Un Peuple – Un But – Une Foi

Ministère de l'Enseignement Supérieur et de la Recherche
Université Gaston Berger de Saint Louis
UFR Science Appliquée à la Technologie

RAPPORT DU PROJET D'APPLICATION MOBILE

Présenté par :

Amassagou Michael Saye

Sous la tutelle de :

Pr. Ousmane Sall

Enseignant Chercheur

Année académique 2020-2021

PLAN DU RAPPORT

Introduction générale

L'entreprise Esther- Philippe souhaite mettre en place une application mobile qui est surnommée **Magic Ciné** SG… qui permettra à ses utilisateurs de faire une réservation facile de billet afin d'éviter la queue aux guichets.

Pour cela, l'administrateur du système doit insérer toutes les informations nécessaires dans la base de données : films ; salles ; places ; projections ; billets.

Il a aussi la possibilité de modifier, supprimer les informations citées ci-dessus pour le bon fonctionnement du système.

Le système permet aux utilisateurs de consulter le planning afin de se renseigner sur les films qui seront projetés dans les différentes salles ainsi que les places disponibles.

Un utilisateur quelconque peut consulter les plannings sans pour autant s'authentifier mais en ce qui concerne la réservation du billet, il doit mettre des informations nécessaires.

Quant aux administrateurs, ils devront utiliser toutes les fonctionnalités du système.

Les acteurs du système sont : Utilisateur, client ; administrateur.

Les clients et les administrateurs sont considérés comme les utilisateurs.

Reformulation des besoins

L'entreprise Esther Philippe souhaite mettre en place la gestion des cinémas à travers un système de réservation de billet par internet :

- Chaque cinéma se trouvant dans une ville est définie par son code, son nom et sa position géographique ;
- Le cinéma contient un ensemble de salles ;
- Chaque salle qui est définie par son numéro, son nom, contient un ensemble de places ;
- Chaque place a un numéro et positionnée géographiquement ;
- Quotidiennement, on programme plusieurs films projections de films dans les salles ;
- Chaque projection se déroule dans une séance, concerne un film, et se déroule dans une salle à une date de projection et un prix fixe ;
- Chaque séance est définie par son numéro et l'heure de début de la séance ;
- Pour chaque projection on prévoit un ensemble de billets ;
- Chaque billet concerne une place et défini par le nom du client, le prix du billet, et le code de paiement ;

L'application se compose de deux (2) parties :

La partie **backend** et la partie **frontend.**

- **Les exigences fonctionnelles de l'application sont** :
    - Gestion des cinémas (consultations, saisie, ajout, édition, mise à jour et suppression) ;
    - Gestion des salles et des places ;
    - Gestion films ;
    - Gestion projection ;
    - Gestion des ventes et des billets ;
    - A chaque fois qu'une fonctionnalité est sollicité le système interagit avec la base de données ;

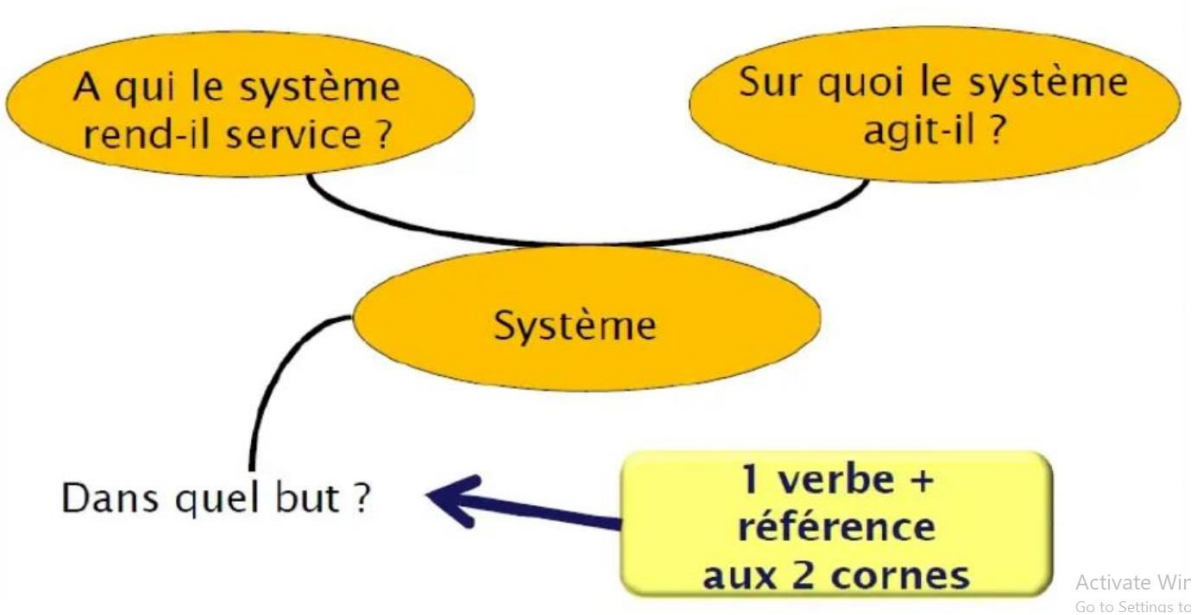
- La partie **backend** est basée sur **spring** et se compose des couches **DAO**, **service(métier) et web** ;
    - La couche **DAO** est basée sur **spring DATA, JPA, Hibermate** ;
    - La couche **métier** est définie par une interface et une implémentation, quelques spécifications fonctionnelles qui nécessites des calculs ou des traitements particuliers ;
    - La couche **web** est basée sur des **API Restful** basée sur **Spring DATA Rest** ou un **Restcontroller** ;
- La partie **Frontend** est basée sur **Framework Angular** ;
    - La **sécurité** est basée sur **Spring security et Json Web Token** ;
    - Les données seront stockées dans un système de gestion de base de données **MYSQL**.

Analyse fonctionnelle classique

Le but de l'analyse fonctionnelle est d'optimisé la conception ou la ré-conception de produits en s'appuyant sur les fonctions que doit réaliser le produit. Enfin d'éviter certains pièges classiques de la conception et Rendre possible un dialogue entre les intervenants du projet. (Tirer du cours numéro 4 Analyse Fonctionnelle de Professeur Diattara).

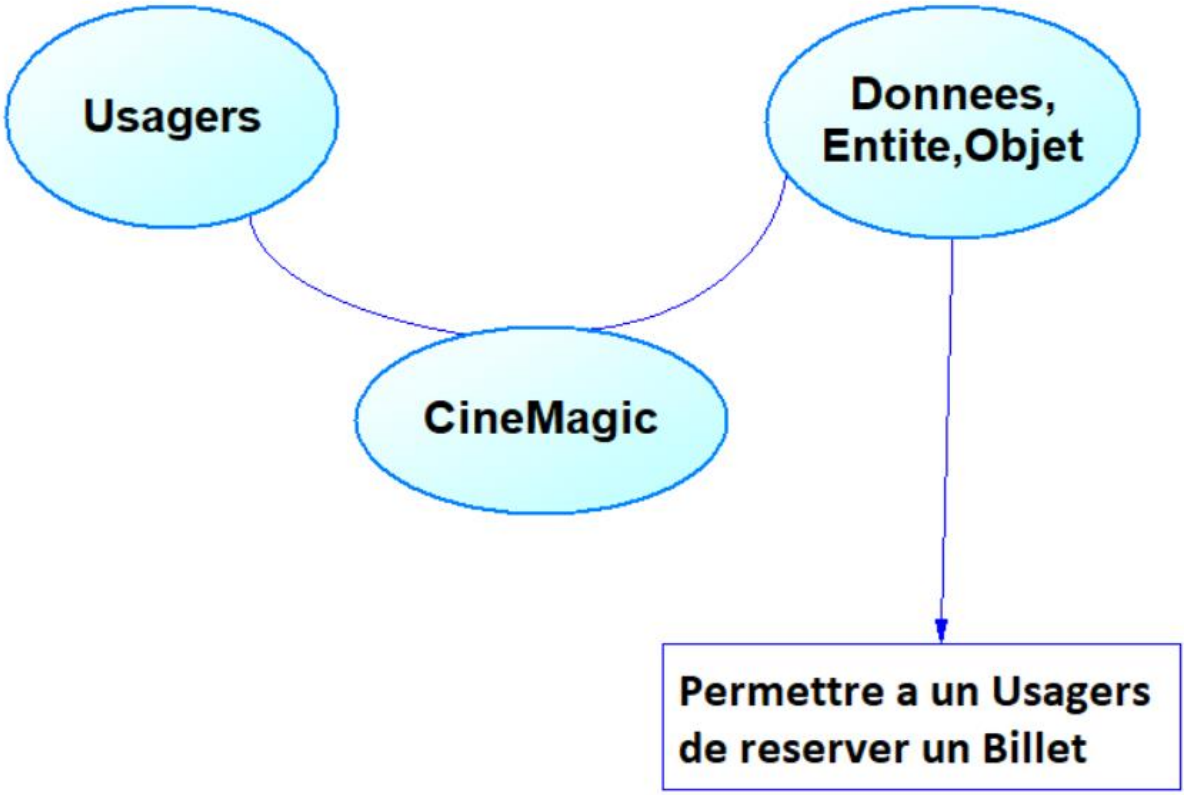
Diagramme de Bête a Cornes

Pour faire un bon produit, il faut avoir identifié le vrai **besoin et** l'objectif de diagramme bête à cornes a pour but d'identifier le besoin.



Voici le **diagramme Bête à Cornes** qui identifie notre produit :

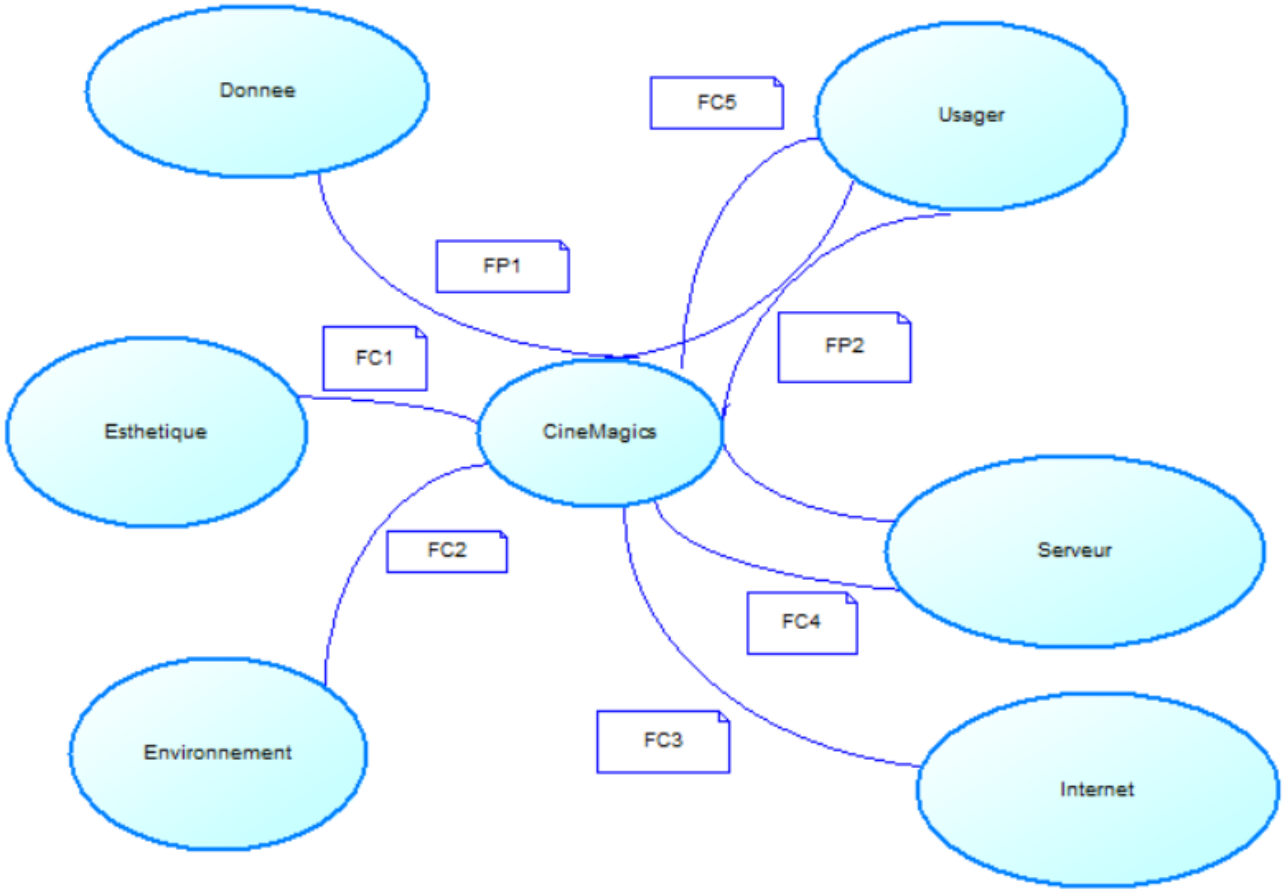
A qui le produit rend – il service ? **Sur quoi agit le système ?**



Dans quel but le système existe-il ?

Diagramme de Pieuvre

Le besoin étant spécifié, il faut maintenant recenser toutes les **fonctions** que le produit doit satisfaire. Pour cela il faut recenser deux types de fonctions : **Fonction Principale(FP) et Fonction Contraintes(FC)**



FP1 : Permettre à un usager de réserver un Billet

FP2 : Permettre à un usager de communiquer avec le Serveur

FC1 : Plaire à l'œil

FC2 : Respecter Environnement

FC3 : Accéder par internet

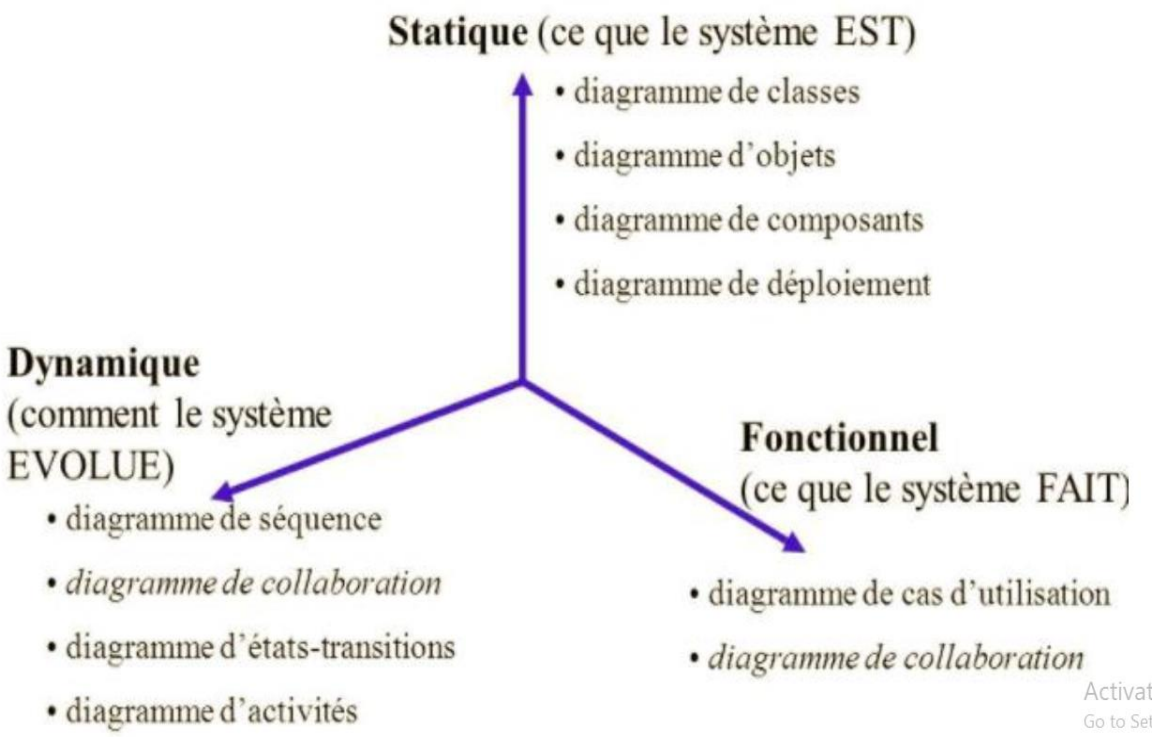
FC4 : Héberger sur un serveur

FC5 : Faciliter l'utilisation du système pour usager

Analyse Orienté Objet

L'objectif d'Approche orientée objet est fait pour identifier les objets manipulés par le système, avec leurs états et leurs comportements.

Axes de modélisation du Système :



Nous verrons que trois(3) types de modélisation qui seront décrit ci-dessous :

- **Diagramme de Cas d'Utilisation**
- **Diagramme de Classe**
- **Diagramme de Séquence**

Diagramme de Cas d'Utilisation

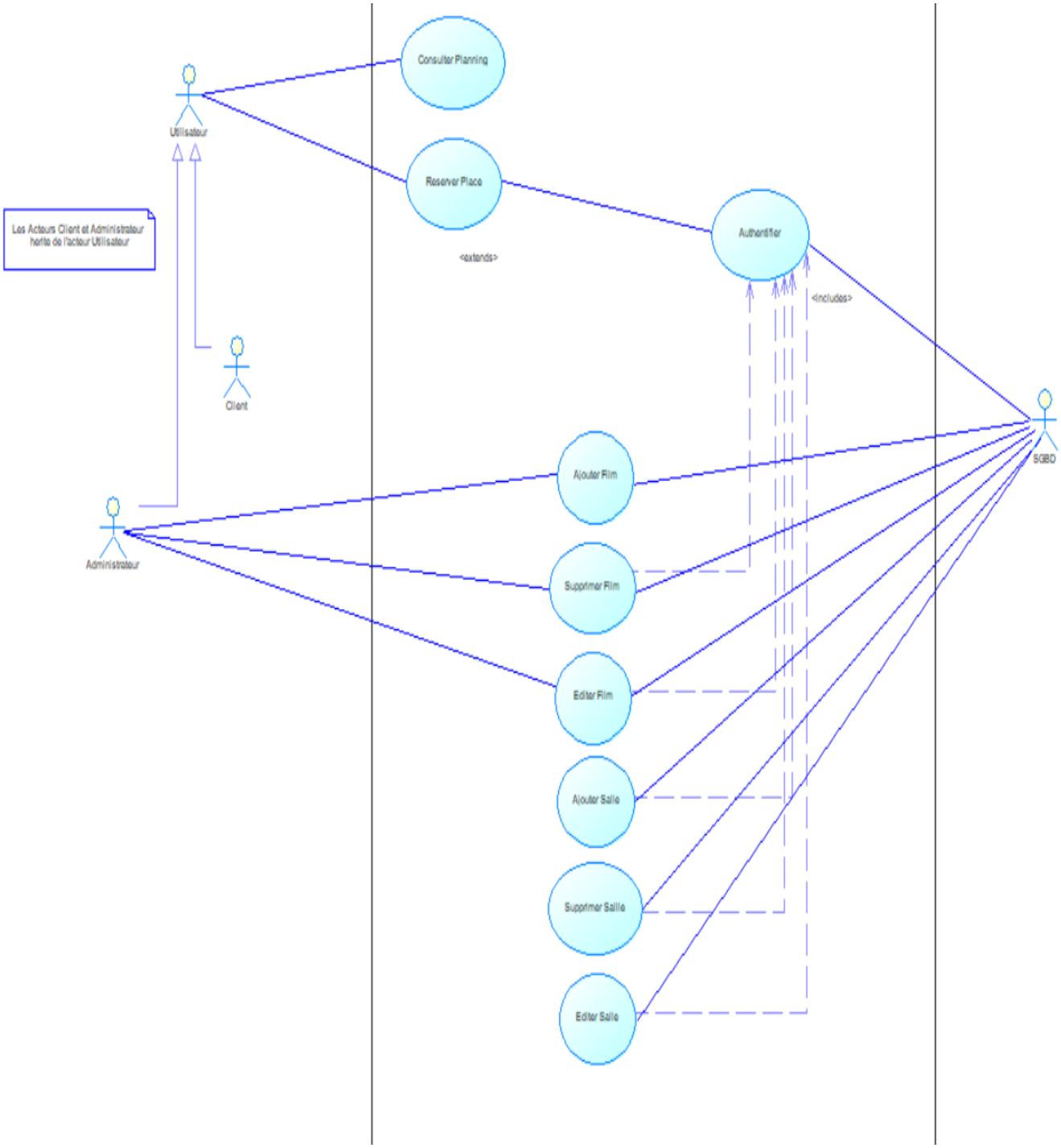
Avant de développer un système, il faut savoir **précisément** à *QUOI* il devra servir, *cad* à quels besoins il devra répondre.

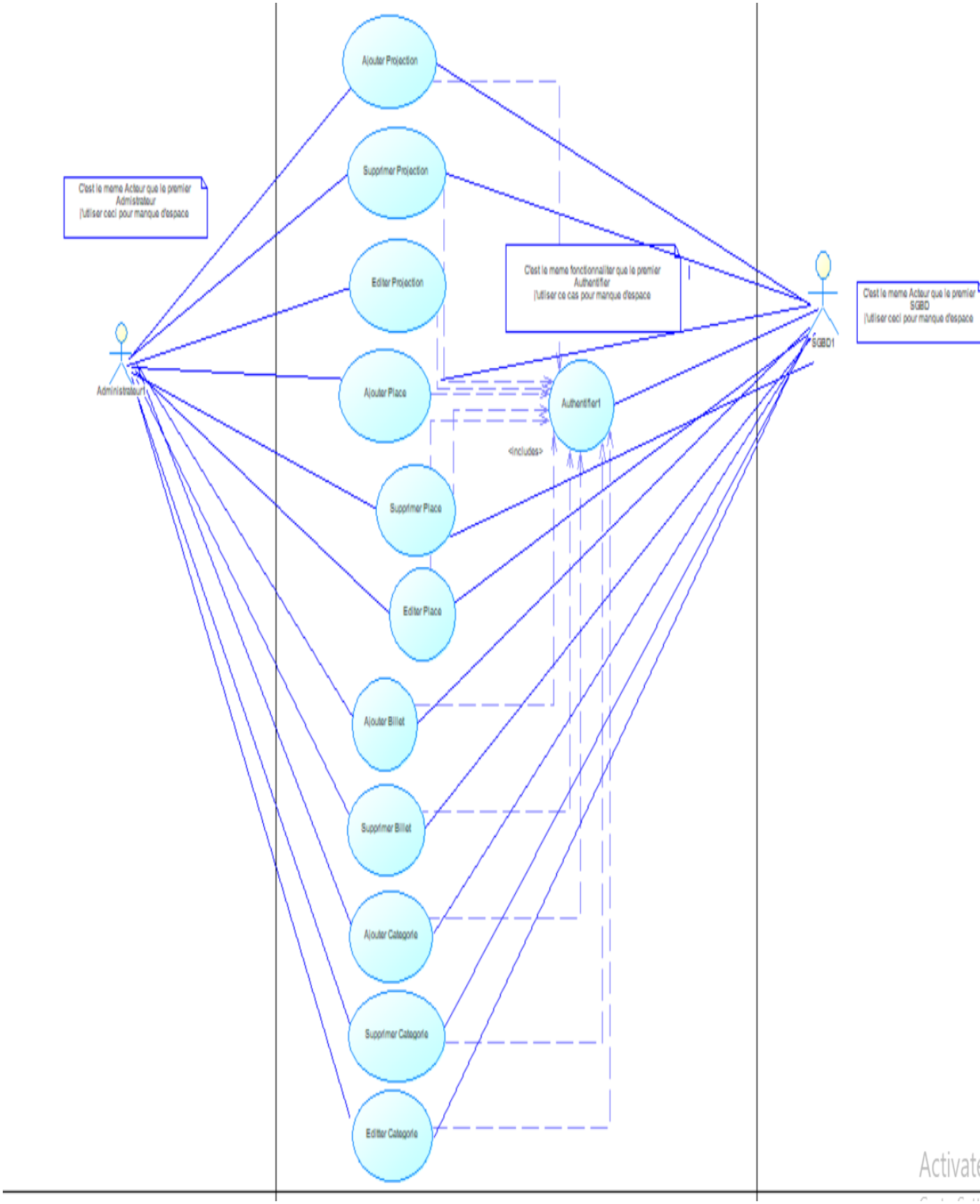
- **Modéliser les besoins** permet de :
  - Faire l'inventaire des fonctionnalités attendues ;
  - Organiser les besoins entre eux, de manière à faire apparaître des relations (réutilsations possibles, ...).
- Avec UML, on modélise les besoins au moyen de **diagrammes de cas d'utilsation**.

(Tirer du cours numéro 5 Analyse Oriente Objet de Professeur Diattara).

Voici notre **diagramme de Cas d'Utilisation** :

En raison de plusieurs cas de fonctionnalité nous étions contraint de diminuer la taille et de réaliser notre diagramme de cas d'utilisation en plusieurs étapes et nous avons fait les commentaires pour que la lecture puisse être facile. Pour cela nous allons ajouter notre modélisation effectuer sur le **POWER AMC Design** au dossier du projet envoyer pour une meilleure compréhension du diagramme.




Ajouter Projection
Supprimer Projection
Editer Projection
Ajouter Place
Supprimer Place
Editer Place
Ajouter Billet
Supprimer Billet
Ajouter Categorie
Supprimer Categorie
Editer Categorie
Authentifier1
<includes>
Administrateur1
SGBD1
C'est le meme Acteur que le premier Administrateur
Utiliser ceci pour manque d'espace
C'est le meme fonctionnalitar que le premier Authentifier
Utiliser ce cas pour manque d'espace
C'est le meme Acteur que le premier SGBD
Utiliser ceci pour manque d'espace

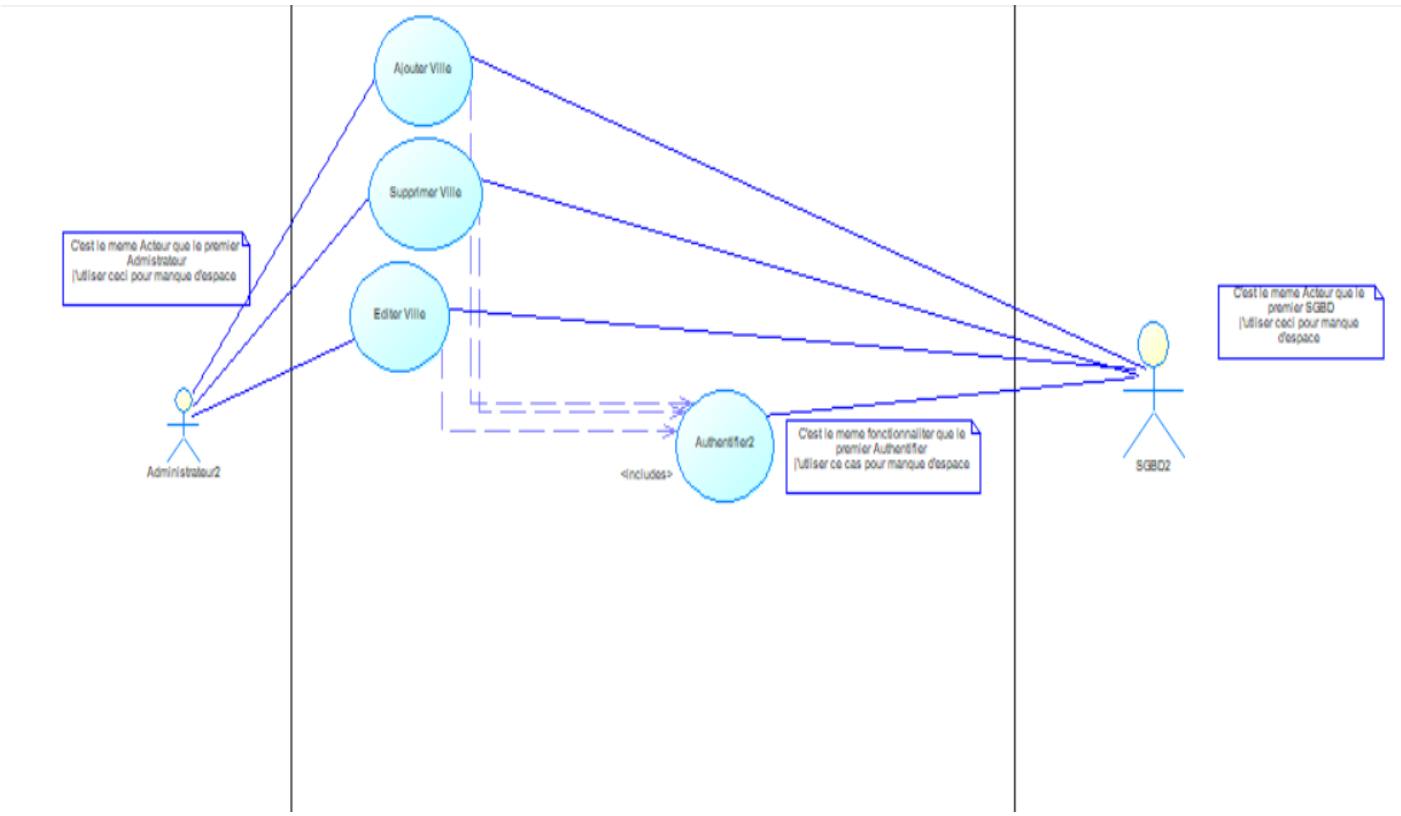


Diagramme de Classe

Le diagramme de classe est le point central dans un développement orienté objet. En analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classe représente la structure d'un code orienté objet ou, à un niveau de détail plus important, les modules du langage de développement.

Comment le représenter ?

Le diagramme de classe met en œuvre des classes, contenant des attributs et des opérations, et reliées par des associations ou des généralisations.

Bref l'objectif de Diagramme de Classe Permet de définir l'ensemble des classes et leurs relations d'une application.

,

Voici notre **Diagramme de Classe** :

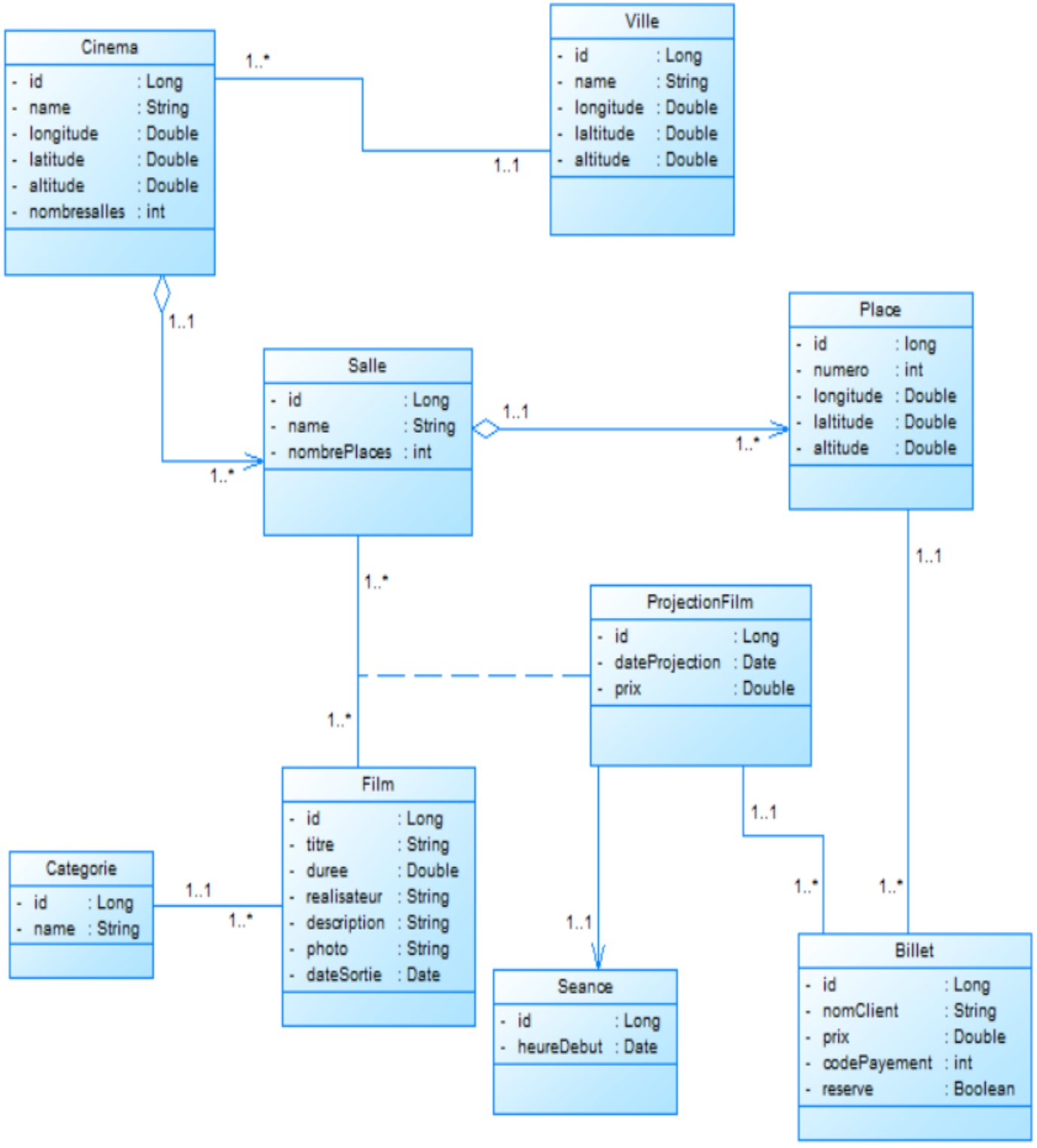


Diagramme de Séquence

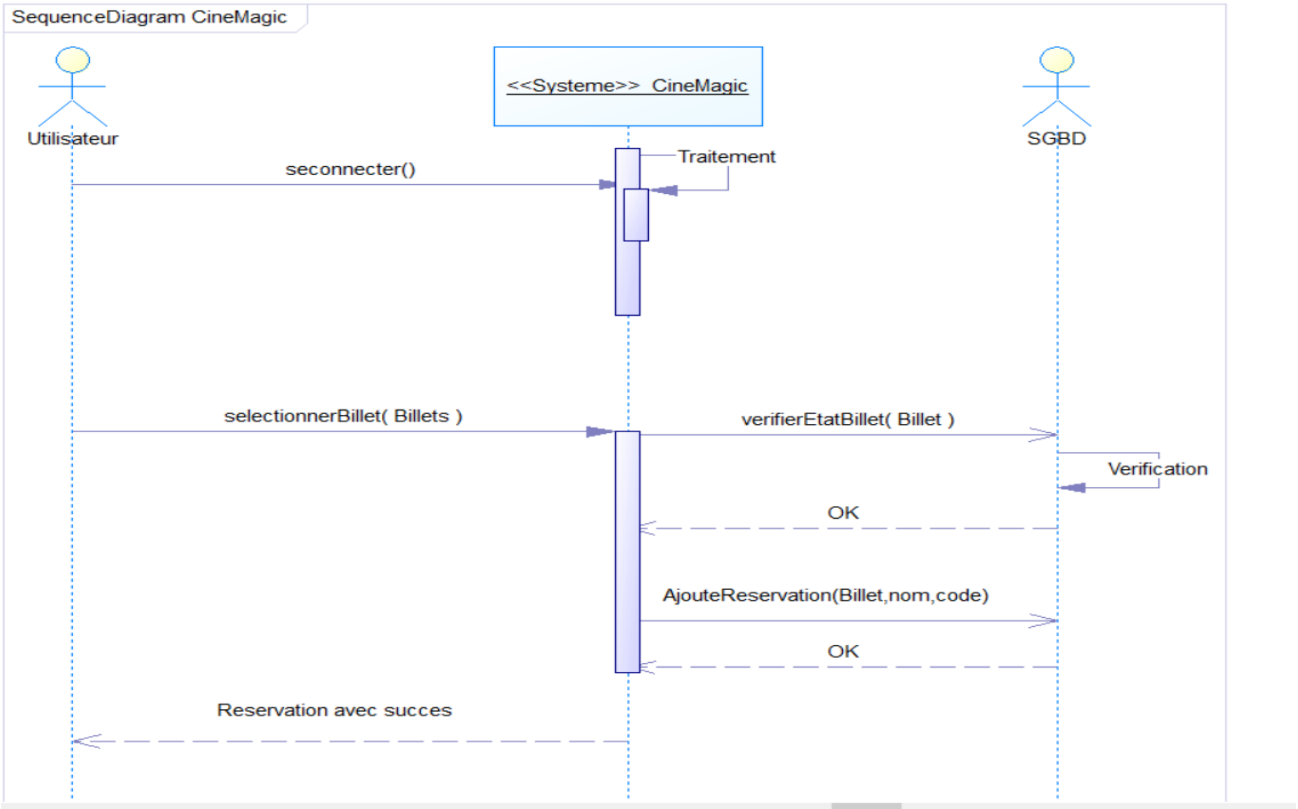
- Décrit la dynamique d'une sous-fonction du système.
- La dynamique globale d'un système nécessite plusieurs diagrammes de séquences.
- Modélise les messages échangés entre objets en mettant l'accent sur la chronologie des messages.
- Décrit les interactions entre un groupe d'objets en montrant, de façon séquentielle, les envois de message qui interviennent entre les objets.

nous utilisons la convention graphique suivante :

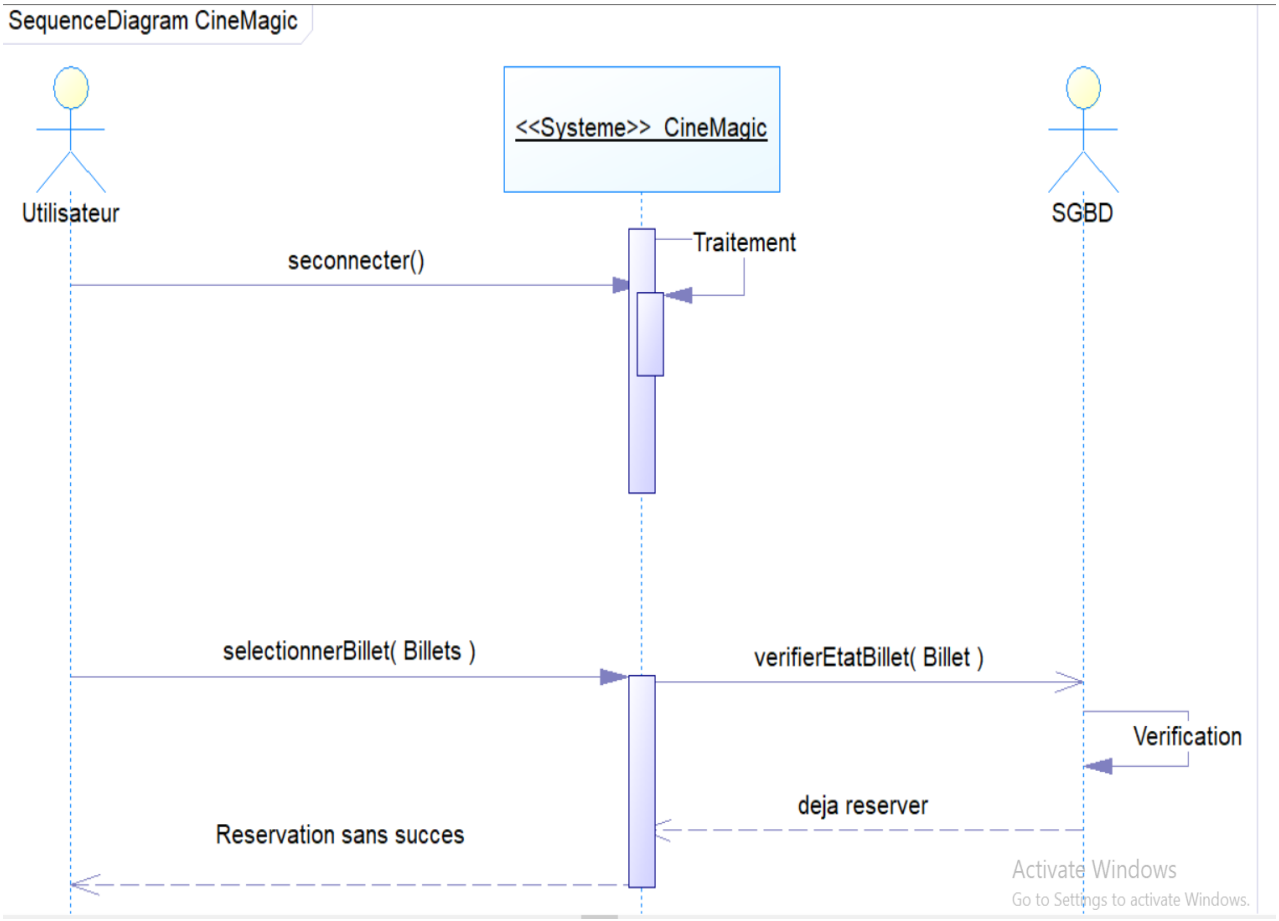
- l'acteur principal **Appelant** à gauche ;
- une ligne de vie représentant au milieu ;
- l'acteur secondaire **Standard** à droite

Voici le diagramme de séquence système scenario nominal **réservation d'un billet par un utilisateur :**

- **Réservation avec succès**



- **Réservation sans succès**



Conclusion

En définitive la réalisation de ce projet a été possible par la prise en compte de certains paramètres tel que le génie logiciel ; l'architecture JEE ; Framework flutter etc.

Notons que le principe du génie logiciel permet à un développeur de faire la conception d'une application mobile grâce à ses principes nous pouvons facilement identifier le besoin et le valider ensuite nous pouvons détecter les fonctions (principales et contraintes).

IL permet aussi de modéliser le système à travers des diagrammes. Ces diagrammes permettent de mieux comprendre les fonctionnalités du système pour pouvoir bien les réalisés mais il faut savoir que la faisabilité de tous les diagrammes est difficile mais les trois principaux diagrammes nécessaires sont obligatoires dont nous citons entre autres :

- Le diagramme de cas d'utilisation : permet de modéliser les besoins c'est-à-dire Identifier les acteurs et les fonctions du système
- Le diagramme de classe : permet quant à lui de définir l'ensemble des classes et leur relation.
- Le diagramme de séquence : permet de représenter graphiquement les interactions entre les acteurs et le système selon un ordre chronologique.

Pour pouvoir réaliser mon application j'ai dû choisir l'architecture JEE (pour la backend) car je suis un novice en la matière afin de pouvoir apprendre les principes de l'architecture JEE qui sont assez volumineux et je crois que tout ingénieur développeur doit avoir ne serait-ce quelques bases : Spring Data JPA ; Rest Repositories ; Lombok ; Spring Boot Dev/Tools etc.

Concernant le Framework flutter (pour le frontend) j'ai eu à tirer quelque notions très intéressantes grâce à la réalisation de ce projet, c'est la technologie qui est en actualité.

En fin la réalisation de ce projet m'a permis de concevoir beaucoup de notions en développement d'application Mobile.