# Introduction

Depuis quelques années, le marché du travail au Cameroun en particulier et dans le monde en général tend à se tourner vers des solutions numériques. Fort de ce constat, le département d’informatique de l’université de Yaoundé I a intégré dans l’exercice de son cours de modélisation un travail pratique. Durant ce cours nous avons appris la modélisation UML pour analyser et concevoir des systèmes informatique, dans l’application de nos connaissances les problèmes de gestion et d’acquisition des logements dans une mini-cité nous a été pose. Ainsi nous allons mettre en pratique nos connaissances pour y répondre en commençant par analyser le problème ensuite faire une modélisation pour enfin produire une application web permettant de gérer un mini cité de façon automatique.

# Analyse de la solution

La phase d’analyse est considérée comme la plus importante dans le processus de développement d’une application informatique. Dans cette partie il sera question pour nous d’identifier l’ensemble des fonctionnalités du système et de les modéliser grâce à la méthode UML

## Méthode d’analyse

### UML

UML est un langage graphique qui permet de représenter et de faire communiquer les divers aspects d’un système. UML dans sa version 2.0 comporte ainsi treize diagrammes permettant de représenter le système sur plusieurs vues distinctes. Ils se répartissent en trois grands groupes :

* **Diagrammes fonctionnels**

Ils comprennent le diagramme des cas d’utilisation et le diagramme d’activités.

* **Diagrammes statiques**

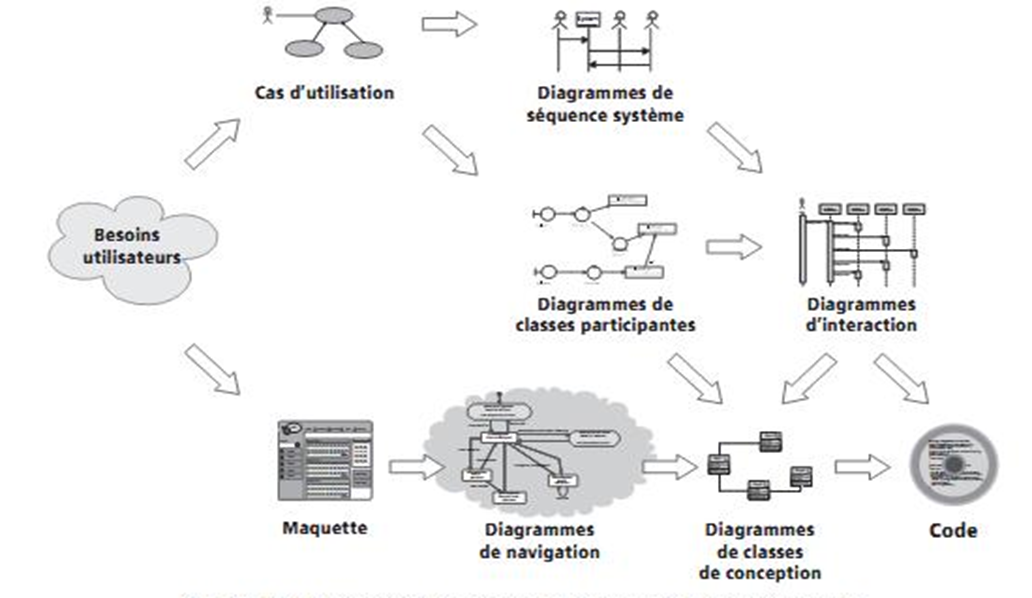
Ils comprennent le diagramme de classe, le diagramme d’objets, le diagramme de composants, le diagramme de déploiement.

* **Diagrammes dynamiques**

Ils comprennent le diagramme d’état de transition, le diagramme de séquence, le diagramme de collaboration.

### Formalisme

UML, telle qu’elle présente de nombreux diagrammes, présente tout aussi un formalisme pour chaque diagramme que nous allons utiliser tout en montrant les interactions entre eux dans une application web.



## Analyse des besoins et spécification des fonctionnalités

### Spécification

L’ambition de ca projet est de contribuer a l’émergence des TCI dans notre quotidien, il vice a automatisé le principe de gestion et d’acquisition d’un espace dans une mini cité de la place

En étant citoyen, il vous est déjà surement arrivé de cherche à acquérir un espace dans un mini cité. Parfois trouver un lieu de résidence peut être aussi facile que de résoudre une petite équation, mais cela peut ce relever très fastidieux dans certains cas. Les problèmes que rencontré au quotidien sont les suivant :

* La plus part des espaces dans la cité sont déjà sont déjà réserve
* Le respect de délais de paiement dans la cité n’est pas respecté
* Le responsable ne connait pas en tête toute les date line de paiement des habitants de la cite

Alors il serait préférable pour un responsable d’une cité de facilement connaitre les date line pour un contrat avec un locataire. Pour un visiteur de facilement trouver un maison dans une mini-cité.

### Identification des besoins

#### Besoins fonctionnelles

Ici, sont listés l’ensemble des besoins fonctionnels du système.

* Recherche une maison
* Créer une mini cité
* Enregistrer un locataire
* Enregistrer une maison
* Attribuer une maison a un locataire
* Consulter une maison
* Etablir un contrat
* Modifier l’état d’une maison
* Mise à jour du versement

#### Besoins non fonctionnels

* Notifier un locataire
* Notifier un bailleur

### Représentation des besoins fonctionnels : DUC

#### Définition

Un diagramme de cas d’utilisation capture le comportement d’un système, d’un sous-système tel qu’un utilisateur extérieur le voit. Il scinde les fonctionnalités d’un système en unités cohérentes. Un cas d’utilisation est une unité cohérente représentant une fonctionnalité visible de l’extérieure ; il réalise un service de bout en bout, avec un déclanchement, un déroulement et une fin.

#### Identification des acteurs

Les acteurs sont des entités externes qui interagissent avec le système, comme une personne humaine, un autre système ou un robot. Les acteurs sont représentés par un pictogramme sous-titré par le nom de l'acteur.

Les principaux acteurs qui auront à utiliser le système sont les suivants :

|  |  |
| --- | --- |
| Acteur | Rôles |
| Bailleur | Ajouter des maisons, ajouter des locataires, créer une mini cite. |
| Visiteur | Faire une recherche de maison libre |

#### Description détaillée et représentation des cas d’utilisation



|  |  |
| --- | --- |
| Nom du cas d’utilisation | Rechercher une maison |
| Acteur principal | Visiteur |
| Acteur secondaire |  |
| Objectif | Trouver des maisons libres |
| Précondition |  |
| Scenario alternatif | * Aucune maison n’est disponible pour sa recherche |
| Scenario nominale | * Des maison son libre pour sa recherche |
| Post condition | * Présentation de l’interface d’accueil |

## Diagramme de classe

### Définition

Le diagramme de classe de conception représente bien la structure statique du code, par le biais des attributs et des relations entre classes C'est un diagramme principal qui est la vue de plus haut niveau avec l'ensemble des classes de l'application, Il montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations, etc. on peut avoir des vues logiques plus spécialisées si nécessaire :

 Vue des classes participant à un scénario

 Vue des classes "privées" dans un package

 Vue d'une classe avec ses attributs et ses opérations

 Vue d'une hiérarchie d'héritage

### Représentation du diagramme de classe



# Conception