Conception et développement d'une plateforme d'hébergement d'événements

[CircusPlanner]

Équipe projet [E2] 17

- Nicolas
- Johan
- Rostagnat Aline
- Thomas
- Mathias

Table des matières

Consignes et évaluation du dossier de conception

Ce dossier de conception est un template à partir duquel vous pouvez vous inspirer afin de réaliser votre dossier d'analyse et de conception. Les indications entre crochets sont là pour vous aider à structurer votre dossier. Elles sont à retirer dans le rendu final. Seront évalués :

- La pertinence et la validité des modèles créés ;
- La conformité des modèles entre eux (par exemple le diagramme de séquence détaillé par rapport aux scénarios nominaux) ;
- La conformité du code par rapport aux modèles créés ;
- Le respect de la notation UML ;
- La clarté et la qualité du dossier de conception, y compris de l'introduction, de la conclusion, et de la description que vous ferez des différents modèles.

Quelques conseils afin d'améliorer la clarté de votre dossier :

- Ne pas se contenter d'insérer des images des modèles créés, mais les accompagner d'explications textuelles (même brèves) ;
- Apporter une attention particulière à la clarté des diagrammes réalisés :
    - ☑ ~~Vérifier la résolution et la lisibilité des diagrammes ;~~
    - ☑ ~~Vérifier que la notation UML est parfaitement respectée ;~~
    - ☑ ~~Ne pas hésiter à éclater un diagramme en deux s'il s'avère trop gros ;~~
    - ☑ ~~Le flot d'interaction entre un diagramme de séquences détaillé et un scénario de haut niveau doivent correspondre ;~~
    - ☑ ~~Supprimer le fond bleu dans Visual Paradigm.~~
- Vérifier que toutes les informations sont présentes dans les modèles :
    - ☑ ~~Chaque association doit faire apparaître un nom de rôle et une multiplicité au bout de l'association ;~~
    - ☑ ~~Si une association est bi-directionnelle, alors deux noms de rôle et deux multiplicités ;~~
    - ☑ ~~Une association qualifiée doit faire référence à un attribut de la classe sur laquelle pointe l'association ;~~
    - ☑ ~~Toutes les méthodes dans un diagramme de séquences doivent apparaître dans le diagramme de classes détaillé ;~~
    - ☑ ~~Une ligne de vie dans un diagramme de séquences correspond à un objet et non une classe ;~~

Conserver ce paragraphe dans le rendu final et cocher les cases qui auront été réalisées/vérifiées.

Introduction

Notre projet consiste en la création d'une application de création d'évènements à destination des cirques ambulants dans le but de leur permettre de planifier leurs tournées rapidement et de minimiser les imprévus. Pour se faire nous allons dans premier temps procéder à la conception de notre plateforme en dégageant les cas d'utilisations puis les architectures détaillées qui en découlent. Pour ce faire, nous avons choisi de se partager la tâche de la manière suivante : Aline et Johan se chargeront de la descriptions des cas d'utilisations déjà mis au point en amont, Thomas s'occupera de la prioritisation des cas d'utilisation ainsi que de la mise au point de l'architecture proposée et de l'architecture détaillée avec l'aide de Nicolas, Mathias et Johan. Pour finir, Aline se chargera de la rédaction de l'introduction ainsi que de la conclusion de ce document.

Expression des besoins et Analyse

Description des cas d'utilisation

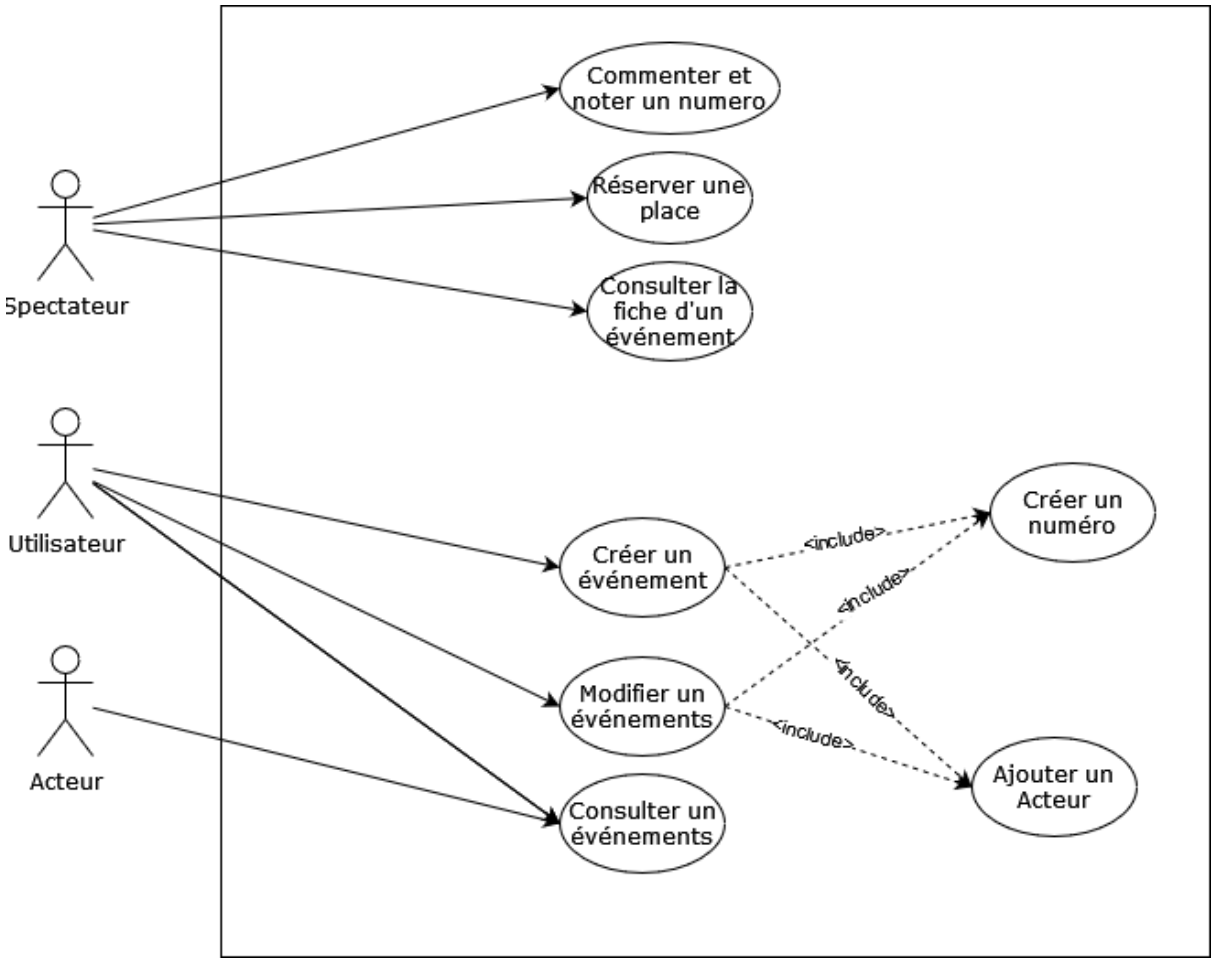

Figure 1. Diagramme de cas d'utilisation de la plate-forme d'hébergement d'événements

Nom du cas d'utilisation	UC1

Libellé du cas d'utilisation	Créer un événement
Description	Le gérant de cirque demande à la plate-forme d'hébergement d'événements de créer un nouvel événement identifié par un nom, une date de début, et une date de fin.

Nom du cas d'utilisation	UC1.1
Libellé du cas d'utilisation	Créer un spectacle
Description	Le gérant de cirque demande à la plate-forme d'hébergement d'événements de créer un nouveau spectacle identifié par un nom.

Nom du cas d'utilisation	UC1.2
Libellé du cas d'utilisation	Créer un numéro
Description	Le gérant de cirque demande à la plate-forme d'hébergement d'événements de créer un nouveau numéro identifié par un nom.

Nom du cas d'utilisation	UC1.3
Libellé du cas d'utilisation	Ajouter un acteur
Description	Le gérant de cirque peut ajouter un acteur et ses attributs à l’application

Nom du cas d'utilisation	UC2
Libellé du cas d'utilisation	Modifier un événement déjà enregistré
Description	Le gérant de cirque peut modifier son événement s'il rencontre des problèmes. Un modérateur peut modifier les informations si cela va à l'encontre des CGU.

Nom du cas d'utilisation	UC3
Libellé du cas d'utilisation	Consulter les informations d’un événement déjà enregistré
Description	Le gérant de cirque peut visualiser les événements qu’il a créé.

Nom du cas d’utilisation	UC3.1
Libellé du cas d'utilisation	Consulter la fiche d’un événements
Description	Le spectateur peut visualiser les informations publiques de l’événement

Nom du cas d'utilisation	UC4
Libellé du cas d'utilisation	Réserver une place
Description	Le spectateur peut réserver une place pour un numéro

Nom du cas d'utilisation	UC5

Libellé du cas d'utilisation	Commenter et noter un événement
Description	Le spectateur peut donner une note à un événement après l'avoir vu

Scénarios nominaux

Nom du scénario nominal	UC1.Création de l'événement
Description	1. L'utilisateur fournit à la plateforme le nom de l'événement à créer. 2. La plate-forme demande à l'utilisateur de fournir une date de début, une date de fin, et une description 3. L'utilisateur fournit ces trois informations. 4. La plate-forme crée l'événement et notifie l'utilisateur

Nom du scénario nominal	UC1.1.Création d'un spectacle
Description	1. L'utilisateur fournit à la plateforme le nom du spectacle à créer. 2. La plate-forme demande à l'utilisateur de fournir une description et optionnellement un événement auquel ajouter ce numéro. 3. L'utilisateur fournit l'information. 4. La plate-forme crée le numéro et notifie l'utilisateur.

Nom du scénario nominal	UC1.2.Création d'un numéro
Description	1. L'utilisateur fournit à la plateforme le nom du numéro à créer. 2. La plate-forme demande à l'utilisateur de fournir une description et optionnellement un spectacle auquel ajouter ce numéro. 3. L'utilisateur fournit l'information. 4. La plate-forme crée le numéro et notifie l'utilisateur.

Nom du scénario nominal	UC1.3.Ajout d'un acteur à l'application
Description	1. L'utilisateur fournit à la plateforme le nom et le prénom de l'acteur à ajouter ainsi que les numéros auxquels il participe. 2. L'application met à jour les informations.

Nom du scénario nominal	UC2.Modifier un événement déjà enregistré
Description	1. L'utilisateur se connecte à la plateforme. 2. L'application présente une liste des événements modifiables par cet utilisateur 3. L'utilisateur choisit un événement 4. L'utilisateur peut modifier chaque donnée de l'événement

	5. L’application passe l'événement en visualisation seulement lorsque la date de fin de celui-ci est passée.

Nom du scénario nominal	UC3.Consulter un événement déjà enregistré
Description	1. L'utilisateur se connecte à la plateforme. 2. L’application affiche une liste des événements accessibles depuis cet utilisateur 3. L’utilisateur choisit un événement 4. Toutes les données de l’événement sont affichées

Nom du scénario nominal	UC3.1 consulter la fiche d’un événements
Description	1. L’application affiche une liste des événements disponibles 2. Le spectateur choisit un événement 3. Certaines informations de l’événement sont affichées

Nom du scénario nominal	UC4.Réserver une place
Description	1. L'utilisateur se connecte à la plateforme. 2. Il enregistre son nom et son prenom 3. L’application Génère un id permettant de reconnaître l’utilisateur de manière unique 4. L’utilisateur doit pouvoir payer le billet de manière instantanée et sécurisée. 5. L'utilisateur peut visualiser ou télécharger le billet contenant son nom, son prénom, son id ainsi que le nom et la date de l'événement.

Nom du scénario nominal	UC5.Commenter et noter un événement
Description	1. L'utilisateur se connecte à la plateforme. 2. L’utilisateur réserve une place 3. L’application envoie un message à l’utilisateur pour lui proposer de noter l'événement à j+1 après la date de fin de celui-ci. 4. La note et le commentaire de l’utilisateur sont enregistrés par l’application..

Scénarios alternatifs

Nom du scénario alternatif	UC1.événement déjà existant
Description	À l'étape 1, si l'utilisateur donne comme nom d'événement un nom déjà utilisé, la plate-forme retourne un message d'erreur et invite l'utilisateur à donner un nouveau nom.

Nom du scénario alternatif	UC1.date d’un événement erronée
Description	À l'étape 1, si l'utilisateur donne comme date de fin d'événement une date passée ou si la date de fin se trouve avant la date de début, la plate-forme retourne un message d'erreur et invite l'utilisateur à donner de nouvelles dates.

Nom du scénario alternatif	UC1.1.numéro déjà existant
Description	À l'étape 1.1, si l'utilisateur donne comme nom de numéro un nom déjà utilisé, la plate-forme retourne un message d'erreur et invite l'utilisateur à donner un nouveau nom.

Nom du scénario alternatif	UC4.réserver un événement
Description	À l'étape 4, si l'utilisateur donne de fausses coordonnées bancaires, la plate-forme retourne un message d'erreur et invite l'utilisateur à donner de nouvelles coordonnées.

Priorisation des cas d’utilisation

Priorité	Cas d'utilisation	Justification
1	UC1, UC1.1, UC1.2, UC1.3	La fonction première de notre application est de faciliter la création d'événements.
	UC2	Il est impératif de disposer de la possibilité de modifier un événement pour une gestion optimale.
	UC3	La consultation préalable de l'événement est une étape obligatoire pour procéder à sa modification ou à son ajustement.
2	UC3.1	Compte tenu de la complexité de la différenciation des rôles, cette tâche ne sera pas considérée comme une priorité, et une unique interface sera créée pour le projet.
3 (non réalisé)	UC4	La réservation d'une place ne constitue pas une exigence fondamentale pour le responsable de l'événement.
	UC5	L'évaluation du public d'un numéro n'est pas un facteur déterminant pour l'organisateur.

Architecture proposée

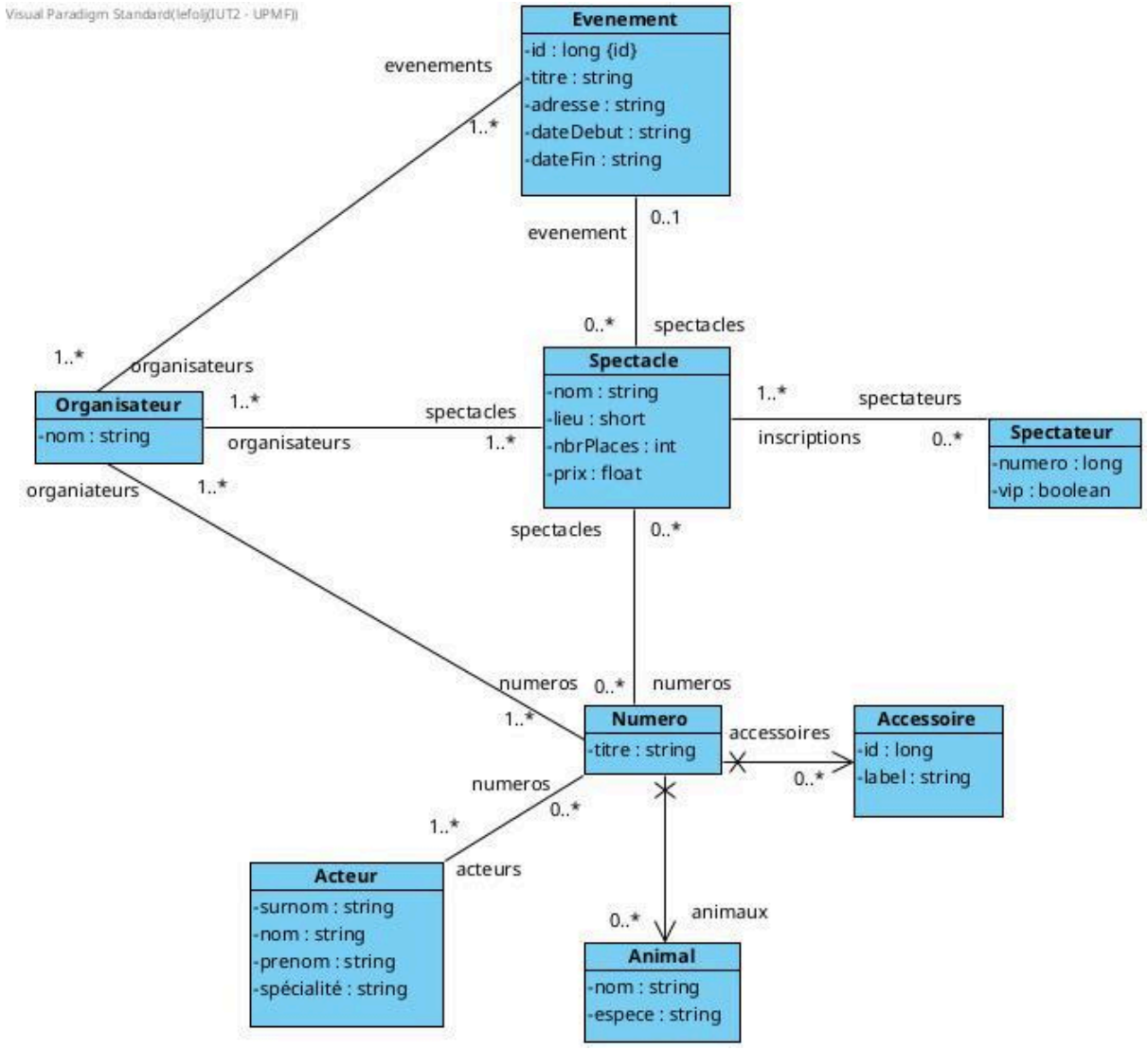

Figure 2. Diagramme de classe métier

L'organisateur est l'utilisateur principal de l'application (son interface sera donc la seule que nous coderons) mais il en existe aussi d'autres tels que les spectateurs et les acteurs. L'organisateur aura l'occasion de créer et gérer des événements, des spectacles et des numéros. Le spectateur ne pourra seulement visualiser les informations des spectacles et les acteurs pourront de la même manière visualiser les informations plus détaillées des numéros sous la seule condition d'être attitré à ceux-ci. Un événement peut être constitué de plusieurs spectacles et un spectacle de plusieurs numéros mais un numéro n'est pas forcément attribué à un spectacle et il en va de même pour un spectacle et un événement. Un numéro peut ne posséder aucun ou plusieurs animaux, accessoires et acteurs.

Exemples d'utilisation

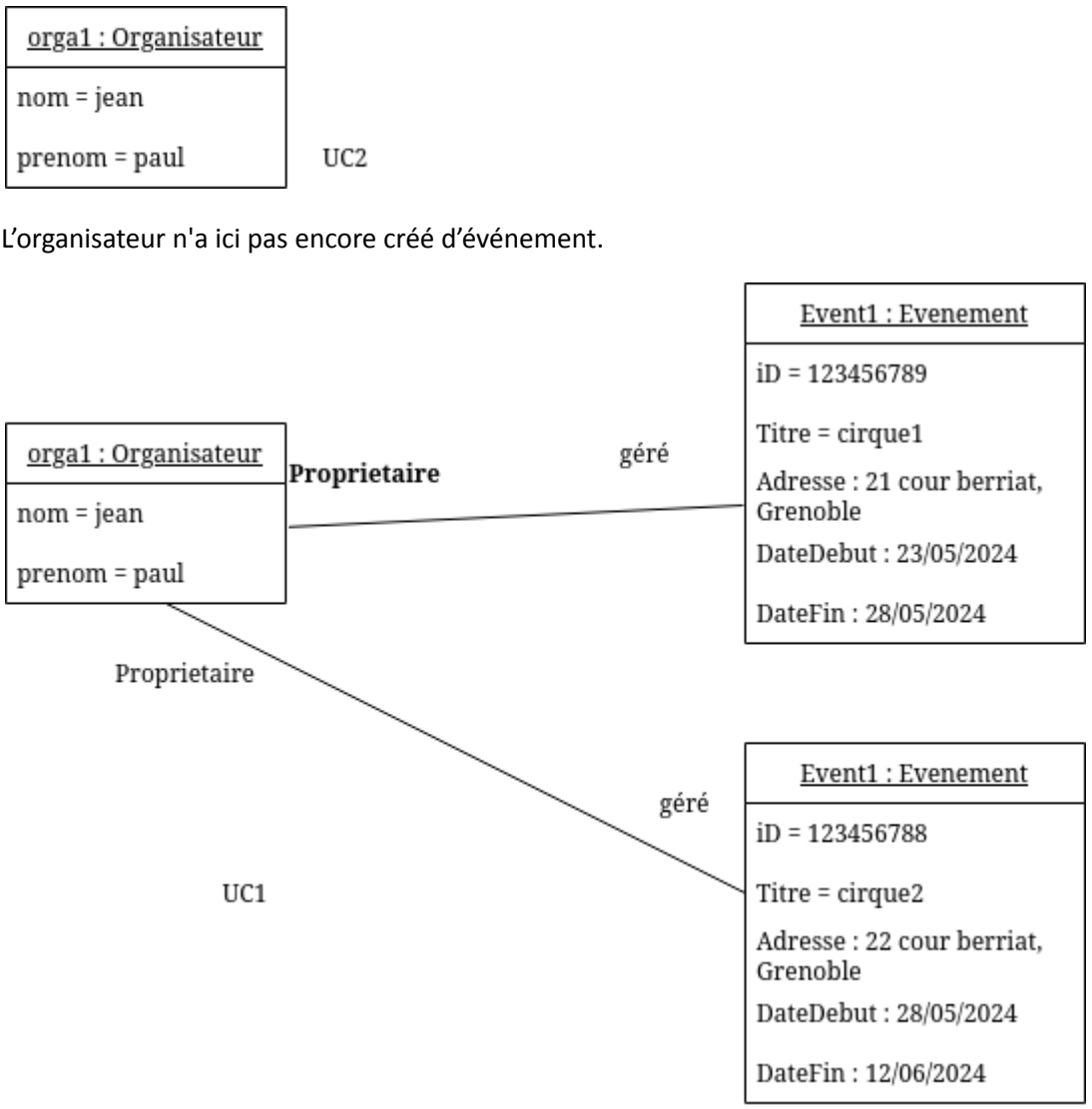

Figure 3. Diagrammes d'objet illustrant la réalisation du scénario nominal du cas d'utilisation UC1. L'image en haut présente l'état du système avant réalisation. L'image de droite présente l'état après la réalisation.

Nous pouvons voir que l'organisateur, ici Paul Jean, est en mesure de créer un ou plusieurs événement qui on eut même leurs propres informations telles que leur ID ou leur titre. Chaque événement est indépendant des autres mais est géré par l'organisateur qui l'a créé.

Conception détaillée

Architecture détaillée

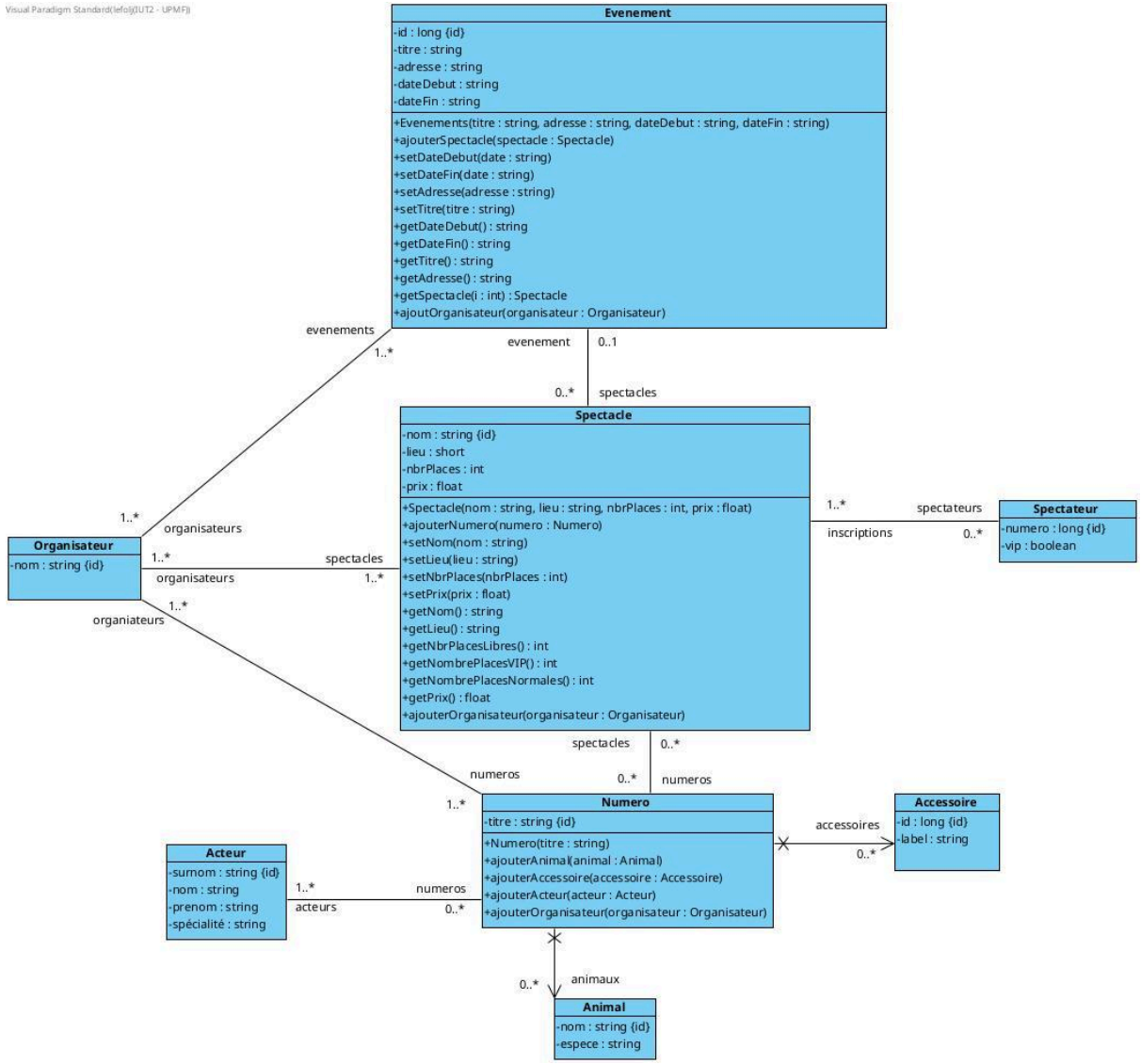

Figure 4. Diagramme de classe détaillé métier.

La plupart des classes possèdent des getteurs et setteur pour chacuns de leurs attributs afin de pouvoir les modifier facilement au besoin. Un événement étant composé de spectacle, il lui sera possible d'en ajouter dans son arrayList de spectacles par la fonction *ajouterSpectacle()*. Il en va de même pour les spectacles et les numéros avec la méthode *ajouterNumero()*. La classe Numéro possède elle aussi de quoi ajouter des animaux, des accessoires et des acteurs au besoin. Enfin chacune des classes Événement, Spectacle et Numéro possède une méthode afin d'ajouter un nouvel organisateur.

Détail des scénarios

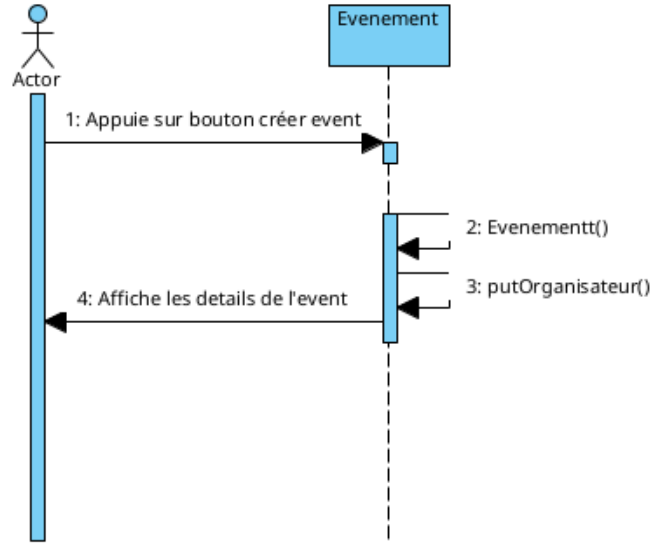

Figure 5. Diagramme de séquences détaillées du scénario nominal de UC1, l'acteur pourra appuyer sur un bouton afin de créer et devenir l'organisateur d'un nouvel événement.

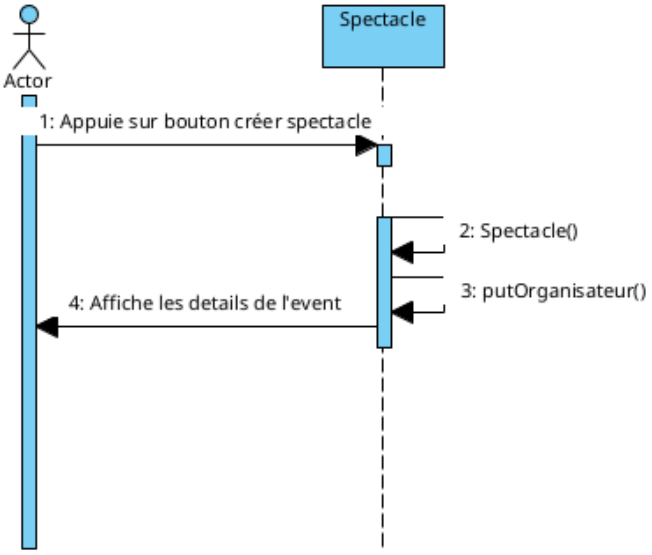

Figure 6. Diagramme de séquences détaillées du scénario nominal de UC1.1, l'acteur pourra appuyer sur un bouton afin de créer et devenir l'organisateur d'un nouveau spectacle.

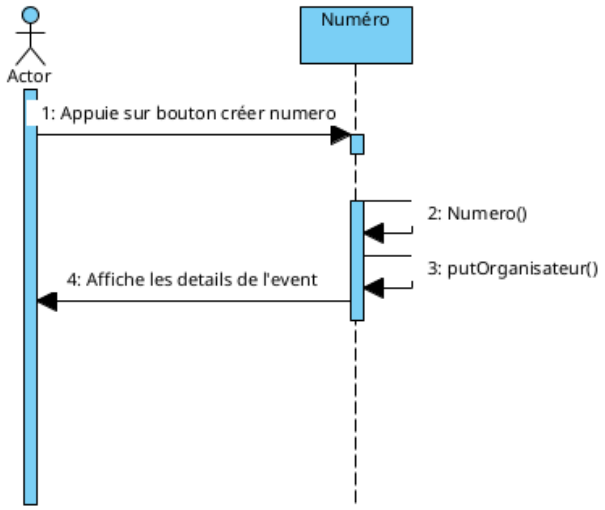

Figure 7. Diagramme de séquences détaillées du scénario nominal de UC1.2, l'acteur pourra appuyer sur un bouton afin de créer et devenir l'organisateur d'un nouveau numéro.

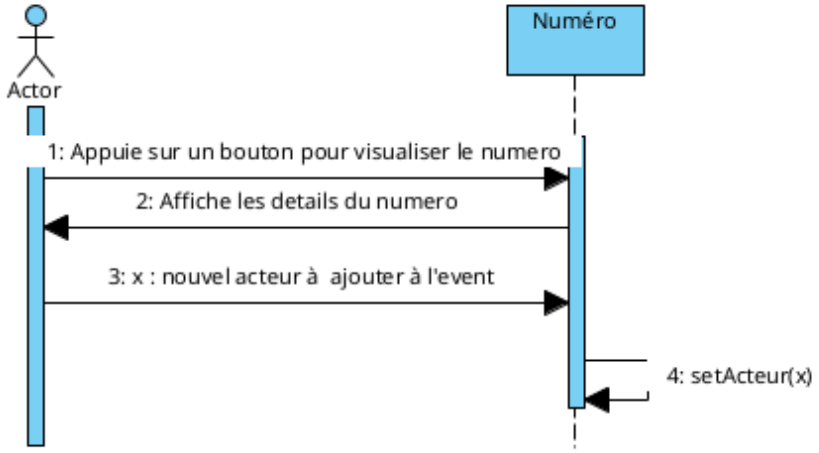

Figure 8. Diagramme de séquences détaillées du scénario nominal de UC1.3, l'acteur pourra appuyer sur un bouton afin d'ajouter un acteur à un numéro.

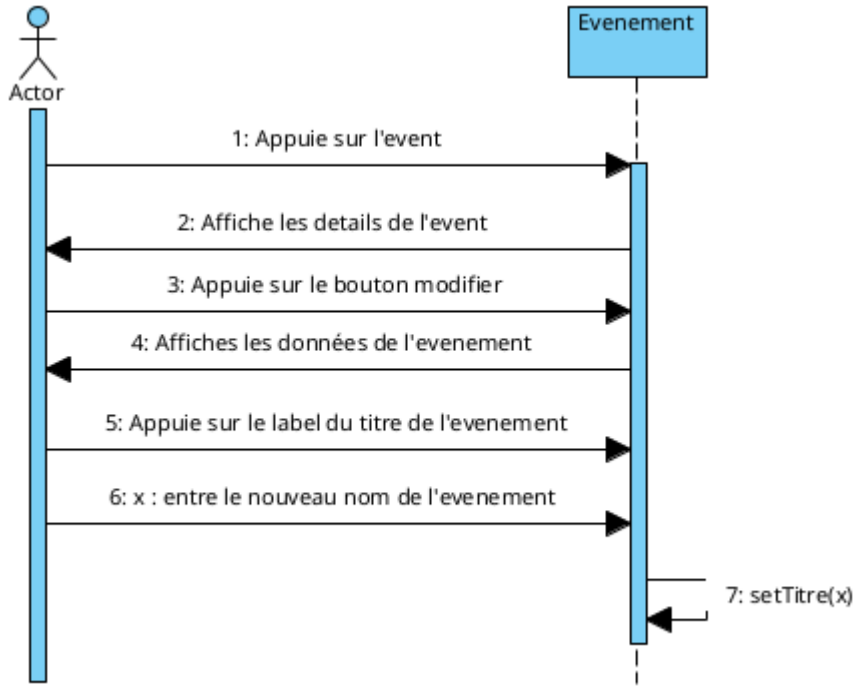

Figure 9. Diagramme de séquences détaillées du scénario nominal de UC2, l'acteur pourra appuyer sur un bouton afin de modifier les détails d'un événement déjà enregistré.

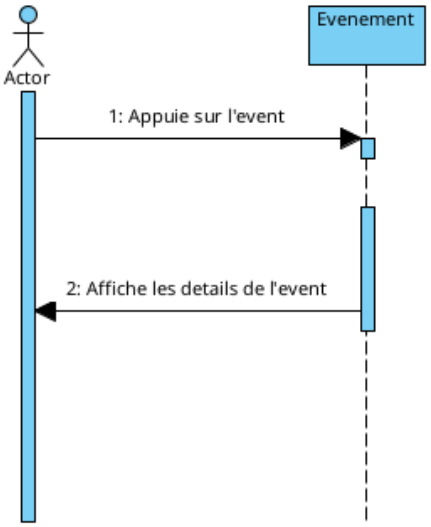

Figure 10. Diagramme de séquences détaillées du scénario nominal de UC3, l'acteur pourra appuyer sur un bouton afin de visualiser les détails d'un événement déjà enregistré.

Conclusion

Notre projet consiste en la conception d'une application de création d'évènements à destination des cirques ambulants dans le but de leur permettre de planifier leurs tournées rapidement et de minimiser les imprévus.

Elle permet également de créer un planning des tours et numéros ainsi que les acteurs relatifs à chacuns accompagnés de leurs accessoires permettant au propriétaire du cirque

d'organiser précisément, clairement et de rendre disponible en tout temps et lieux ces informations cruciales.

Pour ce faire nous allons réaliser une application en Java et JavaFX qui ne stockera pas de données. Notre application aura plusieurs scènes pour permettre l'affichage de différentes fenêtres permettant à terme de faire différents affichages en fonction du profil de l'utilisateur de l'application notamment pour les spectateurs des cirques et les créateurs d'événements.

Durant la phase de création du dossier de conception, nous avons utilisé le site draw.io pour faire les diagrammes de classe, et nous avons eu quelques problèmes de synchronisation ce qui à causé un certain nombre de perte de données. Nous avons aussi eu des problèmes de licences avec Visual Paradigm ce qui a entraîné des ralentissement durant la phase de conception.

Pour nous permettre de communiquer efficacement et de pouvoir partager des fichiers nous avons créé un serveur discord, nous nous sommes aussi servi de Google docs pour un travail unifié. Enfin, pour pouvoir coder de manières collaborative nous nous servirons soit de Github soit du Gitlab mis à notre disposition.