

## **Module de Synthèse**

### **« Cow-Lock »**

Identification des différentes étapes de conception :

- Présentation du projet
- Cahier des charges
- Étude préliminaire
- Capture des besoins fonctionnels
- Capture des besoins techniques (non soumise)
- Analyse
- Conception générique (non soumise, dépend de la capture des besoins techniques)
- Conception préliminaire
- Conception détaillée
- Codage, tests, recette



## **I. Présentation du projet**

« Cow-Lock » est une application présentée sous forme de site web qui va permettre de définir toutes les tâches directement liées à une colocation afin d'en assurer la gestion sous tous ses aspects, du jour où l'état des lieux est effectué, en passant par un changement de colocataire(s) en cours de mois, jusqu'au rendu des clefs en fin de colocation.

Après constatation et expérimentation personnelle et devant la difficulté croissante d'autres personnes sur des thèmes impliquant directement la gestion des loyers, la gestion des courses alimentaires, l'entretien du logement, et le partage de ces dépenses dans un foyer, l'idée d'une application qui recense tous les éléments relatifs à une colocation et la mise en œuvre des solutions liées aux contraintes quotidiennes que pourront éprouver les colocataires fera je pense un bon sujet pour un module de synthèse et pourra bénéficier à d'autres personnes une fois le projet finalisé.

## **II. Cahier des charges**

« Cow-Lock » est une application web offrant gratuitement aux internautes concernés par une colocation de créer une instance logicielle détaillant les spécificités de leur colocation, de l'enregistrer et de se greffer à cette instance en tant que colocataires, quel que soit leur nombre ou leur implication active dans cette colocation.

Une fois ces paramètres saisis, et selon leur droits d'accès qu'ils auront récupérés sur le système en fonction de ce qu'aura défini le créateur de cette instance de colocation, il leur sera permis de gérer et de compléter certains aspects du logiciel au fil des mois et d'accéder aux informations finales offertes par le système en fonction de ces renseignements afin d'y voir plus clair et de les aider dans leur gestion.

Ce logiciel souhaite principalement contribuer à :

- Créer une instance. Enregistrer les différents utilisateurs du système et définir les modalités de la colocation.
- Répartir le montant du loyer entre les différents colocataires.
- Assurer le suivi des dépenses de façon mensuelle afin de les gérer et de les répartir équitablement. Partager également les dépenses pour les travaux à effectuer dans le foyer.
- Assurer un ajout, retrait ou modification d'un colocataire en cours d'exercice.

### Positionnement :

Le site Web « Cow-Lock » souhaite se placer sur le marché de gestion des colocations sur Internet. Bien qu'un sujet tel que celui-ci n'ait pas de concurrence existante directe internationalement et qu'un seul projet dans le passé fût développé (<http://xuazak.free.fr/coloc/>), il a été rapidement abandonné faute de temps à y consacrer par son créateur. Il existe beaucoup d'entreprises proposant sur Internet des annonces de recherche de colocation et mettant en relation des colocataires entre eux, cela gratuitement ou moyennant paiement. Ces entreprises pourraient très bien être intéressées par l'activité de ce projet en contribuant à leur évolution

### III. Étude préliminaire

#### 1. Grands Choix techniques :

Afin de définir tous les besoins techniques qui pourront participer à l'élaboration complète du projet, « Cow-Lock » souhaite adopter une approche itérative et incrémentale fondée sur le processus en « Y » qu'apporte UML en particulier le processus 2TUP.

Pourquoi ?

- Car il est piloté par les risques, évitant par exemple que l'architecture technique soit incapable de répondre aux contraintes opérationnelles ou bien permettant d'éviter l'inadéquation du développement avec le besoin des utilisateurs.
- Car il est construit autour de la création et de la maintenance du projet et permet ainsi de participer à son évolution. Cette démarche apporte la souplesse nécessaire, autant pour le modèle que pour le logiciel lui-même.
- Car il est orienté composants et utilisateurs. La spécification et la conception du programme est construite à partir des modes d'utilisation attendus par les acteurs du système.

Les technologies qui seront donc utilisées sont :

- a. La modélisation objet avec UML.
- b. Une architecture 2 tiers avec SQL Server choisi comme SGBDR.
- c. Le déploiement en client web utilisant ASP.NET et ADO.NET.
- d. L'utilisation du langage C# pour la programmation du projet.

#### 2. Recueil des besoins fonctionnels :

##### *Enregistrement d'une colocation et/ou d'un colocataire*

Lors d'un premier enregistrement, une nouvelle instance de colocation sera créée sur la base de donnée par un des colocataires qui devient « Gestionnaire » de la colocation. Il sera demandé pour cela toutes les caractéristiques relatives au logement, le loyer, le propriétaire, ses coordonnées, la date de début d'entrée dans l'habitation et les informations personnelles.

Parallèlement, un login et un mot de passe utilisateur ainsi qu'un nom de colocation et un mot de passe de colocation associés seront saisis.

Si un colocataire souhaite s'inscrire sur une colocation déjà existante, les renseignements personnels, la date d'entrée effective, les données d'identification ainsi que le nom et le mot de passe de l'instance en question seront demandés.

Une fois confirmé, ces informations sont mises à disposition de tous les utilisateurs de l'instance choisie ainsi qu'au système pour effectuer plus tard toutes les opérations de gestion. Seul le gestionnaire peut modifier les informations de la colocation.

En fonction de la date où un colocataire s'inscrit, le montant du loyer qui doit être intégré pour lui pour le mois en cours ainsi que la répartition du nouveau montant entre chaque colocataire est calculée par le système (*non implémenté, fonctionnalité prévue*).

### *Supprimer ou modifier un colocataire*

Un colocataire peut à tout moment modifier ses informations personnelles. Le gestionnaire peut également supprimer chaque colocataire de la colocation si la suppression ne concerne pas un gestionnaire.

S'il quitte le logement, et en fonction de la date où cette action s'effectue, le montant du loyer qu'il lui reste à payer ainsi que la nouvelle répartition entre chaque colocataire pour le mois en cours est calculée par le système *(non implémenté)*.

### *Enregistrement des dépenses*

Un colocataire peut à tout moment ajouter une dépense pour le mois en cours, que ce soit alimentaire, matériel ou concernant des travaux à réaliser dans le logement; tant que cette dépense concerne tous les colocataires présents.

En effet, les dépenses personnelles ne concernent pas la colocation directement et ne sont pas traitées par le logiciel.

La dépense est enregistrée dans le système avec ses informations pour un suivi ultérieur. La dépense peut être modifiée ou supprimée si elle est traitée dans le mois la concernant.

### *Supprimer ou modifier une dépense*

Le gestionnaire peut modifier ou supprimer une dépense. Le montant final des répartitions est modifié par le système.

### *Suivi des dépenses*

Chaque colocataire peut suivre en temps réel les dépenses effectuées pour le mois en cours. Que ce soit le loyer, les dépenses alimentaires, les dépenses matérielles ou bien celles pour les travaux.

A partir de ces éléments, le système calcule et affiche la répartition de ces dépenses pour chaque colocataire pour le mois en cours à la date de visualisation.

En cas de dépense alimentaire, si un colocataire est absent pendant une semaine ou bien une autre durée déterminée et qu'il ne jouit pas de cette dépense en particulier, l'initiateur de la colocation peut exclure le colocataire en question de cette dépense ce qui sera répercuté bien entendu sur le montant de la répartition entre tous.

A tout moment, chaque colocataire peut demander au système d'effectuer des statistiques pour la semaine, mois, année en cours ou passée ainsi que pour l'exercice total depuis le début effectif de la colocation, séparant le loyer et les dépenses ou bien affichant le montant total *(juste le suivi pour le mois choisi est disponible)*.

### 3. Recueil des besoins opérationnels :

#### *Volume de données*

Toutes les données du système doivent être traitées en temps réel sur demande : la gestion des colocations, la gestion des colocataires, la gestion des dépenses, le suivi des dépenses, la clôture en fin de mois, la consultation d'historique et les statistiques.

Une première évaluation du volume de données a donné le modèle suivant :

Type de données	Volume unitaire (ko)	Quantité (par mois)	Volume total (Mo)
Ajout de colocation	0,5	10 000	5 000
Modification de colocation	0.1	2 000	200
Suppression de colocation	0.1	50	5
Ajout de colocataire	0.2	30 000	6 000
Modification de colocataire	0.1	6 000	600
Suppression de colocataire	0.1	150	15
Ajout de dépense	0.2	1 080 000	216 000
Modification de dépense	0.1	216 000	21 600
Suppression de dépense	0.1	43200	4320
Suivi des dépenses	0.1	1 080 000	108 000
Clôture fin de mois	0.1	216 000	21 600
Données d'exploitation			383 340

Type de données	Volume unitaire (ko)	Quantité	Volume total (Mo)
Colocation	5	1 000	4,9
Colocataire	2	3 000	5,9
Dépense	2	108 000	211
Feuille Historique (mois)	3	15 000	44
Données statiques			265,8

#### *Sécurité*

Lors de sa connexion, chaque colocataire doit être reconnu du système par un nom d'utilisateur et un mot de passe.

Chaque colocation est accessible par son nom et son mot de passe.

Un colocataire peut avoir deux rôles, gestionnaire ou simple colocataire.

Un administrateur système devra être identifié par le système et pourra gérer les profils.

#### 4. Acteurs du système :

**Utilisateur :** Un utilisateur s'est enregistré et identifié sur le site mais il n'est pas encore associé à une colocation. Un utilisateur possède un attribut ayant pour valeur « 0 ».

**Colocataire :** Un colocataire peut ajouter des dépenses. Il peut se greffer à une instance de colocation s'il possède les identifiants de la colocation. Il peut également consulter les informations relatives à la colocation, aux colocataires présents, effectuer un suivi des dépenses pour le mois en cours et modifier ses informations.  
Un colocataire possède un attribut ayant pour valeur « 1 ».

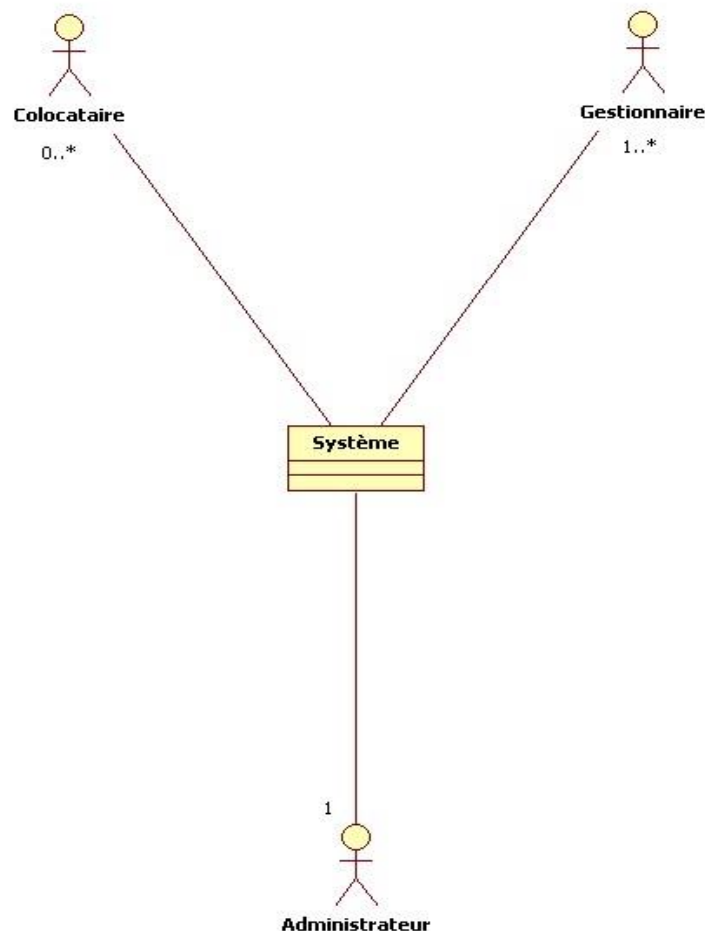
**Gestionnaire :** Le gestionnaire peut, en plus des actions possibles pour un simple colocataire, supprimer des colocataires ou modifier et supprimer des dépenses puis modifier ou supprimer son instance de colocation. Il peut définir un autre gestionnaire parmi les colocataires. Il peut être celui qui a créé la colocation ou bien avoir été défini par un autre gestionnaire.

Un gestionnaire est un colocataire avec un attribut ayant pour valeur « 2 ».

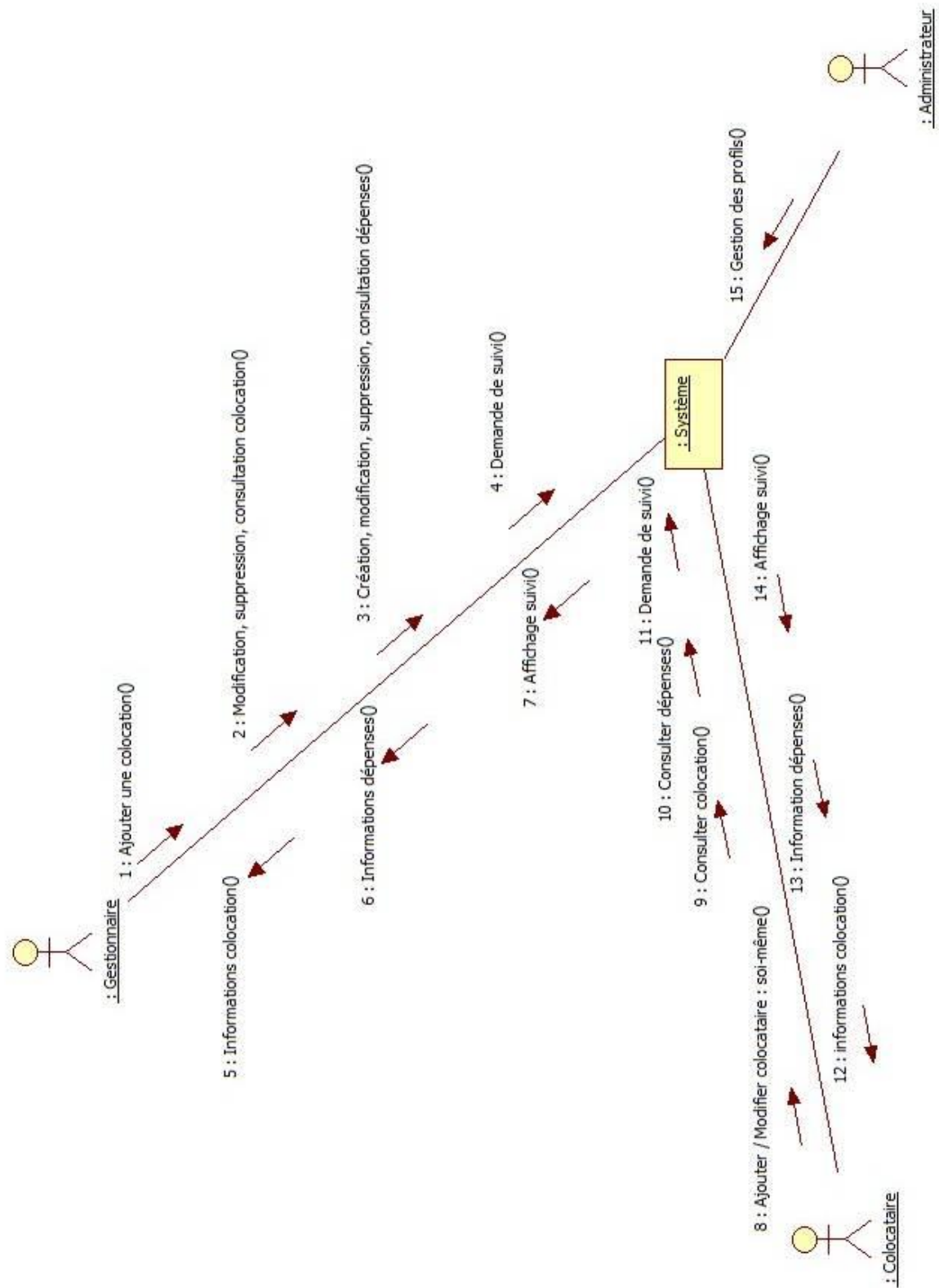
Le gestionnaire créateur de la colocation ne peut être supprimé.

**Administrateur système :** L'administrateur modifie les rôles et veille à la sécurité et au support client par l'intermédiaire des fonctions mise à disposition par le système.

#### 5. Diagramme de contexte statique



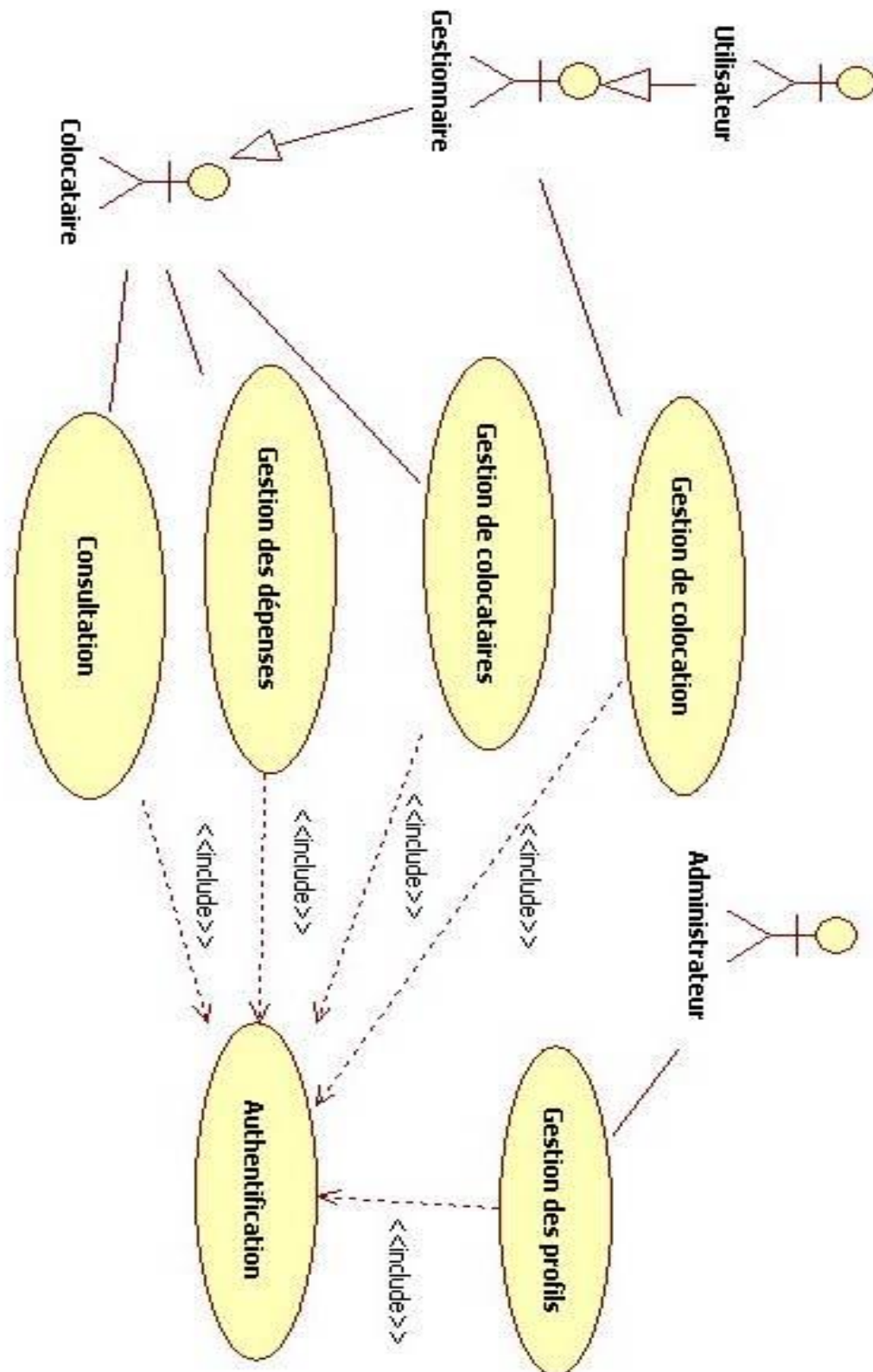
## 6. Diagramme de contexte dynamique (collaboration)





## IV. Capture des besoins fonctionnels

### 1. Diagramme de cas d'utilisations :



## 2. Cas d'utilisations :

### Cas d'utilisation 1 : Gestion de colocataires

But : Ajouter ou modifier un colocataire.

Acteurs : Gestionnaire, Utilisateur

Date de création : 09/09/2008

Date de mise à jour : 09/09/2008

Version : 1.0

Responsable : Thierry PROST

#### Description des enchaînements :

Pré conditions : Si l'utilisateur n'est pas authentifié, enchaînement Ajout.

Si l'utilisateur est authentifié, enchaînement Modifier.

#### Enchaînement 1 : Ajouter un colocataire

- Le système demande les informations personnelles (nom, prénom, adresse, code postal, ville, sexe, date de naissance, téléphone).
- L'utilisateur enregistré fournit les informations personnelles.
- Le système demande les informations d'authentification (login, mot de passe, email, question de sécurité, réponse de sécurité).
- L'utilisateur non enregistré fournit les informations d'authentification.
- L'utilisateur non valide la demande de création de compte personnel.
  - o Si login déjà utilisé, exécuter Exception 1.
- Le système enregistre le compte.
- Le système authentifie l'utilisateur sur l'application.

#### Enchaînement 2 : Modifier un colocataire

- Le système fournit les informations de l'utilisateur enregistré qui effectue la demande.
- L'utilisateur enregistré fournit les modifications.
- L'utilisateur enregistré valide les modifications.
- Le système enregistre les modifications.

#### Exceptions :

- Exception 1 : Il n'y a que le gestionnaire qui puisse effectuer des suppressions de colocataires sur la colocation.
- Exception 2 : Il est impossible de supprimer un gestionnaire. Vous devez enlever le statut de gestionnaire au colocataire que vous désirez supprimer auparavant.

#### Post-conditions :

- Aucunes.

#### Besoin d'IHM (Interface Homme Machine) :

- L'acteur peut à tout moment annuler et revenir à la page précédente.

### Exigences non fonctionnelles :

**Confidentialité :** Le gestionnaire ou le colocataire est identifié par un nom d'utilisateur et un mot de passe.

**Fréquence du cas d'utilisation :** Plusieurs fois par jour.

**Temps de réponse :** - de 5sc.

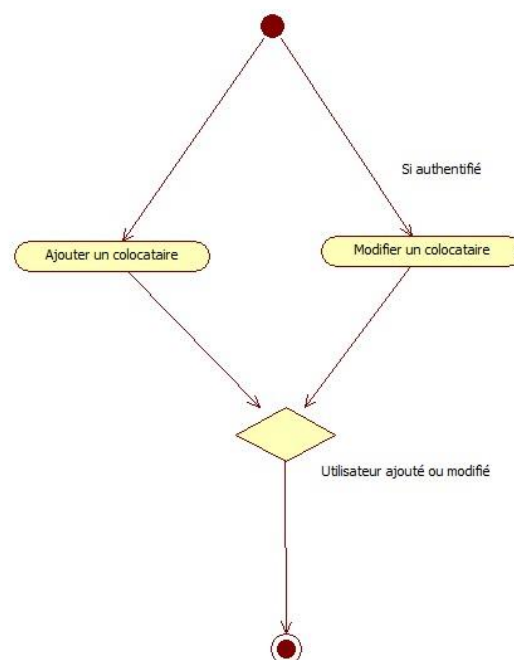
**Disponibilité :** 7 jours par semaine 24/24h.

**Concurrence :** 10 personnes.

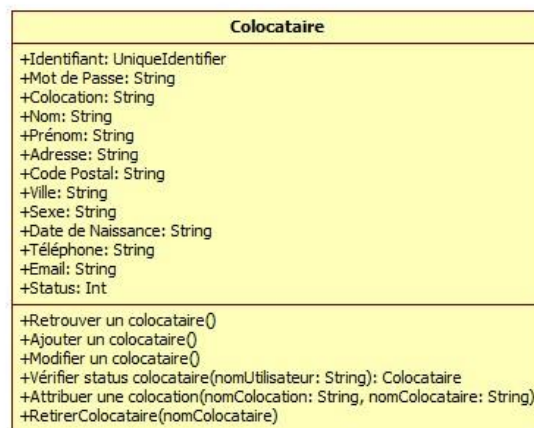
**Intégrité :** 1.

**Volumétrie :** 200 par jour.

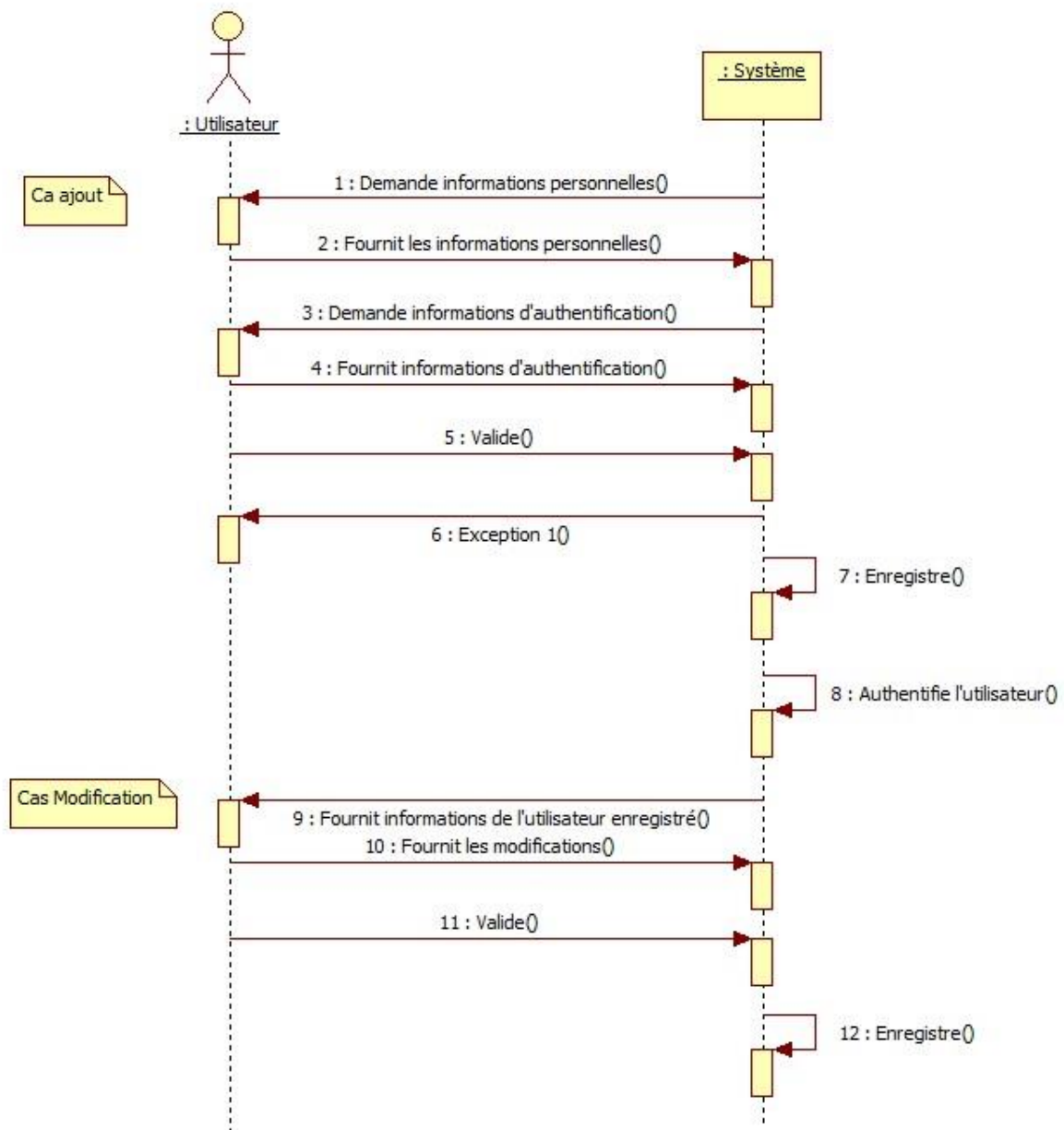
### Diagramme d'activité :



### Diagramme de classes participantes :



## Diagramme de séquence :



## Cas d'utilisation 2 : Gestion de colocation

But : inscrire une colocation, se greffer sur une colocation, consulter/modifier une colocation.

Acteurs : Utilisateur Enregistré, Gestionnaire

Date de création : 09/09/2008

Date de mise à jour : 09/09/2008

Version : 1.0

Responsable : Thierry PROST

*Description des enchaînements :*

Pré-conditions : L'utilisateur doit avoir créé un compte auparavant.

Si l'utilisateur n'a pas de colocation associée, il peut en créer une ou se greffer sur une colocation.

Si l'utilisateur possède une colocation, il pourra la consulter / modifier.

### Enchaînement 1 : Vérification statut

- Le système vérifie que l'utilisateur soit authentifié.
- Le système vérifie que l'utilisateur authentifié possède une colocation.
  - S'il est authentifié sans colocation, le système fournit le choix entre « Inscrire une colocation » ou bien « Se greffer sur une colocation existante ».
  - Si l'utilisateur authentifié possède une colocation, le système fournit la possibilité de consulter / modifier une colocation.

### Enchaînement 2 : Inscrire une colocation

- Le système demande les renseignements du logement (adresse, code postal, ville, surface en m², nombre de chambres, date de début, type de bail, montant total du loyer).
- L'utilisateur enregistré fournit les renseignements du logement.
- Le système demande les informations sur le propriétaire (nom, prénom, adresse, code postal, ville, téléphone, portable, email).
- L'utilisateur enregistré fournit les informations du propriétaire.
- Le système demande les informations d'identification de la colocation (identifiant, mot de passe).
- L'utilisateur fournit les informations d'identification de la colocation.
- L'utilisateur enregistré valide la demande de création de colocation.
  - Si l'identifiant choisi pour la colocation est déjà utilisé, exécuter Exception 1.
- Le système associe l'utilisateur enregistré à la colocation. L'utilisateur devient gestionnaire.
- Le système enregistre la colocation (informations logement, informations propriétaire, informations d'identification).

### Enchaînement 3 : Se greffer sur une colocation

- Le système demande les renseignements de la colocation (nom de la colocation, mot de passe de la colocation).
- L'utilisateur enregistré fournit les renseignements de la colocation.
  - o Si le nom de colocation n'existe pas, exécuter Exception 2.
  - o Si le mot de passe de la colocation est incorrect exécuter Exception 3.
- Le système associe l'utilisateur enregistré à la colocation.

### Enchaînement 4 : Consulter / Modifier une colocation

- Le système vérifie si la demande est effectuée par un gestionnaire ou un colocataire.
- Le système fournit les informations sur les colocataires (nom, prénom, statut).
- Le système fournit les informations sur la colocation (adresse, code postal, ville, surface en m², nombre de chambres, date de début, type de bail, montant total du loyer).
- Le système fournit les informations sur le propriétaire (nom, prénom, adresse, code postal, ville, téléphone, portable, email).

*Si c'est un gestionnaire :*

- Le gestionnaire fournit les modifications.
- Le gestionnaire valide les modifications.
- Le système enregistre les modifications.

### Exceptions :

- Exception 1 : L'identifiant choisi pour la colocation doit être unique.
- Exception 2 : Le nom de colocation saisi n'existe pas.
- Exception 3 : Le mot de passe de colocation ne correspond pas.

### Post-conditions :

- Le gestionnaire est associé à la colocation.

### Besoin d'IHM (Interface Homme Machine) :

- L'acteur peut à tout moment annuler et revenir à la page précédente.

### Exigences non fonctionnelles :

*Confidentialité* : Le gestionnaire est identifié par un nom d'utilisateur et un mot de passe.

*Fréquence du cas d'utilisation* : Plusieurs fois par jour.

*Temps de réponse* : - de 5sc.

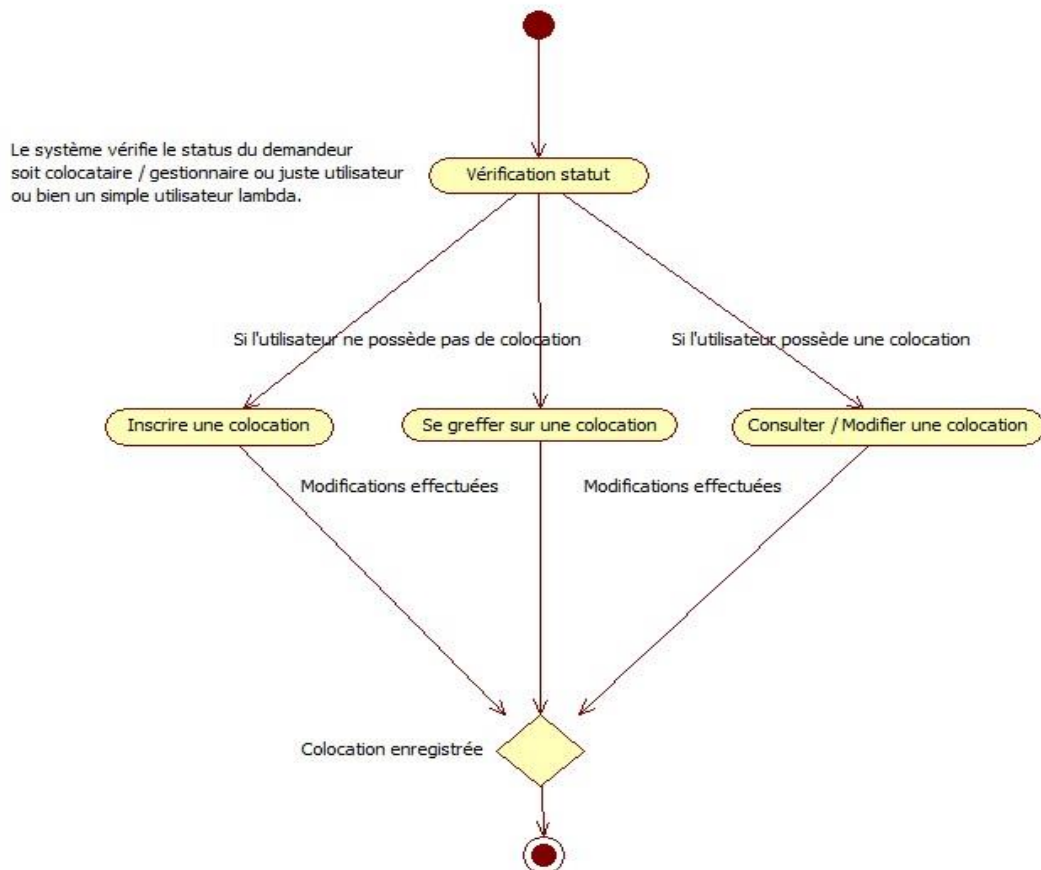
*Disponibilité* : 7 jours par semaine 24/24h.

*Concurrence* : 5 personnes.

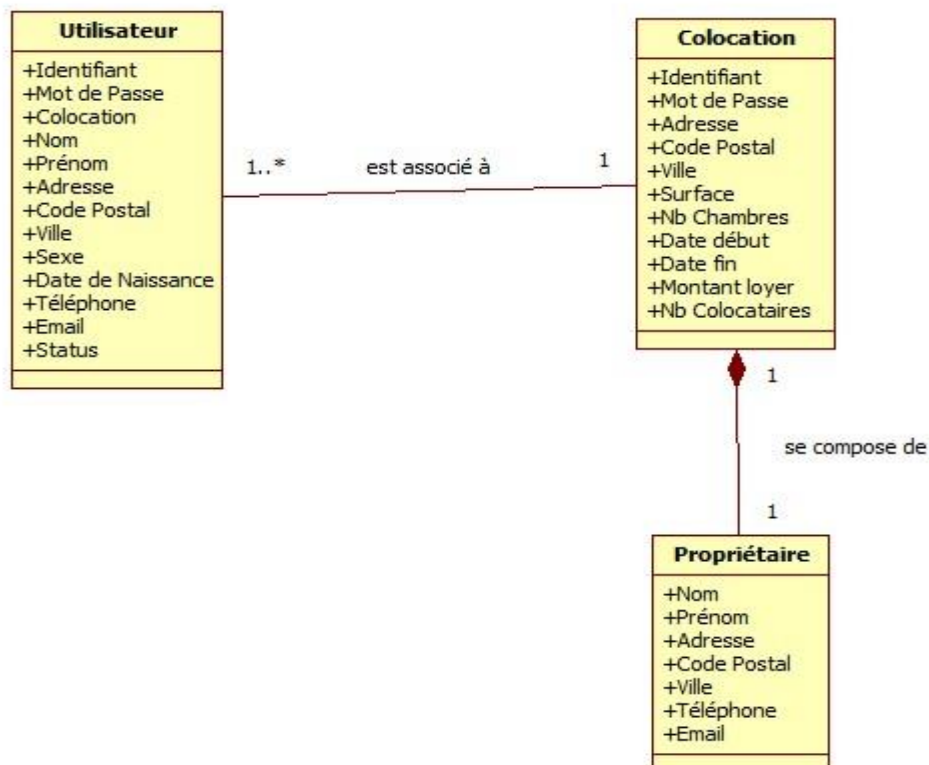
*Intégrité* : 1.

*Volumétrie* : 20 créations de colocation par jour.

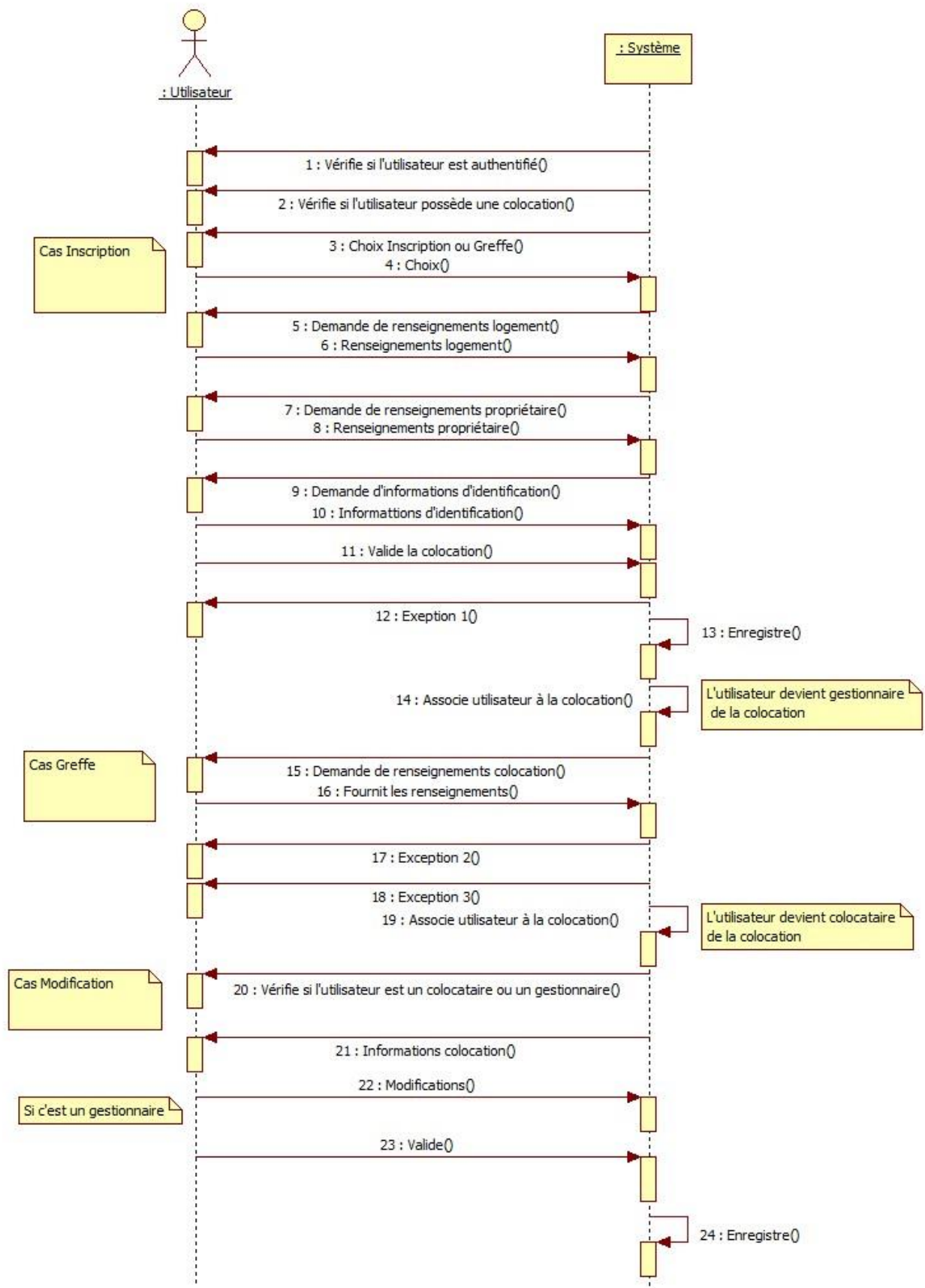
## Diagramme d'activité :



## Diagramme de classes participantes :



## Diagramme de séquence :





### Cas d'utilisation 3 : Gestion des dépenses

But : Consulter, ajouter, modifier ou supprimer une dépense.

Acteur : Gestionnaire, Colocataire

Date de création : 09/09/2008

Date de mise à jour : 09/09/2008

Version : 1.0

Responsable : Thierry PROST

*Description des enchaînements :*

Pré conditions : Le colocataire doit être identifié.

Le colocataire peut juste ajouter des dépenses et les consulter. Le gestionnaire peut en plus les modifier

Enchaînement 1 : Consulter une dépense

- Le système affiche les mois (mois).
- Le colocataire sélectionne le mois désiré.
- Le système fournit la liste des dépenses pour le mois choisi (n°, type dépense, libellé, date, montant total).
- Le colocataire sélectionne la dépense.
- Le système fournit les informations de la dépense (n°, type dépense, libellé, date, montant total, liste des colocataires concernés, si ces colocataires ont payé ou pas).

Enchaînement 2 : Ajouter une dépense

- Le système fournit la liste des types de dépense (type dépense).
- Le système fournit les colocataires présents dans la colocation (nom, prénom).
- Le colocataire fournit les informations de la dépense (type, libellé, date, montant, colocataires concernés, si ces colocataires ont payés ou pas).
- Le colocataire valide la demande d'ajout de dépense.
- Le système enregistre la dépense.

Enchaînement 2 : Modifier une dépense

- Le système vérifie le statut du colocataire (colocataire ou gestionnaire).
- Le colocataire fournit les modifications.
- Le colocataire valide la demande de modification.
- Le système enregistre la demande de modification.

Enchaînement 4 : Supprimer une dépense

- Le système vérifie si la demande est effectuée par un colocataire ou un gestionnaire.
- Le colocataire valide la demande de suppression.
- Le colocataire confirme la demande de suppression.
- Le système supprime la dépense.

Exceptions :

Post-conditions :

- Aucunes.

Besoin d'IHM (Interface Homme Machine) :

- L'acteur peut à tout moment annuler et revenir à la page précédente.

Exigences non fonctionnelles :

*Confidentialité* : Le gestionnaire ou le colocataire est identifié par un nom d'utilisateur et un mot de passe.

*Fréquence du cas d'utilisation* : Plusieurs fois par jour.

Temps de réponse : - de 5sc.

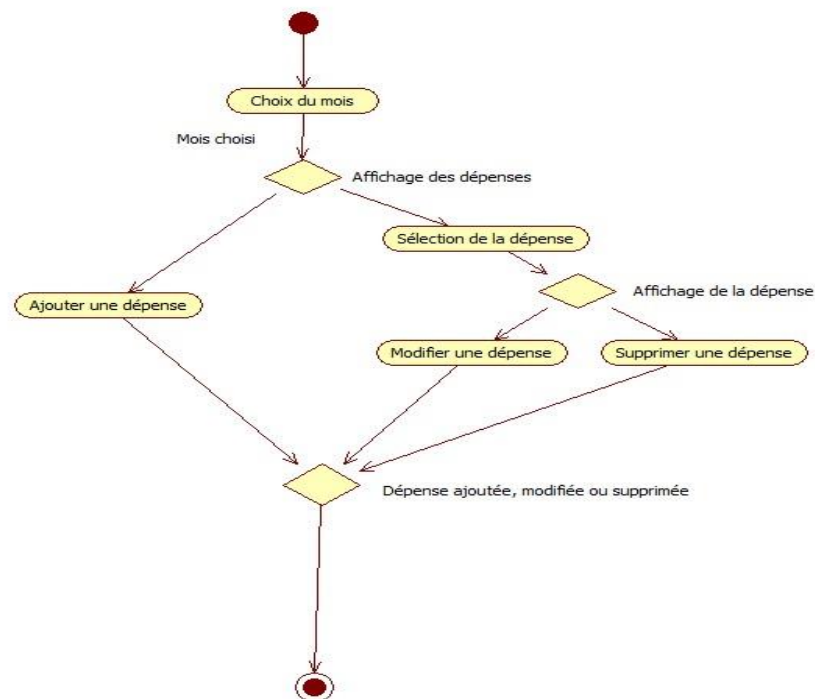
*Disponibilité* : 7 jours par semaine 24/24h.

*Concurrence* : 10 personnes.

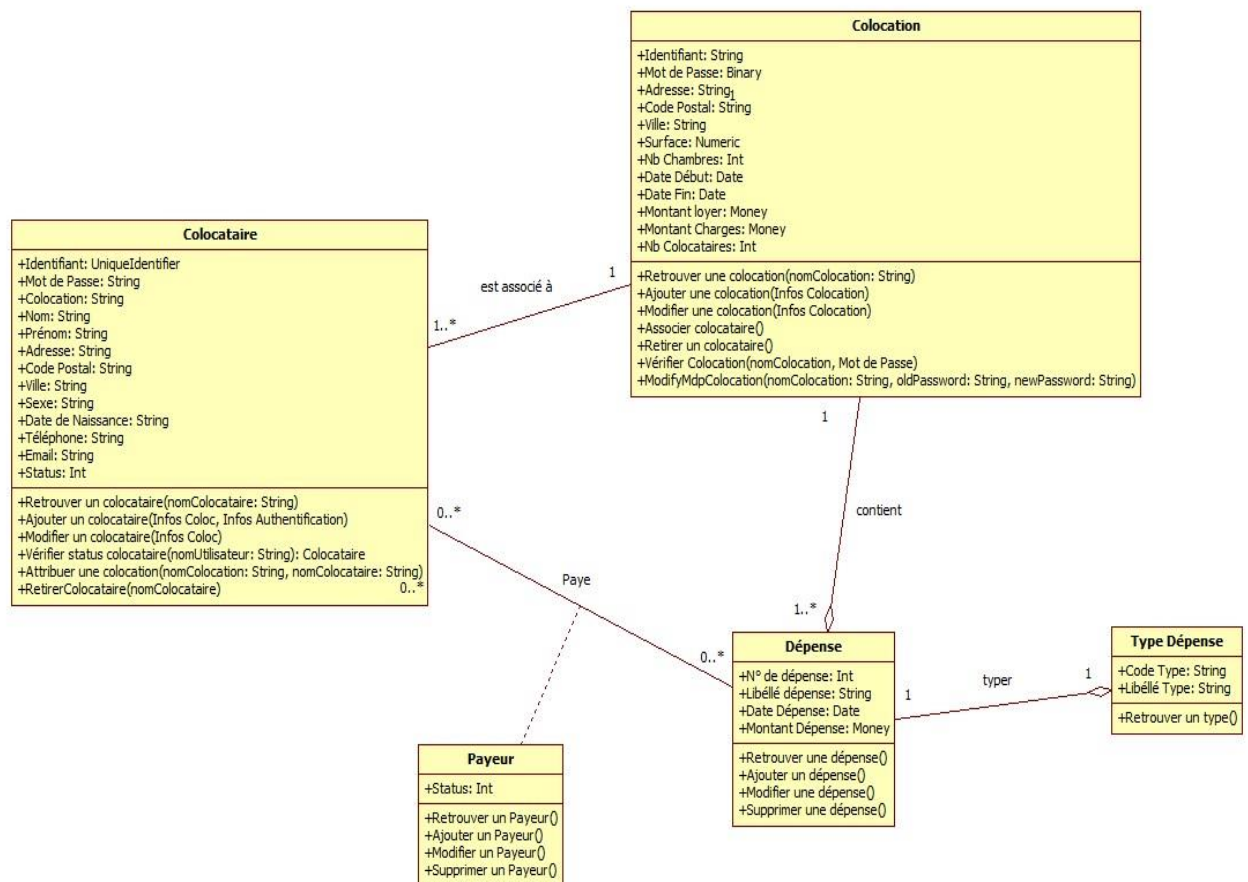
*Intégrité* : 1.

*Volumétrie* : 200 par jour.

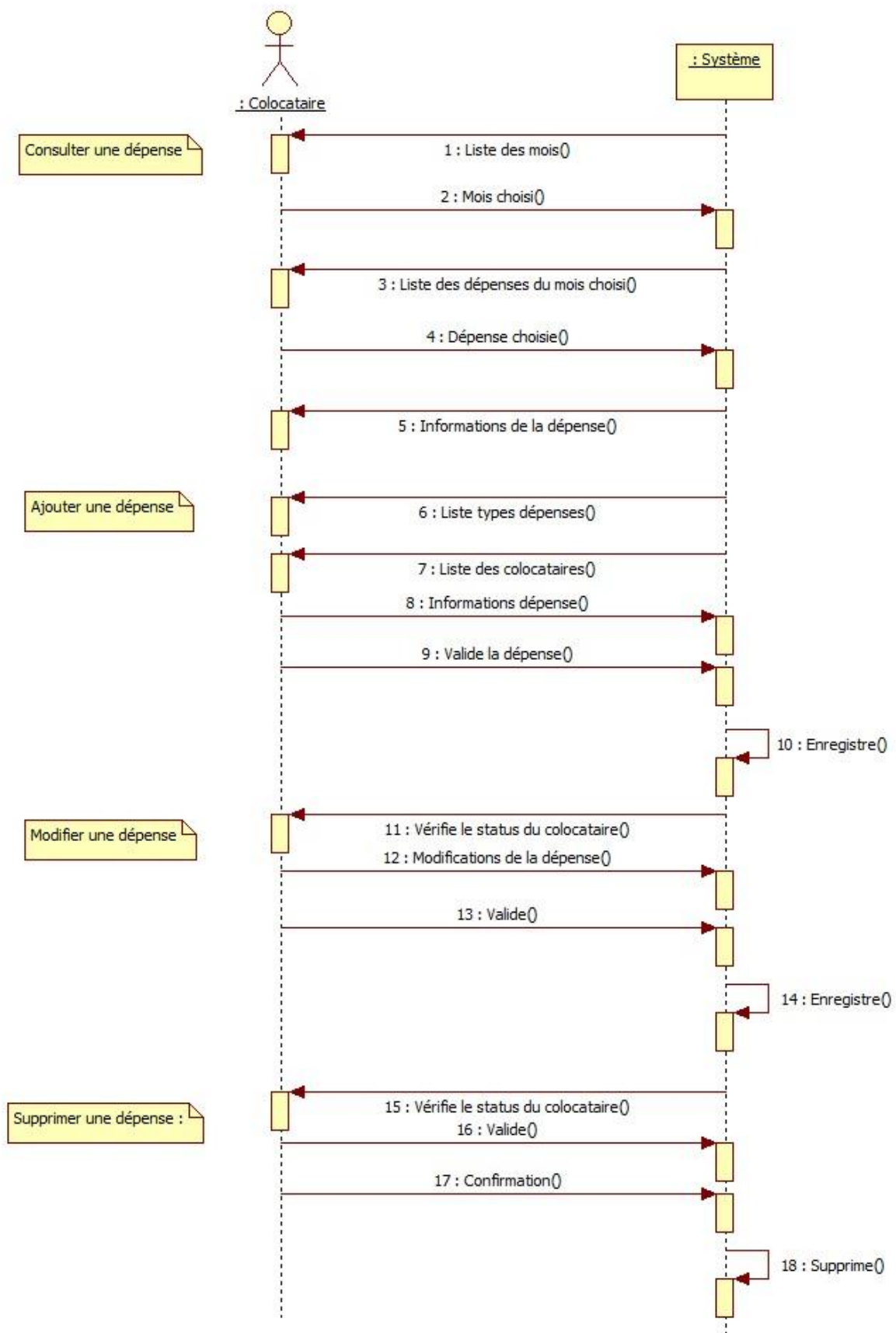
## Diagramme d'activité :



## Diagramme de classes participantes :



## Diagramme de séquence :



## Cas d'utilisation 4 : Consultation

But : Consulter les informations sur la colocation, les colocataires, les dépenses et les statistiques. Effectuer un suivi des dépenses pour le mois en cours.

Acteur : Colocataire

Date de création : 09/09/2008

Date de mise à jour : 09/09/2008

Version : 1.0

Responsable : Thierry PROST

*Description des enchaînements :*

Pré conditions : Le colocataire (principal).

Enchaînement 1 : Consulter les informations sur la colocation

- Le système fournit les informations sur la colocation (informations logement, informations propriétaire).

Enchaînement 2 : Consulter les informations sur un colocataire

- Le système fournit la liste des colocataires (nom, prénom).
- Le colocataire sélectionne le colocataire voulu.
- Le système fournit les informations sur le colocataire choisi (nom, prénom, date d'entrée, date de sortie, sexe, date de naissance, téléphone, portable, email).

Enchaînement 3 : Consulter les informations sur une dépense

- Le système fournit la liste des dépenses pour le mois en cours (n°, type dépense, libellé, montant total).
- Le colocataire sélectionne la dépense à consulter.
- Le système fournit les informations de la dépense (n°, type dépense, libellé, date, montant total, liste des colocataires exclus).

Enchaînement 4 : Consulter les statistiques

- Le système fournit la liste des mois clos (mois).
- Le colocataire sélectionne le mois désiré.
- Le système fournit les statistiques pour le mois choisi (mois, montant loyer, montant total des dépenses, montant total, total par colocataire).

Enchaînement 5 : Suivi des dépenses

- Le système fournit la liste des dépenses pour le mois en cours (mois, montant loyer, montant total des dépenses, montant total, total par colocataire).

Exceptions :

- Aucunes.

Post-conditions :

- Aucunes.

Besoin d'IHM (Interface Homme Machine) :

- L'acteur peut à tout moment revenir au menu.

Exigences non fonctionnelles :

*Confidentialité* : Le colocataire est identifié par un nom d'utilisateur et un mot de passe.

*Fréquence du cas d'utilisation* : Plusieurs fois par jour.

Temps de réponse : - de 5sc.

*Disponibilité* : 7 jours par semaine 24/24h.

*Concurrence* : 50 personnes.

*Intégrité* : 1.

*Volumétrie* : 500 par jour.

## Cas d'utilisation 6 : Authentification

But : S'authentifier sur le système. Enregistrer un compte utilisateur.

Acteur : Utilisateur

Date de création : 09/09/2008

Date de mise à jour : 09/09/2008

Version : 1.0

Responsable : Thierry PROST

*Description des enchaînements :*

Pré conditions : ...

*Description des enchaînements :*

### Enchaînement 1 : Authentification

#### *Cas utilisateur non enregistré (Utilisation « Membership » Classes ASP.Net) :*

- L'utilisateur fournit ses informations personnelles (id, mot de passe, nom, prénom, adresse, code postal, ville, sexe, date de naissance, téléphone, portable, email).
- L'utilisateur valide sa demande.
- Le système enregistre la demande.
- Le système authentifie l'utilisateur sur le système.
- Le système envoie un mail de confirmation à l'utilisateur.

#### *Cas utilisateur enregistré : (Utilisation « Membership » Classes ASP.Net) :*

- L'utilisateur fournit ses informations de connexion (nom d'utilisateur, mot de passe).
- L'utilisateur valide sa demande de connexion.
- Le système vérifie les informations fournies.
- Le système authentifie l'utilisateur sur le système.

Exceptions :

Post-conditions :

Besoin d'IHM (Interface Homme Machine) :

- L'acteur peut à tout moment revenir à la page d'accueil.

Exigences non fonctionnelles :

*Confidentialité* : aucune.

*Fréquence du cas d'utilisation* : Plusieurs fois par jour.

*Temps de réponse* : - de 5sc.

*Disponibilité* : 7 jours par semaine 24/24h.

*Concurrence* : 50 personnes.

*Intégrité* : 1.

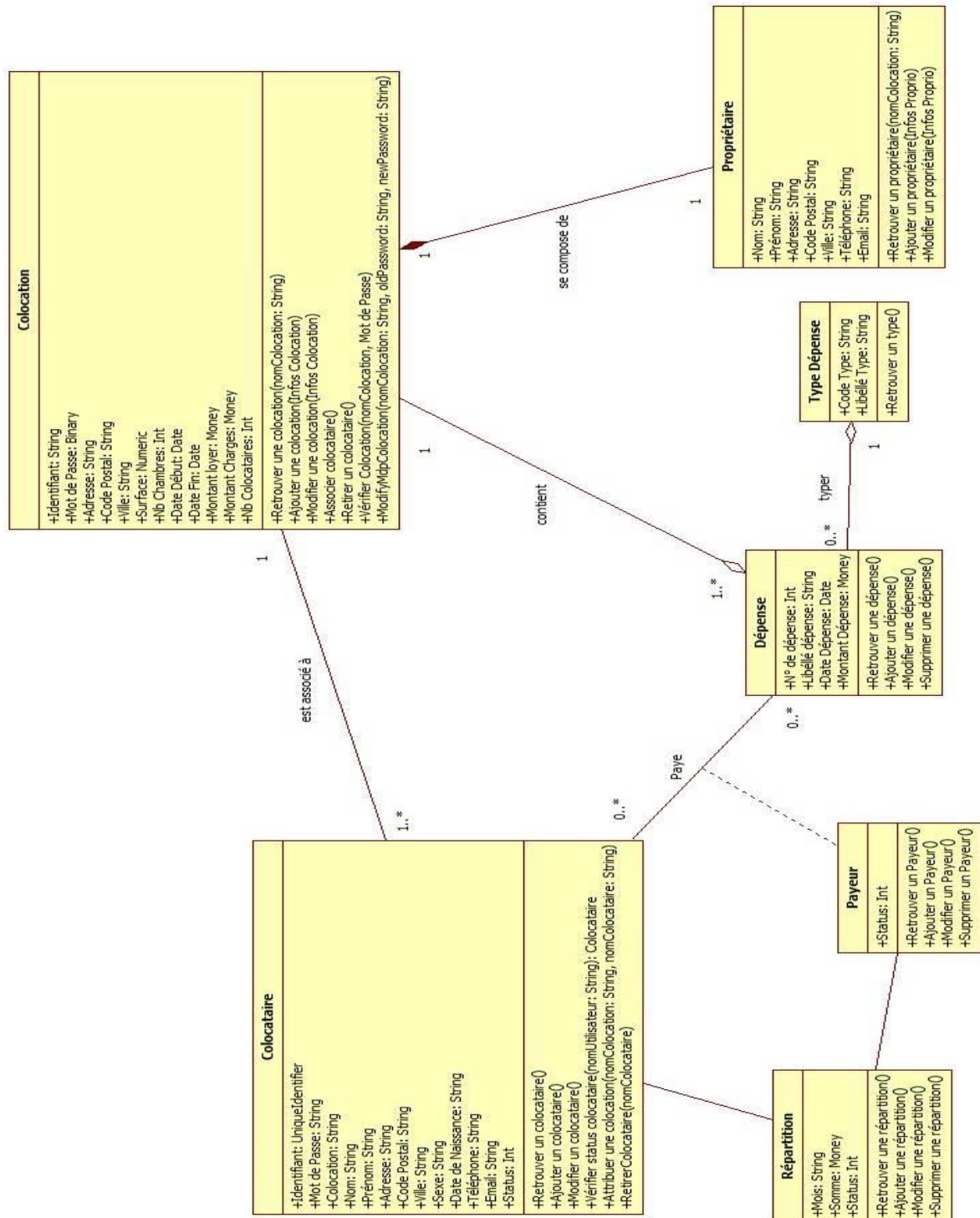
*Volumétrie* : 500 par jour.

## V. Analyse

### 1. Découpage en catégories :

Ce projet ne nécessite pas de découpage en catégories car il y'a peu de classes et une seule catégorie est présentée.

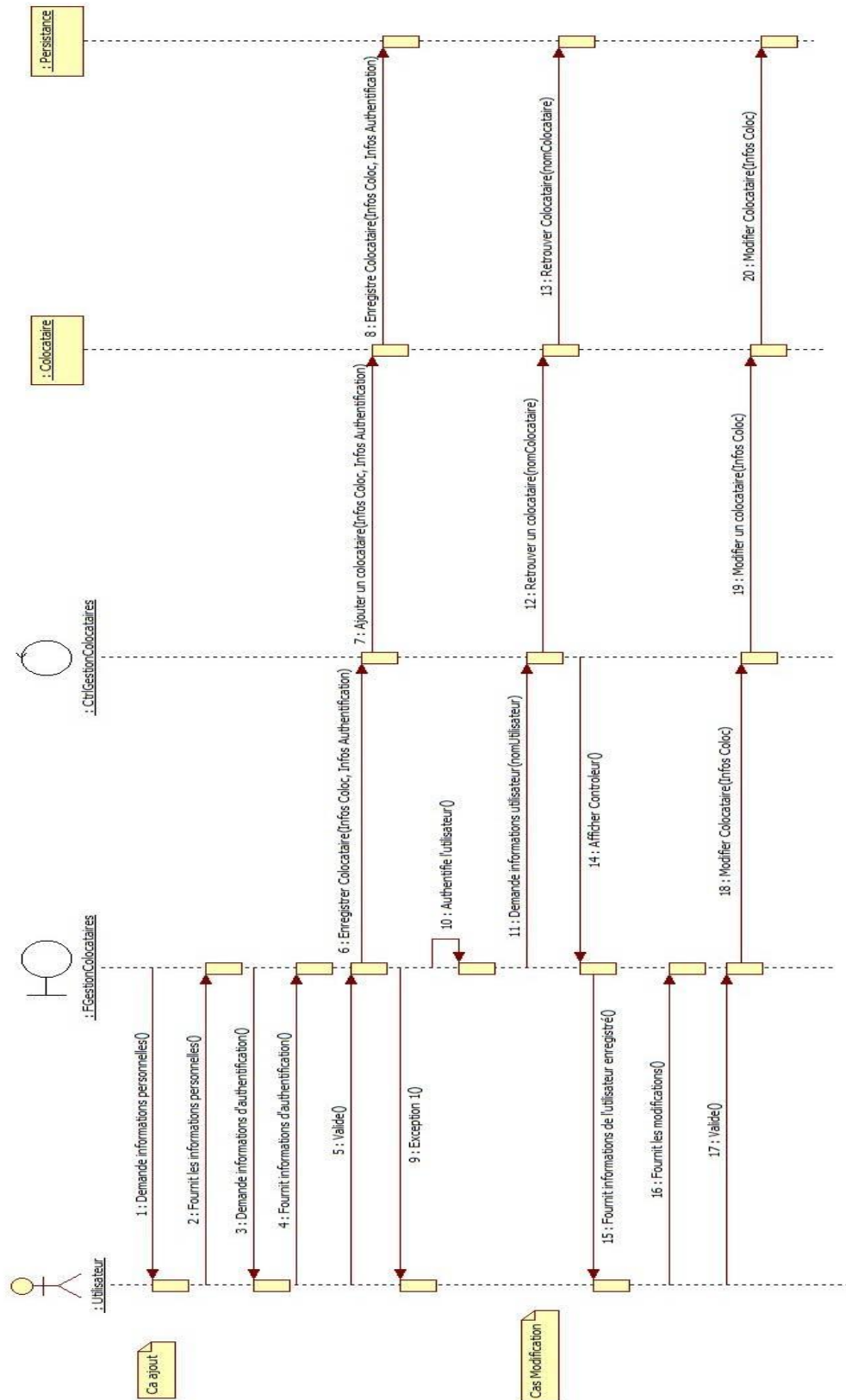
### 2. Développement du modèle statique :



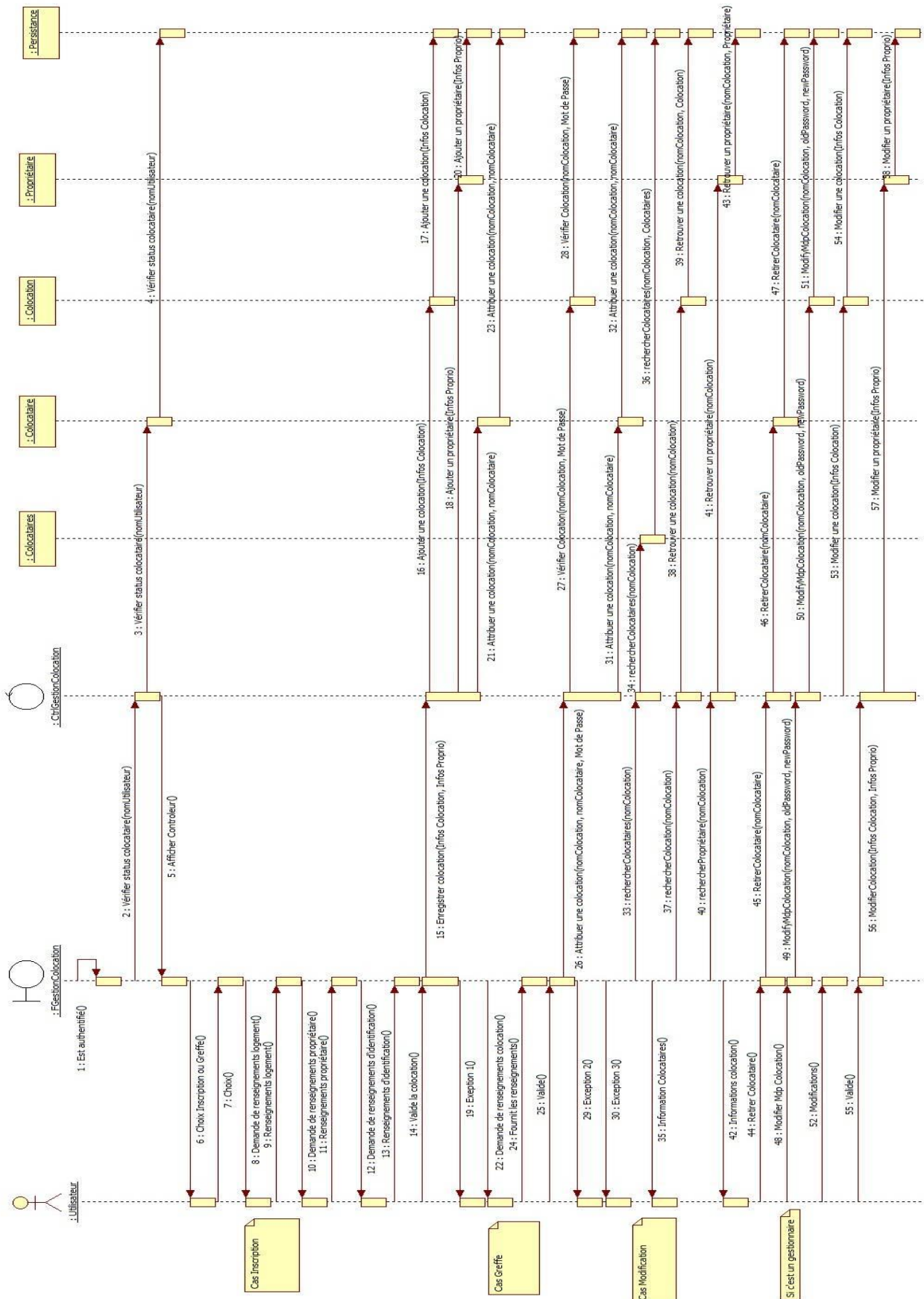


## VI. Conception Préliminaire

### Cas d'utilisation 1 : Gestion de colocataires



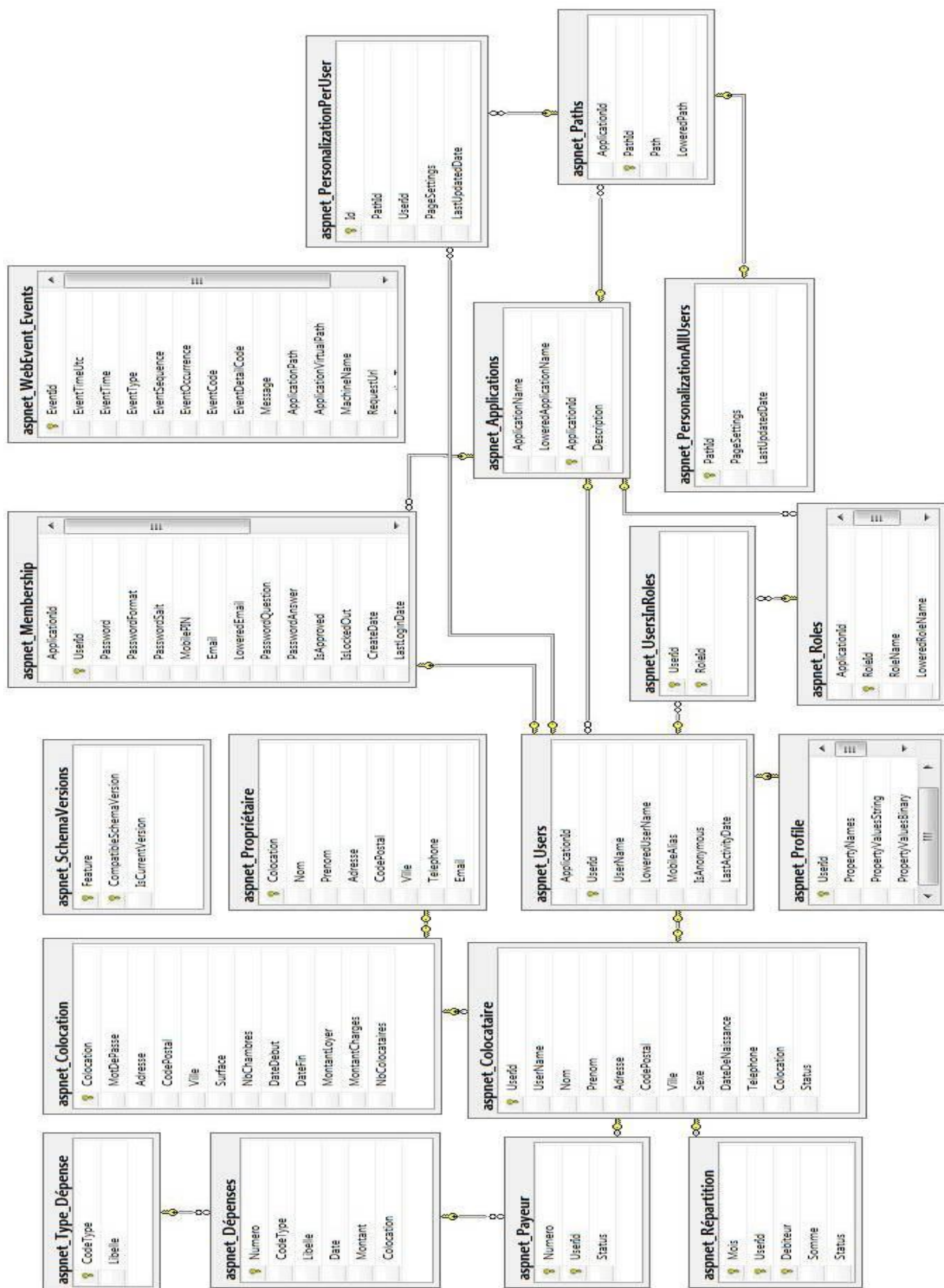
## Cas d'utilisation 2 : Gestion de colocation





## **VII. Conception Détaillée**

1. Schéma de la base de données (avec Membership Provider ASP.NET) :



## 2. Procédures stockées :

### Procédure 1 : Création d'un utilisateur

dbo.aspnet\_Register

```
@UserId uniqueidentifier,  
@UserName varchar(50),  
@Nom varchar(50),  
@Prenom varchar(50),  
@Adresse varchar(50),  
@CodePostal varchar(10),  
@Ville varchar(50),  
@Sexe nchar(1),  
@DateDeNaissance varchar(10),  
@Telephone varchar(20),  
@Status int = 0
```

AS

BEGIN

SET NOCOUNT ON

```
INSERT INTO dbo.aspnet_Colocataire (UserId, UserName,  
Nom, Prenom, Adresse, CodePostal, Ville, Sexe, DateDeNaissance, Telephone,  
Status)
```

```
VALUES (@UserId, @UserName, @Nom, @Prenom,  
@Adresse, @CodePostal, @Ville, @Sexe, @DateDeNaissance, @Telephone,  
@Status)
```

RETURN

END

## Procédure 2 : Modifier un utilisateur

dbo.aspnet\_Update\_User

```
@UserName varchar(50),  
@Nom varchar(50),  
@Prenom varchar(50),  
@Adresse varchar(50),  
@CodePostal varchar(10),  
@Ville varchar(50),  
@Sexe nchar(1),  
@DateDeNaissance varchar(10),  
@Telephone varchar(20)
```

AS

BEGIN

SET NOCOUNT ON

```
UPDATE dbo.aspnet_Colocataire SET Nom = @Nom, Prenom =  
@Prenom, Adresse = @Adresse, CodePostal = @CodePostal,  
Ville = @Ville, Sexe = @Sexe, DateDeNaissance =  
@DateDeNaissance, Telephone = @Telephone WHERE UserName = @UserName
```

RETURN

END

## Procédure 3 : Création de colocation

```

dbo.aspnet_Create_Colocation

    @Colocation varchar(10),
    @MotDePasse varchar(50),
    @Adresse varchar(50),
    @CodePostal varchar(10),
    @Ville varchar(50),
    @Surface numeric(18, 0),
    @NbChambres int,
    @DateDebut datetime,
    @MontantLoyer money,
    @MontantCharges money,
    @NbColocataires int

AS
BEGIN

    SET NOCOUNT ON

    DECLARE @PassPhrase nvarchar(50)
    SET @PassPhrase = N'cow'

    DECLARE @MotDePasseModifie nvarchar(50)
    SET @MotDePasseModifie = N''
    SELECT @MotDePasseModifie = @MotDePasseModifie +
@MotDePasse

    DECLARE @EncryptedPassword varbinary(256)
    SET @EncryptedPassword = EncryptByPassPhrase(@PassPhrase
, @MotDePasseModifie, NULL)

    INSERT INTO dbo.aspnet_Colocation (Colocation,
MotDePasse, Adresse, CodePostal, Ville, Surface, NbChambres, DateDebut,
MontantLoyer, MontantCharges, NbColocataires)
        VALUES (@Colocation, @EncryptedPassword, @Adresse,
@CodePostal, @Ville, @Surface, @NbChambres, @DateDebut, @MontantLoyer,
@MontantCharges, @NbColocataires)

    RETURN

END

```

#### Procédure 4 : Modifier Colocation

```

dbo.aspnet_Update_Colocation

    @Colocation varchar(10),

```

```

        @Adresse varchar(50),
        @CodePostal varchar(10),
        @Ville varchar(50),
        @Surface numeric(18, 0),
        @NbChambres int,
        @MontantLoyer money,
        @MontantCharges money,
        @NbColocataires int

AS

BEGIN

    SET NOCOUNT ON

    UPDATE dbo.aspnet_Colocation SET Adresse = @Adresse,
CodePostal = @CodePostal, Ville = @Ville, Surface = @Surface, NbChambres =
@NbChambres,
MontantLoyer = @MontantLoyer, MontantCharges =
@MontantCharges, NbColocataires = @NbColocataires WHERE Colocation =
@Colocation

RETURN

END

```

#### Procédure 5 : Modifier Propriétaire

```

dbo.aspnet_Update_Propriétaire

        @Colocation varchar(10),
        @Nom varchar(50),
        @Prenom varchar(50),
        @Adresse varchar(50),
        @CodePostal varchar(10),
        @Ville varchar(50),
        @Telephone varchar(20),
        @Email varchar(50)

AS

BEGIN

    SET NOCOUNT ON

    UPDATE dbo.aspnet_Propriétaire SET Nom = @Nom, Prenom =
@Prenom, Adresse = @Adresse, CodePostal = @CodePostal,
Ville = @Ville, Telephone = @Telephone, Email =
Email WHERE Colocation = @Colocation

RETURN

END

```

#### Procédure 6 : Récupérer Id

```

dbo.aspnet_GetId

        @Colocation varchar(50)

```



```

AS
    BEGIN

        SET NOCOUNT ON

        DECLARE @PassPhrase nvarchar(50)
        SET @PassPhrase = N'cow'

        SELECT CONVERT(nvarchar, DecryptByPassphrase(@PassPhrase,
MotDePasse, NULL)) AS 'DecryptedPassword' FROM dbo.aspnet_colocation WHERE
Colocation = @Colocation

        RETURN

    END

```

### Procédure 7 : Modifier Mot de Passe

```

dbo.aspnet_Modify_Pass

    @Colocation varchar(10),
    @MotDePasse varchar(50)

AS
    BEGIN

        SET NOCOUNT ON

        DECLARE @PassPhrase nvarchar(50)
        SET @PassPhrase = N'cow'

        DECLARE @MotDePasseModifie nvarchar(50)
        SET @MotDePasseModifie = N''
        SELECT @MotDePasseModifie = @MotDePasseModifie +
@MotDePasse

        DECLARE @EncryptedPassword varbinary(256)
        SET @EncryptedPassword = EncryptByPassphrase(@PassPhrase
, @MotDePasseModifie, NULL)

        UPDATE dbo.aspnet_Colocation SET MotDePasse =
@EncryptedPassword WHERE Colocation = @Colocation

        RETURN

    END

```

### FONCTIONALITÉS ENLEVÉES :

- Ajout Colocataire Manuellement.
  - Finalement, inutile, chaque colocataire décide de se greffer lui-même. On ne peut pas ajouter un utilisateur enregistré du site manuellement comme colocataire sans son consentement.

Voir validation par email du dit utilisateur...
- Suppression Colocation
  - Pressé par le temps, développement prévu...
- Supprimer / Modifier un Colocataire ou une colocation
  - Seul le gestionnaire peut supprimer des colocataires pour l'instant.
- Sortie de Colocataire en cours de mois et Calcul Associé.
  - Nécessite une autre table avec le nom du colocataire, le nom de la / les colocation(s) dont il a fait partie, et la date d'entrée et de sortie pour chacune de ces colocations. Pressé par le temps, à développer plus tard.
- Assignment et Dés assignment de Gestionnaires.
  - Pressé par le temps, développement prévu.
- Historique
  - Le suivi des dépenses depuis le début de la colocation, périodes supérieures à un mois ne sont pas implémentées encore.

### IDEES NOUVELLES :

- Forum.
- Petites annonces relié au système de gestion.
- Amélioration de l'interface.
- Gestion de compte utilisateur.
- Système de discussion interne. Salons IRC (Internet Relay Chat).
- Menu d'aide + documents légaux.
- Internationalisation.
- Idée de Bruno ??