Diagramme de classes détaillé

En faisant des transformations sur le diagramme de classes, on obtient le MLD :

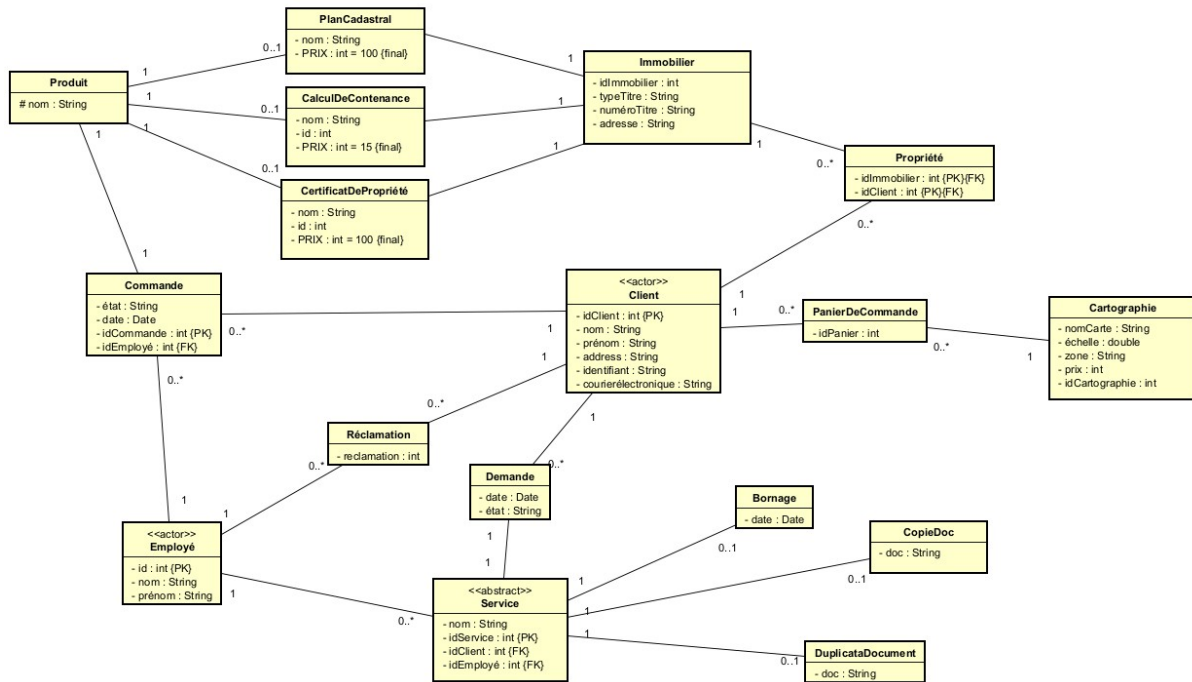


FIGURE 7.2 – Modèle logique de données

# Conclusion

Le projet Mohafadati s'inscrit dans une dynamique de digitalisation et d'amélioration des services fonciers, contribuant ainsi à une meilleure accessibilité et efficacité des démarches administratives. En s'appuyant sur des outils modernes de modélisation et de développement, le rapport met en lumière une méthodologie rigoureuse depuis l'identification des besoins jusqu'à la conception détaillée. Ces étapes, centrées sur l'utilisateur final, démontrent l'importance d'un cadre structuré pour garantir une solution intuitive et performante.

La collaboration entre les différents acteurs, notamment l'ANCFCC, et l'équipe de développement a permis d'identifier les défis majeurs et de concevoir des solutions adaptées. Le projet, une fois déployé, aura un impact significatif sur la simplification des interactions entre l'administration et les citoyens.

Enfin, les perspectives d'évolution incluent l'intégration de technologies émergentes comme l'intelligence artificielle pour améliorer la prédiction des prix ou la gestion des litiges, garantissant ainsi une amélioration continue du système.

# Bibliographie

- [1] Franck Vallée PASCAL ROQUES. *UML 2 à l'action : De l'analyse de besoins à la conception*. 4nd. 61, bd Saint-Germain 75240 Paris Cedex 0 : EYROLLES, février 2007.