

## THESIS / THÈSE

### MASTER EN INGÉNIEUR DE GESTION À FINALITÉ SPÉCIALISÉE EN ANALYTICS & DIGITAL BUSINESS

Identification et développement d'un cadre méthodologique pour la business analysis  
application à un cas d'étude

Bellenger, Joris

*Award date:*  
2021

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



# Identification et développement d'un cadre méthodologique pour la business analysis : application à un cas d'étude

**Joris BELLENGER**

**Directeur: Prof. S. FAULKNER**

Mémoire présenté  
en vue de l'obtention du titre de  
Master 120 en ingénieur de gestion, à finalité spécialisée  
en Analytics & Digital Business

**ANNEE ACADEMIQUE 2020-2021**

# Abstract

This thesis concerns the identification and development of a methodological framework for business analysis. The importance of the role played by the business analyst in IT projects is no longer in question. This work presents the different stages of work and analysis of the business analyst, namely: the context study, the analysis of the solution and the design of the solution. For each stage, several analysis and modelling tools are proposed. All of this is put into practice through a case study.

Ce mémoire traite de l'identification et développement d'un cadre méthodologique pour la business analysis. L'importance du rôle joué par le business analyst dans les projets informatiques n'est plus à prouver. Cet ouvrage présente les différentes étapes de travail et d'analyse du business analyste, à savoir : l'étude de contexte, l'analyse de la solution et la conception de la solution. Pour chaque étape, il est proposé plusieurs outils d'analyse et de modélisation. Le tout est mis en application à travers une étude de cas.

# Avant-propos

Je souhaite avant tout remercier les différentes personnes qui m'ont aidé et soutenu dans la réalisation de ce mémoire.

D'abord, je remercie mon directeur de mémoire, le professeur Stéphane Faulkner, pour ses conseils et son suivi. Je remercie également mes parents, mon frère et Florence pour leur soutien inconditionnel. Enfin, je tiens à remercier Gautier Claudel pour l'aide apportée dans la correction et relecture de ce mémoire.

# Table des matières

|   |           |
|---|-----------|
| <b>Introduction</b>   | <b>6</b>  |
| <b>1 Généralités</b>  | <b>8</b>  |
| 1.1 La business analysis . . . . .  | 8         |
| 1.2 Les concepts fondamentaux de la business analysis . . . . .   | 9         |
| 1.3 Le corpus de compétences du business analyst . . . . .  | 10        |
| 1.3.1 L'ingénierie des exigences . . . . .  | 11        |
| 1.4 Le processus de l'analyse business . . . . .  | 13        |
| 1.5 Les cycles de vie des systèmes d'information . . . . .  | 14        |
| 1.6 Les différents niveaux d'exigences dans le cadre de la business analysis<br>[Beatty et Wiegers, 2013] . . . . . | 16        |
| 1.7 Les approches en business analysis . . . . .  | 18        |
| <b>2 Introduction à l'étude de cas</b>  | <b>19</b> |
| 2.1 Mise en contexte . . . . .  | 19        |
| 2.2 Méthodologie . . . . .  | 20        |
| 2.2.1 Processus . . . . .   | 20        |
| 2.2.2 Approches . . . . .   | 21        |
| 2.2.3 Les outils . . . . .  | 21        |
| <b>3 Étude de contexte</b>  | <b>22</b> |
| 3.1 Analyse du contexte métier . . . . .  | 23        |
| 3.2 Identification et analyse des parties prenantes . . . . .   | 26        |
| 3.3 Système actuel . . . . .  | 27        |
| 3.3.1 Conclusion . . . . .  | 33        |
| 3.4 Analyse du système futur . . . . .  | 35        |
| 3.5 Analyse des livrables . . . . .   | 36        |
| 3.6 Modélisation de l'environnement actuel et futur avec i* . . . . .   | 37        |
| 3.6.1 Concepts de base . . . . .  | 37        |
| 3.6.2 Processus d'utilisation . . . . .   | 41        |
| 3.6.3 Application au cas d'étude . . . . .  | 41        |

|          |  |           |
|----------|--|-----------|
| <b>4</b> | <b>Analyse de la solution</b>  | <b>46</b> |
| 4.1      | Les exigences logicielles . . . . .  | 46        |
| 4.1.1    | Taxonomie des exigences logicielles . . . . .                                | 47        |
| 4.2      | Affinement des parties prenantes . . . . .                                   | 48        |
| 4.3      | Analyse du domaine . . . . .   | 50        |
| 4.3.1    | Le diagramme entité-association . . . . .                                    | 50        |
| 4.3.2    | Concepts de base . . . . .   | 50        |
| 4.3.3    | Application au cas d'étude . . . . .   | 51        |
| 4.4      | Analyse des exigences de données . . . . .                                   | 52        |
| 4.4.1    | Diagramme relationnel . . . . .  | 52        |
| 4.4.2    | Data dictionary . . . . .  | 54        |
| 4.5      | Analyse des exigences fonctionnelles . . . . .                               | 56        |
| 4.5.1    | UML . . . . .  | 56        |
| 4.5.2    | Processus . . . . .  | 57        |
| 4.5.3    | Élaboration du digramme de cas d'utilisation . . . . .                       | 58        |
| 4.5.4    | Priorisation des cas d'utilisation . . . . .                                 | 61        |
| 4.5.5    | Élaboration des scénarios des cas d'utilisation . . . . .                    | 62        |
| 4.5.6    | Élaboration du diagramme de classes fonctionnel . . . . .                    | 64        |
| 4.5.7    | Élaboration des diagrammes de séquences . . . . .                            | 66        |
| 4.5.8    | Documentation des règles métiers . . . . .                                   | 68        |
| 4.6      | Analyse des exigences non fonctionnelles . . . . .                           | 69        |
| 4.6.1    | Les exigences non fonctionnelles externes . . . . .                          | 69        |
| 4.6.2    | Les exigences non fonctionnelles internes . . . . .                          | 69        |
| 4.6.3    | Processus . . . . .  | 70        |
| 4.6.4    | Outil . . . . .  | 70        |
| 4.6.5    | Application au cas d'étude . . . . .   | 70        |
| 4.7      | Analyse des exigences relatives aux interfaces externes du système . . . . . | 71        |
| 4.7.1    | Application au cas d'étude . . . . .   | 71        |
| <b>5</b> | <b>Conception de la solution</b>   | <b>72</b> |
| 5.1      | Processus . . . . .  | 72        |
| 5.1.1    | Élaboration du diagramme de classes de conception . . . . .                  | 73        |
| 5.1.2    | Élaboration des diagrammes de séquences . . . . .                            | 75        |
|          | <b>Conclusion</b>  | <b>76</b> |

# Table des figures

|      |   |    |
|------|---|----|
| 1.1  | Rôle de la business analysis . . . . .  | 8  |
| 1.2  | Business Analysis Core Concept Model. Adapté de « <i>Babok : A Guide to the Business Analysis Body of Knowledge</i> » par International Institute of Business Analysis, 2015, p. 14. Adapté avec permission. . . . .                                      | 10 |
| 1.3  | Processus de développement des exigences . . . . .  | 12 |
| 1.4  | Les activités de l'ingénierie des exigences . . . . .   | 13 |
| 1.5  | Processus de la business analysis. Adapté de « <i>Introduction to Business Analysis</i> » par Milani F., 2019, p. 9. Adapté avec permission. . . . .  | 14 |
| 1.6  | Modèle en cascade . . . . .   | 16 |
| 1.7  | Modèle incrémental . . . . .  | 16 |
| 3.1  | Plan de la salle de théâtre du Medallion. Reproduit de « Systems Analysis, Design and Development Case Study : Medallion Theater - Ticket Sales System » par Fox T., 2017, <i>Journal of the International Academy for Case Studies</i> , p. 24 . . . . . | 23 |
| 3.2  | Business Model Canvas du théâtre Medallion . . . . .  | 25 |
| 3.3  | Analyse des 5 Whys du théâtre Medallion . . . . .   | 28 |
| 3.4  | Les différentes composantes d'un processus métier. Reproduit de « Introduction to business process management » par M. Dumas et al., <i>Fundamentals of business process management</i> , 2013, p. 6 . . . . .  | 29 |
| 3.5  | Notations de base BPMN 2.0 [Signavio, 2012] . . . . .   | 30 |
| 3.6  | Modélisation du processus de réservation d'un ticket pour une performance au théâtre Medallion. . . . .   | 31 |
| 3.7  | Représentation du diagramme de Fishbone . . . . .   | 34 |
| 3.8  | Présentation des différentes causes de l'insatisfaction client dans la réservation de tickets au théâtre Medallion . . . . .  | 34 |
| 3.9  | Les objets de relations dans $i^*$ . . . . .  | 38 |
| 3.10 | Les relations de dépendance dans $i^*$ . . . . .  | 38 |
| 3.11 | Les types de liens moyens-fins dans $i^*$ . . . . .   | 39 |
| 3.12 | Les types de liens de décomposition dans $i^*$ . . . . .  | 40 |
| 3.13 | Modèle de dépendance stratégique sans le système de réservation informatisé   | 42 |

|      |  |    |
|------|--|----|
| 3.14 | Modèle de dépendance stratégique avec le système de réservation informatisé        | 43 |
| 3.15 | Modèle de relation stratégique sans le système de réservation informatisé          | 44 |
| 3.16 | Modèle de relation stratégique avec le système de réservation informatisé          | 45 |
| 4.1  | Taxonomie des exigences logicielles  | 48 |
| 4.2  | Diagramme entité-association générique   | 51 |
| 4.3  | Diagramme entité-association du théâtre Medallion                                  | 52 |
| 4.4  | Diagramme relationnel générique  | 53 |
| 4.5  | Diagramme relationnel du théâtre Medallion   | 54 |
| 4.6  | Diagramme des cas d'utilisation générique  | 58 |
| 4.7  | Relations des cas d'utilisation  | 59 |
| 4.8  | Diagramme des cas d'utilisation du théâtre Medallion                               | 60 |
| 4.9  | Les éléments généraux du diagramme d'activités                                     | 64 |
| 4.10 | Les classes du digramme de classes   | 65 |
| 4.11 | Visibilité des attributs et méthodes   | 65 |
| 4.12 | Diagramme de classes fonctionnel du système de réservation du théâtre<br>Medallion | 66 |
| 4.13 | Diagramme de séquences générique   | 67 |
| 5.1  | Relation d'héritage dans un diagramme de classes                                   | 73 |
| 5.2  | Interface dans un diagramme de classes   | 73 |
| 5.3  | Diagramme de classes de conception du cas d'étude                                  | 74 |



# Liste des tableaux

|     |  |    |
|-----|--|----|
| 3.1 | Identifiants des sièges par catégorie. Reproduit de « Systems Analysis, Design and Development Case Study : Medallion Theater - Ticket Sales System » par Fox T., 2017, <i>Journal of the International Academy for Case Studies</i> , p. 22 . . . . . | 24 |
| 3.2 | Identification des rôles des parties prenantes pour le projet du théâtre Medallion . . . . .   | 27 |
| 3.3 | Identification des solutions pour le théâtre Medallion . . . . .   | 35 |
| 3.4 | Identification des écarts pour le théâtre Medallion . . . . .  | 36 |
| 4.1 | RACI matrice du projet Medallion . . . . .   | 49 |
| 4.2 | Data dictionary pour les classes . . . . .   | 55 |
| 4.3 | Data dictionary pour les associations . . . . .  | 56 |
| 4.4 | Priorisation des cas d'utilisation pour le théâtre Medallion . . . . .   | 62 |
| 4.5 | Règles métier du théâtre Medallion . . . . .   | 68 |
| 4.6 | Une partie des mots clés existant en Planguage . . . . .   | 70 |
| 4.7 | Spécification des performances requises en termes de temps de réponse pour la génération des rapports . . . . .  | 71 |

# Introduction

Les systèmes d'information sont devenus incontournables dans les entreprises. Les projets de développement de ces systèmes restent toujours un défi. En effet, beaucoup de projets ne respectent pas parfaitement les différentes contraintes et exigences fixées par les organisations. Malheureusement, une proportion encore importante de ces projets sont challengés, voire pire, échoués. Dans les raisons qui expliquent ces échecs, on remarque un manque de travail et d'analyse des exigences revenant comme l'une des causes principales. Dans ce contexte, il est alors intéressant d'étudier les fondements et sous-jacents de la business analysis, son rôle et son travail.

L'objet de ce mémoire porte donc sur l'identification et le développement d'un cadre méthodologique pour la business analysis. Ce cadre méthodologique sera également mis en pratique au travers d'un cas d'étude afin de rendre plus concret l'utilisation des différents outils.

À l'heure actuelle, il n'existe aucune méthodologie reconnue comme un standard dans l'analyse business. Cette raison s'explique principalement étant donné les caractéristiques des projets informatiques variant effectivement énormément. Pour chaque partie de ce mémoire, il sera proposé de suivre une méthodologie comprenant un processus bien défini ainsi que différents outils et techniques qui aideront à réaliser les différentes activités composant le travail du business analyst.

Ce mémoire comporte cinq chapitres. Le premier présente un ensemble de généralités concernant l'analyse business. Les chapitres suivants se consacrent au processus de la business analysis. Chaque chapitre traite d'une étape de processus d'analyse business.

Le deuxième chapitre concerne la mise en contexte de l'étude de cas. Ce chapitre présente également le processus mais également l'approche qui seront suivis tout au long de ce travail. Ce chapitre traitera en outre de l'importance et de l'intérêt de l'utilisation des outils de modélisation car bon nombre d'entre eux seront présentés et appliqués à l'étude de cas.

Le troisième chapitre se concentre sur la phase de l'étude de contexte. L'importance de cette phase et les différentes étapes du processus de l'analyse de contexte y sont expliquées. Les outils présentés et utilisés au cours de cette phase sont : le Business Model Canvas (BMC), les 5 why's, BPMN, le diagramme Fishbone et i\*.

Le quatrième chapitre consiste en l'analyse de la solution selon différents types d'exigences. Ce chapitre accorde une place importante à l'analyse des exigences fonctionnelles qui suivra le processus unifié (UP) proposé par les auteurs d'UML. Il va alors de soi qu'une partie des diagrammes d'UML seront présentés et utilisés. Plus précisément, le diagramme des cas d'utilisation, le diagramme d'activités, le diagramme de classes et le diagramme de séquences.

Le cinquième et dernier chapitre présentera la phase de conception. Ce chapitre commence avec une discussion autour de la pertinence et la qualification du business analyst dans la réalisation de la phase de conception d'un système. Cette phase ne sera que brièvement entamée avec la présentation et l'élaboration d'un diagramme de classes de conceptions et des diagrammes de séquences.

# Chapitre 1

## Généralités

### 1.1 La business analysis

Dans les années 1980, l'utilisation des systèmes d'information dans les organisations s'étend. La mise en place de tels projets s'avérait très souvent être un échec. En effet, beaucoup d'entre eux ne délivraient pas la valeur attendue ou encore ne respectaient pas les limites budgétaires et temporelles. Il est alors apparu évident qu'une communication efficace entre le domaine business et celui de l'IT jouait un rôle crucial dans la bonne réalisation des projets informatiques. Dès lors, il est devenu important de pouvoir articuler et communiquer les exigences business dans une forme plus adéquate et plus compréhensible pour le domaine de la technologie de l'information. Nait alors le rôle de l'analyste business, ou analyste métier [Milani, 2019]. Comme le montre la figure 1.1, la business analysis permet de faire le lien entre les deux domaines.

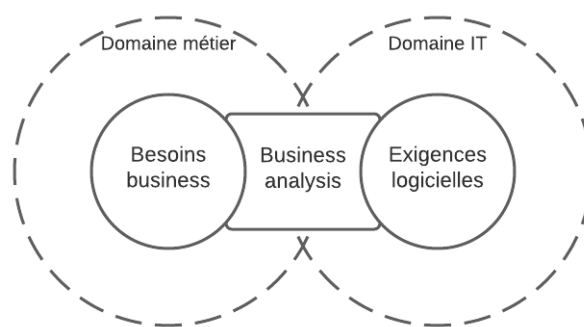


FIGURE 1.1 – Rôle de la business analysis

Le BABOK (2015) définit l'analyse business comme étant « la pratique qui permet le changement dans une entreprise en définissant les besoins et recommandant des solutions qui apportent de la valeur aux parties prenantes. » En d'autres termes, l'analyse business permet d'établir les besoins d'une organisation et la nécessité du changement mais également de concevoir des solutions pour mettre en place ce changement.

Cette définition permet d'évoquer le fait que, dorénavant, l'analyse business ne se limite plus simplement à la compréhension et à la communication des exigences pour le domaine IT. En effet, elle va au-delà et ce, en étendant la pratique sur l'analyse et la communication des raisons de ces changements (business drivers), des effets désirés (business goals), des composantes à mettre en place ou à modifier à dessein d'aboutir aux objectifs (business deliverables) et de la manière dont ces derniers sont atteints (business requirements) et ce dont il est question de changer ou d'adapter dans le business (business rules) [Milani, 2019].

On remarque également que la définition donnée par le BABOK se veut assez générique. En effet, le BABOK (2015, page 2) spécifie que cette pratique peut s'effectuer selon différentes perspectives et donc, ne concerne pas systématiquement la technologie de l'information (IT). Fredrik Milani (2019) l'explique par le fait que l'analyse business, ne se réduisant plus à la simple formulation d'exigences, et considérant également le travail d'analyse effectué avant ces exigences s'est, au fil du temps, étendue à une analyse de résolutions de causes et besoins d'une organisation et à l'évaluation de solutions de manière générale.

## 1.2 Les concepts fondamentaux de la business analysis

On peut caractériser l'analyse business par six concepts fondamentaux :

- Les changements : réalisation de transformations en réponse à un besoin ;
- Les besoins : un problème ou une opportunité à traiter ;
- Les solutions : une façon particulière de répondre à un ou plusieurs besoins dans un contexte déterminé ;
- Les parties prenantes : un individu ou groupe d'individus concernés par le changement, le besoin ou la solution ;
- La valeur : se caractérise par son importance, son utilité pour les parties prenantes ;
- Le contexte : la situation et ses facteurs ayant un impact ou sont impactés et qui permettent de comprendre l'intérêt du changement [IIBA, 2015].

Ces six concepts ont été représentés sous forme d'un modèle appelé Business Analysis Core Concept Model (BACCM) disponible en figure 1.2 [IIBA, 2015].

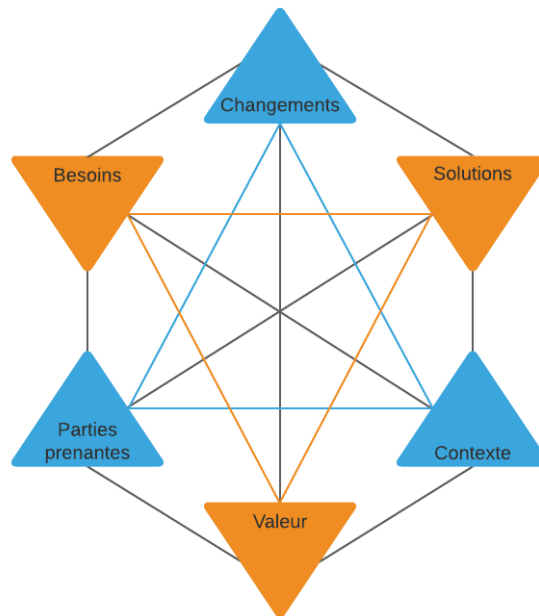


FIGURE 1.2 – Business Analysis Core Concept Model. Adapté de « *Babok : A Guide to the Business Analysis Body of Knowledge* » par International Institute of Business Analysis, 2015, p. 14. Adapté avec permission.

### 1.3 Le corpus de compétences du business analyst

Le BABOK (2015) identifie également une série de compétences requises pour la business analysis. Ce corpus se compose de six domaines :

- La planification et le suivi : concernent toute l'organisation et la coordination du travail à effectuer par le business analyst ;
- L'élicitation et la collaboration : concernent toutes les activités liées à la planification, l'élaboration et la réalisation des phases d'élicitation ;
- Gestion du cycle de vie des exigences : la gestion des exigences reprend tout le travail d'organisation et de coordination lié à la phase de développement des exigences et à la gestion du cycle de vie des exigences. Cette activité fait partie de la pratique de l'ingénierie des exigences ;
- L'analyse et la spécification des exigences : concernent tout le travail de définition et d'organisation des exigences collectées de manière à concevoir la solution proposée. Cette activité est également partie intégrante de l'ingénierie des exigences ;
- L'évaluation des solutions : couvre les activités d'évaluation et d'amélioration de la valeur délivrée par une solution [IIBA, 2015].

### 1.3.1 L'ingénierie des exigences

Une partie importante du corpus de connaissances du business analyst fait appel à une discipline à part entière appelée l'ingénierie des exigences. L'ingénierie des exigences « *is a structured approach to the assessment of the role that a future information system (IS) is to have within a relatively well-delimited human and/or automated environment.* » [Jureta et al., 2008, p. 87] (est une approche structurée de l'évaluation du rôle qu'un futur système d'information (SI) doit jouer dans un environnement humain et/ou automatisé relativement bien délimité [notre traduction]).

Pour davantage de clareté, Karl Wiegiers et Joy Beatty (2013) suggèrent de séparer l'ingénierie des exigences en deux activités : le développement des exigences et la gestion des exigences. En fonction du cycle de vie de développement du système, ces activités se dérouleront à différents moments.

#### Développement des exigences

Le développement des exigences suit un processus itératif (modélisé en figure 1.3) en quatre étapes :

##### 1. L'élicitation

Elle couvre toutes les activités ayant pour objectif de découvrir et d'identifier les exigences. C'est une des tâches les plus importantes et les plus longues du processus. Les étapes clés sont : l'identification et l'analyse des parties prenantes, la compréhension et l'analyse de l'environnement dans lequel la solution est demandée et la compréhension des objectifs business, des tâches et buts des utilisateurs. Le mot élicitation est davantage préféré au terme « collecter » pour insister sur le fait que cette activité ne se réduit pas simplement à poser la bonne question afin de collecter les informations dont l'analyste aurait besoin [Nuseibeh et Easterbrook, 2000, Pandey et al., 2010].

##### 2. L'analyse

Cette partie consiste à redéfinir les exigences élicitées sous une forme plus technique et de les présenter sous différents angles de manière à avoir une compréhension plus riche et plus précise de ces dernières. Les tâches clés sont : l'analyse de l'information élicitée de manière à distinguer les différents types d'exigences, reformuler les exigences de haut niveau vers un niveau plus adapté, dériver les

informations en exigences logicielles, prioriser les exigences et identifier toutes lacunes ou anomalies dans les exigences [Beatty et Wiegers, 2013, Pandey et al., 2010].

### 3. La spécification

Cette activité consiste à présenter et recueillir les exigences dans une forme la plus formelle possible dans l'intention que ces dernières puissent être communiquées aisément aux différentes parties prenantes. La tâche clé de cette activité est de présenter les exigences dans une forme adéquate au niveau de technicité des parties prenantes afin de pouvoir les communiquer avec les autres acteurs du projet en minimisant au maximum l'ambiguïté et l'incompréhension [Beatty et Wiegers, 2013, Pandey et al., 2010].

### 4. La validation

L'étape de la validation permet d'obtenir confirmation auprès de personnes responsables que les exigences, telles que spécifiées dans l'activité précédente, sont correctes, complètes et conformes aux attentes. La validation comprend deux activités clés : réaliser une révision des exigences spécifiées afin de les rectifier si nécessaire pour ensuite les valider et développer des tests dans le but de s'assurer que les exigences respectent bien les attentes et objectifs de l'organisation [Beatty et Wiegers, 2013].

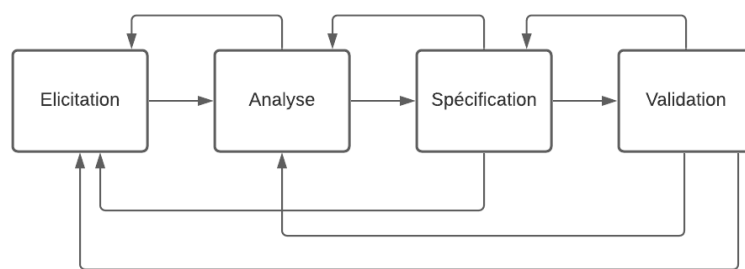


FIGURE 1.3 – Processus de développement des exigences

## La gestion des exigences

La gestion des exigences reprend tout le travail d'organisation et de coordination lié à la phase de développement des exigences et à la gestion du cycle de vie des exigences. Les tâches clés sont notamment : la planification, l'exécution, le suivi et le contrôle [IIBA, 2015]. La planification et l'exécution sont relatives à la description des personnes, des



responsabilités, des outils, des approches qui seront utilisés lors du développement des exigences. Le suivi et contrôle sont relatifs à la description et mise en place d'outils et de critères afin d'évaluer les exigences mises en place. La figure 1.4 résume sous forme d'un diagramme les activités principales de l'ingénierie des exigences [Pandey et al., 2010].

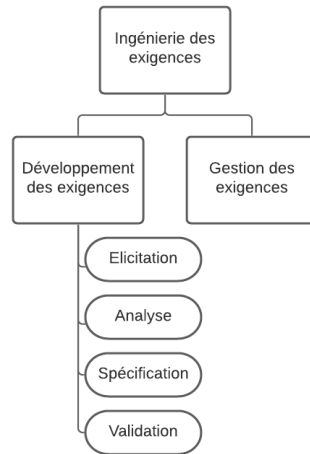


FIGURE 1.4 – Les activités de l'ingénierie des exigences

## 1.4 Le processus de l'analyse business

Il n'existe pas de processus standard pour l'analyse business tant les projets sont variés et différents. Par conséquent, il existe une variété de méthodes, d'outils et de techniques pouvant être appliquées pour cette activité [Milani, 2019].

Fredrik Milani (2019) propose un processus regroupant les activités fondamentales de l'analyse business modélisé en figure 1.5. La première activité traite de tout le travail de préparation et d'organisation du projet à venir. Ensuite, une analyse du système actuel (as-is) est effectuée afin de comprendre en profondeur les problèmes ou opportunités et besoins de l'organisation. Pour donner suite à cette analyse, l'analyste est en mesure de définir le système futur, désiré (to-be) et de proposer une solution et des alternatives qui permettraient de répondre aux besoins de l'organisation. Les solutions sont ensuite évaluées et la solution la plus adaptée est retenue. S'en suit alors une phase de développement et de conception détaillée de la solution pour ensuite pouvoir l'implémenter et la déployer. À terme, cette solution est évaluée afin de définir la valeur délivrée par la solution.

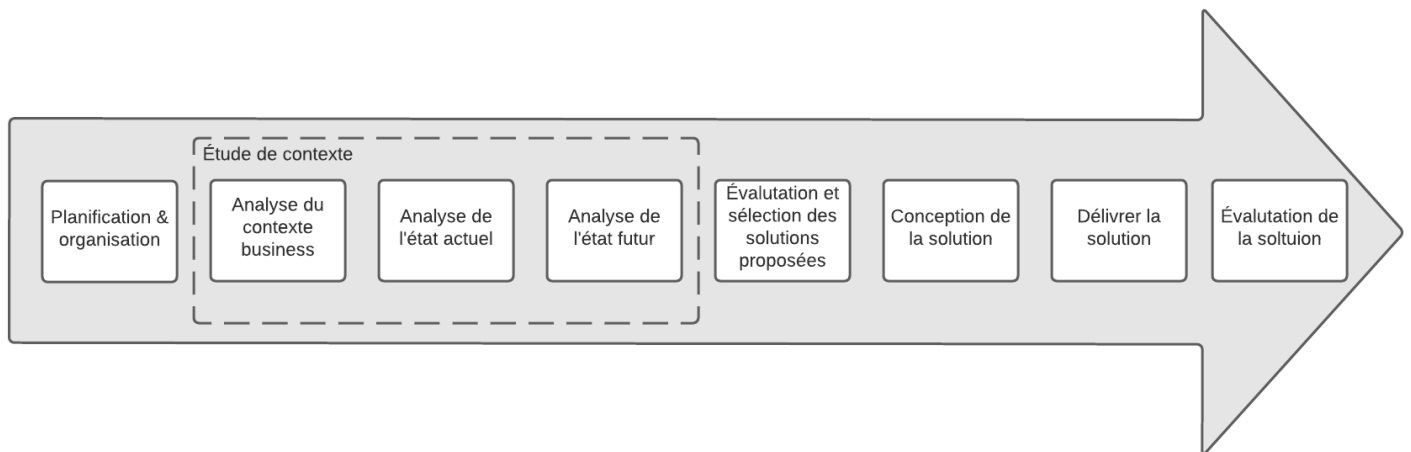


FIGURE 1.5 – Processus de la business analysis. Adapté de « *Introduction to Business Analysis* » par Milani F., 2019, p. 9. Adapté avec permission.

## 1.5 Les cycles de vie des systèmes d’information

Étant donné que la nature de l’analyse business est liée au développement des systèmes d’information, il est intéressant de discuter de leurs cycles de vie afin de pouvoir y situer le rôle du business analyst.

Le cycle de vie est le processus par lequel un système d’information est développé, testé, installé et maintenu [Satzinger et al., 2015]. On retrouve trois grandes phases dans le cycle de vie : le développement, l’utilisation et la maintenance [Faulkner, 2021]<sup>1</sup>. La phase de développement est celle qui nous intéresse le plus. Elle se compose de six étapes :

### 1. Démarrage

Le démarrage est une pré-étude que l’on peut diviser en deux grandes phases. La première phase concerne l’analyse du contexte. Il est essentiel de comprendre un certain nombre d’aspects de haut niveau sur le système comme son environnement dans lequel le système d’information va être développé. Au démarrage, il est aussi important d’identifier les différentes parties prenantes au projet et d’identifier leurs pouvoirs de décision. La deuxième phase concerne une étude de faisabilité sur différents axes qui permet d’évaluer s’il est faisable ou non de développer ce système [Satzinger et al., 2015, Faulkner, 2021<sup>1</sup>].

1. Source dérivée de WebCampus (non accessible au public)

## 2. Exigences

Cette étape comporte le cœur du travail du business analyst. L'objectif de cette étape est donc de comprendre, au niveau de la couche métier, les besoins et exigences de l'organisation et des utilisateurs du futur système informatique et, ensuite, de transcrire ces besoins sous forme de fonctionnalités techniques, appelées exigences logicielles, afin que le système réponde aux attentes de l'entreprise [TP026B, 2017].

## 3. Design

Cette étape apporte les spécifications de l'architecture technique du système. On entre dans une phase de conception du système qui présente et communique la manière dont le système devra être développé [TP026B, 2017].

## 4. Construction

Cette étape est relative à l'implémentation du système. Cette implémentation est réalisée conformément aux exigences spécifiées dans les étapes précédentes par des développeurs [TP026B, 2017].

## 5. Vérification

Le logiciel, une fois développé, est testé. Cette étape permet de s'assurer que le système en ce compris ses fonctions et son comportement conviennent bien aux attentes et exigences de l'entreprise. De plus, au niveau technique, on évalue la robustesse du système face à divers scénarios et à la bonne intégration des composants du système [TP026B, 2017].

## 6. Déploiement

Cette étape consiste à mettre en production la solution développée dans l'environnement de l'organisation [TP026B, 2017].

Ces six étapes peuvent prendre place à des moments différents en fonction du type de cycle de vie choisi pour développer un projet. Tous les cycles de vie varient entre deux types génériques. Le premier est de type séquentiel. Le projet est alors réalisé étape par étape, de manière séquentielle. Après les étapes d'analyse de conception, le logiciel est développé en une fois et délivré au client dans son entièreté. L'un des modèles parmi ceux

des plus connus suivant un cycle séquentiel est le modèle en cascade (ou waterfall model, en anglais) [Ruparelia, 2010]. Il est représenté en figure 1.6. Le second est de type itératif. À l’opposé du type séquentiel, plus traditionnel, le système est développé par constructions ou incréments consécutifs. Le système n’est alors pas développé en une fois mais par lots de fonctionnalités. Un exemple de modèle suivant cette approche est le modèle incrémental présenté en figure 1.7. Ce modèle offre l’avantage de livrer rapidement une partie des fonctionnalités du logiciel, délivrant ainsi plus rapidement une valeur business au client [Isaias et Issa, 2015].

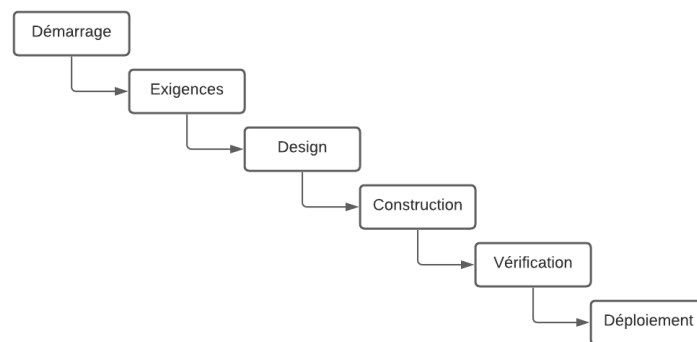


FIGURE 1.6 – Modèle en cascade

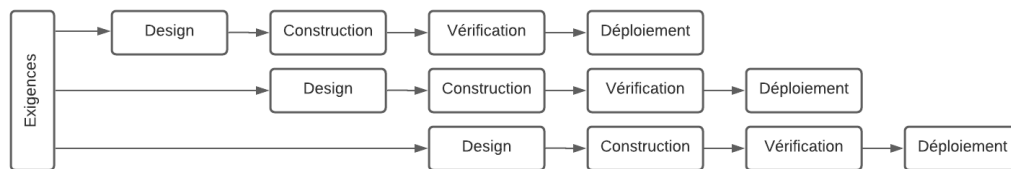


FIGURE 1.7 – Modèle incrémental

## 1.6 Les différents niveaux d'exigences dans le cadre de la business analysis [Beatty et Wiegers, 2013]

Tout au long de son travail, le business analyst sera amené à traiter différents types d'exigences ayant différents niveaux techniques. Au fur et à mesure qu'il avancera dans le processus, ces exigences adopteront un niveau de plus en plus bas.

Il est possible de distinguer trois couches d'exigences, allant d'un haut niveau vers un bas niveau.

## 1. Les exigences métiers

La première couche regroupe les exigences métiers (ou exigences business). Ces exigences seront examinées lors de l'analyse de contexte. Elles décrivent les objectifs de l'organisation permettant de comprendre les raisons du changement souhaité ou en d'autres mots, la mise en place de la solution.

## 2. Les exigences utilisateurs

Les exigences utilisateurs permettent de décrire les tâches et objectifs que le produit (qui représente cette solution) doit pourvoir réaliser de manière à satisfaire les besoins utilisateurs. Ces exigences comprennent également l'ensemble des caractéristiques et propriétés que le produit doit comprendre afin de correspondre aux attentes des utilisateurs et ce, dans son entièreté.

## 3. Les exigences logicielles

Les exigences logicielles comprennent l'ensemble des fonctionnalités, fonctions, comportements, et caractéristiques que le produit doit comprendre. Elles sont différentes des exigences utilisateurs dans leurs spécifications et technicités. Ces exigences sont dérivées des exigences utilisateurs de manière à indiquer aux développeurs ce qu'ils doivent développer.

L'objectif derrière le travail du business analyst sera toujours de partir de la couche d'exigences de haut niveau et de progressivement affiner ces exigences afin d'accéder aux couches suivantes.

Les types d'exigences présentées plus en haut peuvent être influencées par des règles métiers. Ces règles définissent un ensemble de propriétés propres au domaine métier (standards, règles gouvernementales, politique d'entreprise, etc.). Ces règles ne sont donc pas des exigences logicielles en elles-mêmes car elles ont une existence en dehors du logiciel. Toutefois, étant donné leurs propriétés, ces règles métiers seront à l'origine d'exigences logicielles [Beatty et Wiegers, 2013].

## 1.7 Les approches en business analysis

Indéniablement, le choix du cycle de vie d'un logiciel, qui variera entre un type séquentiel ou itératif, aura un impact sur l'approche de travail du business analyst. Dès lors, au même titre que les deux types de cycles de vie générique, l'approche du business analyst variera entre une approche prédictive ou adaptative [Satzinger et al., 2015].

L'approche prédictive requiert de définir l'entièreté de la solution avant son implémentation. Cette approche est privilégiée lorsque l'ensemble des exigences peuvent être élicitées avant la mise en œuvre de la solution. L'approche adaptative préconise de délivrer rapidement de la valeur business en procédant par des courtes itérations. Dès lors, la phase d'analyse ne tient plus en un tout mais est constituée de courtes itérations où une partie des exigences est prise en charge. Il est alors important de les prioriser. Par conséquent, si le cycle de vie suit un type incrémental, l'approche adaptative sera la plus adéquate. Inversement, pour un projet se déroulant de manière classique, une approche prédictive sera alors privilégiée [Satzinger et al., 2015, Leau et al., 2012].

Au même titre que les cycles de vie des projets informatiques, les approches ont également leurs parts d'avantages et d'inconvénients. L'approche prédictive tente de réduire l'incertitude en définissant l'entièreté de la solution avant son implémentation. De cette manière, le risque est minimisé et le contrôle maximisé. Toutefois, il persiste toujours le risque que la totalité des exigences n'ait pu être identifiée au début ou qu'elles aient été mal identifiées. Ceci mènerait donc à une mauvaise implémentation et donc, une solution qui ne remplit pas les exigences. L'approche adaptative, quant à elle, offre l'avantage de délivrer très rapidement de la valeur business. Il faudra cependant accepter un plus grand niveau d'incertitude quant au regard de la solution finale qui sera implémentée [IIBA, 2015, Leau et al., 2012].

# Chapitre 2

## Introduction à l'étude de cas

### 2.1 Mise en contexte

Le théâtre Medallion propose des spectacles divers et variés. Pour les clients, il est possible de réserver une place pour une représentation et de la payer, ensuite, le soir de l'événement. Ce fonctionnement a pour conséquence une grande insatisfaction client. Cette dernière s'explique d'une part, par le fait qu'il arrive de manière récurrente qu'une place, pourtant réservée, ait été vendue à un autre client par erreur. D'autre part, la mauvaise gestion des réservations laisse souvent accidentellement des sièges vides, pensant qu'un siège était réservé alors qu'il n'en était rien. Cela représente dès lors un manque à gagner conséquent pour l'entreprise.

Face à ce problème, le théâtre souhaiterait mettre en place un logiciel de réservation de tickets. Ce dernier est convaincu que l'implémentation d'une telle solution lui serait bénéfique.

Le lecteur pourra trouver en appendice 1 l'énoncé de l'étude cas dans sa totalité. Cet énoncé sera considéré comme la base pour l'élicitation des différentes exigences tout au long de ces cas d'étude. L'étude de cas en question est issue du « Journal of the International Academy for Case Studies » [Fox, 2017].

## 2.2 Méthodologie

### 2.2.1 Processus

Afin de réaliser l'étude de cas, nous partirons du processus générique de Milani (2019) présenté en figure 1.5. Toutefois, nous nous limiterons aux activités comprises dans l'étude de contexte ainsi qu'à l'activité de conception de la solution.

Ce travail commencera donc par une étude de contexte. Cette étude de contexte permettra à l'analyste de capter les exigences métiers et d'avoir une idée sur les exigences utilisateurs à venir. Cette partie comprendra comme étapes :

- Analyse du contexte métier ;
- Identification des parties prenantes ;
- Analyse du système actuel (as-is) ;
- Analyse du système futur (to-be) ;
- Analyse des écarts ;
- Étude de faisabilité.

Ensuite, la deuxième autre partie consistera en l'analyse de la solution. L'objectif de cette partie sera alors de transcrire les exigences utilisateurs en exigences logicielles de manière à communiquer aux développeurs le système qu'ils devront développer (« what »). Pour réaliser cette partie nous suivrons le « Unified Processus ». Les principes de cette méthode seront détaillés plus après. Cette partie se composera de :

- Analyse du domaine ;
- Analyse des exigences relatives aux données ;
- Analyse des exigences fonctionnelles ;
- Analyse des exigences non fonctionnelles ;
- Analyse des exigences relatives aux interfaces externes du système.

La méthode se poursuivra ensuite dans la troisième partie qui traitera des prémices de la phase de conception du système. Cette partie donnera un aperçu sur la manière dont le système devra être développé (« how »). Cette partie comportera deux étapes :

- Développement du diagramme de classes de conception ;
- Développement des digrammes de séquences.



### 2.2.2 Approches

Pour réaliser ce cas d'étude, nous opterons pour une approche prédictive consistant donc à développer l'ensemble des exigences logicielles avant le développement de celui-ci [Satzinger et al., 2015]. D'un point de vue des exigences, comme expliqué plus avant dans le point 1.6, nous partirons d'exigences de haut niveau pour ensuite les affiner de manière à obtenir des exigences de bas niveau. Par conséquent, nous pouvons assimiler cette approche à celle d'une approche top-down.

### 2.2.3 Les outils

Pour la réalisation de ce cas d'étude, un grand nombre d'outils de modélisation seront utilisés. Leurs concepts de base seront également présentés. L'utilisation de la modélisation dans cette pratique est fréquente car elle présente de nombreux avantages. En effet, l'utilisation de certains outils de modélisation permet de rendre les spécifications des exigences plus formelles que si elles étaient spécifiées dans un langage naturel. Cette formalisation rend la communication des exigences plus facile et plus fiable. D'autant plus que les modèles offrent la possibilité d'exposer de nombreuses exigences de manière lisible, au contraire des longues listes d'exigences présentées sous la forme textuelle. Leurs aspects très visuels aident également dans l'analyse des exigences rendant la détection d'erreurs et d'omissions plus facile. La modélisation sert donc à la fois de support pour éliciter, communiquer et analyser les exigences mais également à les documenter [Beatty et Wiegers, 2013]. L'utilisation de la modélisation permet aussi de délimiter un champ d'application des besoins à analyser et ainsi d'éviter de s'étendre dans un travail non pertinent [Cernosek et Naiburg, 2004].

# Chapitre 3

## Étude de contexte

Une étude de contexte débouche sur la création d'un document de vision ou encore business case. IBM donne une définition précise du document de vision :

« Un document de vision définit la portée et l'objectif généraux d'un programme, d'un produit ou d'un projet. Une énonciation claire du problème, de la solution proposée et des fonctions de haut niveau du produit permettent de définir les attentes et de réduire les risques. Cette rubrique fournit un aperçu du contenu potentiel d'un document de vision. » [IBM, 2020]

L'étude de contexte trouve sa pertinence dans le fait que la réalisation d'un projet informatique demande à comprendre d'emblée la raison d'être de ce projet. Il est donc impératif d'analyser le problème à résoudre, pourquoi ce problème doit-il être résolu et quelles en sont les personnes impliquées [Borgida et al., 2009]. Cette investigation constituera les données d'entrée afin de déterminer des objectifs conformes et en phase avec les besoins de l'organisation.

Dans un deuxième temps, il s'agit de traiter les exigences métiers et également d'avoir une idée générale des besoins des potentiels utilisateurs. D'un point de vue des exigences, on appelle cette phase la phase précoce des exigences (early-phase requirements) [Yu, 1997].

Dans la littérature, la spécification et l'analyse des exigences en phase précoce (early phase) sont souvent ignorées alors que cette étape est pourtant essentielle dans la bonne

compréhension des objectifs souhaités du système. Dans les approches traditionnelles de développement de systèmes d'information, la phase d'analyse des exigences logicielles commence avec une description informelle de ce que le système doit faire [Van Lamsweerde, 2001]. John Mylopoulos, Lawrence Chung, Stephen Liao and Huaqing Wang et Eric Yu (2001) soutiennent l'idée que l'analyse des exigences doit aller au-delà de la simple compréhension et spécification des fonctions, données et interfaces d'un système. Les auteurs proposent donc l'ingénierie des exigences orientée objectifs comme une approche permettant d'identifier et d'améliorer les objectifs organisationnels et techniques.

### 3.1 Analyse du contexte métier

Chaque organisation évolue dans un contexte business particulier. Il est important pour le business analyst de prendre connaissance et de se familiariser avec ce contexte. Connaître ce contexte business permettra à l'analyste de se diriger vers une solution plus adaptée à cette dernière [Milani, 2019]. À la suite de cette analyse, il nous sera aussi possible d'identifier les différents acteurs évoluant dans ce contexte.

D'après les informations détaillées dans le cas d'étude [Fox, 2017], le théâtre Medallion dispose d'une unique salle de théâtre de 602 places. Il existe quatre catégories de places différentes : les sièges « Orchestra », « Mezzanine », « Balcony » et « Box ». Ces catégories de sièges sont respectivement vendues à un prix unitaire de 65€, 55€, 40€ et 85€.

Chaque siège bénéficie d'un identifiant unique permettant au client d'identifier le siège acheté. La figure 3.1 présente un plan de la salle du théâtre Medallion.

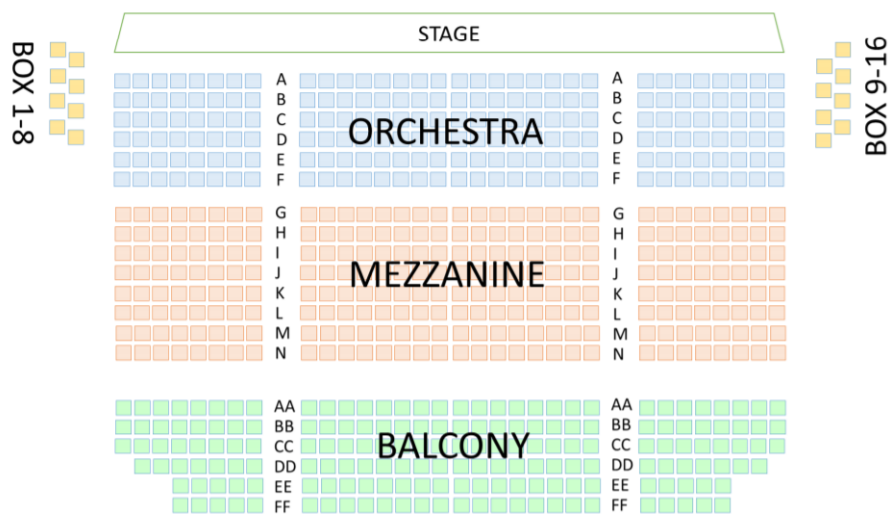


FIGURE 3.1 – Plan de la salle de théâtre du Medallion. Reproduit de « Systems Analysis, Design and Development Case Study : Medallion Theater - Ticket Sales System » par Fox T., 2017, *Journal of the International Academy for Case Studies*, p. 24

Chaque rangée est assignée d'une ou deux lettres. Pour chaque rangée, chaque siège porte un numéro. Le premier siège en partant de la gauche a le numéro 1, le suivant le numéro 2 et ainsi de suite jusqu'au bout de la rangée. La table 3.1 répertorie les différents identifiants des sièges pour chaque catégorie.

| Catégories | Prix | Identifiants des sièges                                |
|------------|------|--|
| Orchestra  | 65€  | A1-A30, B1-B30, C1-C30, D1-D30, E1-E30, F1-F30         |
| Mezzanine  | 55€  | G1-30, H1-30, I1-30, J1-30, K1-30, L1-30, M1-30, N1-30 |
| Balcony    | 40€  | AA1-30, BB1-30, CC1-30, DD1-28, EE1-24, F1-24          |
| Box        | 85€  | X1-16  |

TABLE 3.1 – Identifiants des sièges par catégorie. Reproduit de « Systems Analysis, Design and Development Case Study : Medallion Theater - Ticket Sales System » par Fox T., 2017, *Journal of the International Academy for Case Studies*, p. 22

Le théâtre est ouvert tous les jours de la semaine et propose chaque jour deux performances : une en début d'après-midi, à 14h00 et une en soirée, à 20h00. Ces performances sont de différents types : concerts ou pièces de théâtre. C'est le seul et unique service que propose le Medallion. Dans le cadre de ce travail, nous admettons le fait que le théâtre Medallion est financé par les pouvoirs publics. Ce dernier ayant été créé afin de proposer un espace culturel et éducatif aux citoyens.

Afin de terminer l'analyse du core business, nous proposons l'illustration du business model du client sous forme d'un Business Model Canvas (BMC) que le lecteur pourra trouver en figure 3.2. Il a l'avantage d'offrir un clair aperçu du business model d'une société et de pouvoir rapidement le visualiser et l'évaluer [Osterwalder et Pigneur, 2010]. En effet, les neuf composantes du BMC ne sont pas simplement disposées les unes à la suite des autres, puisqu'elles sont présentées de manière à mettre en évidence certaines relations et problématiques [Hong et Fauvel, 2013].

Les composantes du BMC sont :

- Segments clients : identifie les différents types de clients que l'entreprise touche ;
- Proposition de valeur : définit ce qu'apporte le service ou produit d'un point de vue besoin du client ;
- Distribution : identifie les différents canaux par lesquels cette valeur est délivrée au client ;
- Relation client : identifie les moyens par lesquels l'entreprise alimente et entretient sa relation avec sa clientèle ;

- Sources de revenus : identifie les différentes sources de revenus résultant de la proposition de valeur offerte au client ;
- Ressources clés : identifie les différents facteurs nécessaires pour offrir de manière réussie le service/produit ;
- Activités clés : identifie les différentes activités clés de l'entreprise ;
- Partenaires clés : identifie les différents partenaires avec qui l'entreprise travaille et opère ;
- Structure de coûts : identifie les différents coûts liés à l'activité de l'entreprise [Osterwalder et Pigneur, 2010].

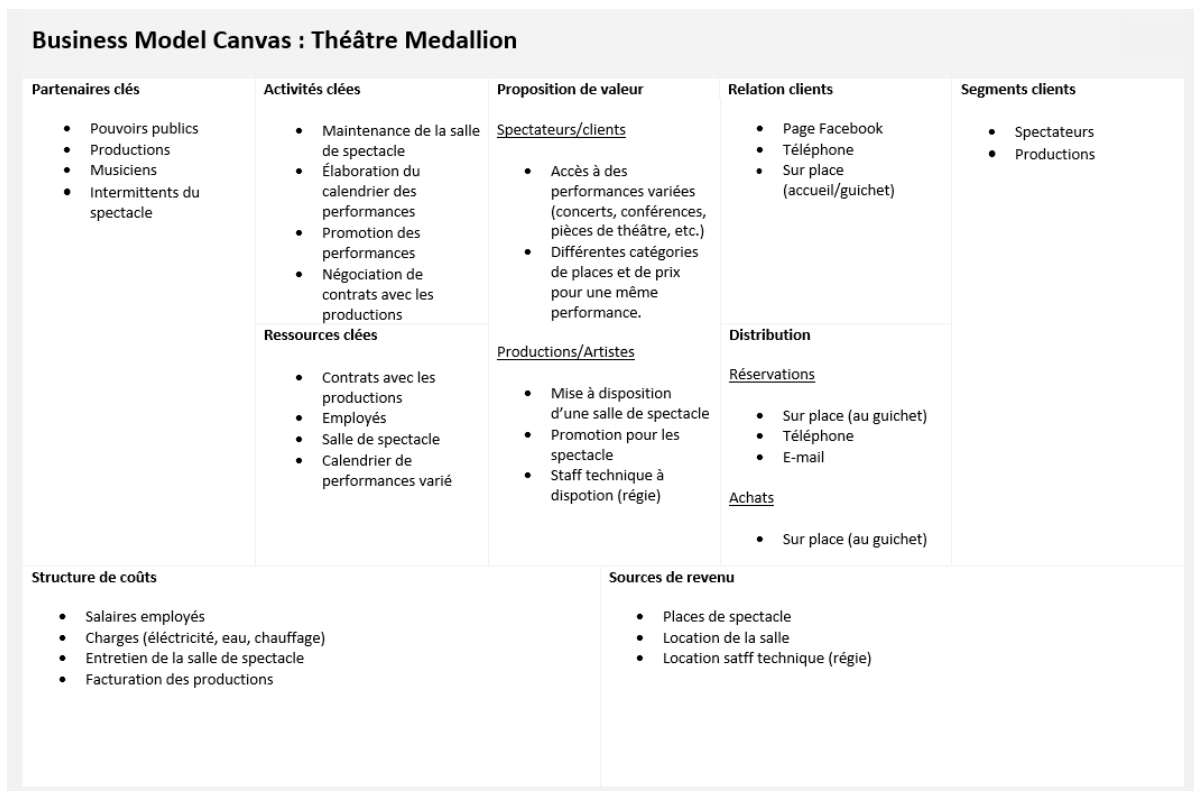


FIGURE 3.2 – Business Model Canvas du théâtre Medallion

## 3.2 Identification et analyse des parties prenantes

Les parties prenantes, ou stakeholders en anglais, sont « *any group or individual who can affect or is affected by the achievement of the organization's objectives.* » [Freeman, 2010, p.52] (tout groupe ou individu qui peut affecter ou est affecté par la réalisation des objectifs de l'organisation [notre traduction]). Ce sont avec ces individus que l'analyste métier va collaborer de manière directe ou indirecte.

Il est donc important d'identifier les parties prenantes avant de commencer l'analyse des besoins. Il s'agit ici d'identifier les personnes impactées par le domaine analysé. L'analyse des stakeholders se fait de manière itérative. En effet, au fur et à mesure de ce travail, il sera possible d'identifier d'autres personnes et d'analyser les parties prenantes plus en profondeur [Institute, 2015].

Il est possible de catégoriser les parties prenantes. Elles peuvent être soit internes, soit externes à l'organisation. D'un autre point de vue, il est aussi intéressant de distinguer les parties prenantes ayant un rôle opérationnel et celles ayant un rôle exécutif. Cette seconde distinction est d'une importance capitale pour le business analyst. En effet, les stakeholders ayant un rôle exécutif sont toutes les personnes ayant une fonction décisionnelle au sein de l'entreprise. Ils prennent des décisions dans l'intérêt du business de l'entreprise. Les parties prenantes de type opérationnel représentent toutes les personnes qui œuvrent sur le terrain. Ceci implique que c'est aux personnes ayant un pouvoir décisionnel que revient l'approbation finale des différentes fonctions et exigences de la solution implémentée [Satzinger et al., 2015]. Les parties prenantes opérationnelles, elles, seront également consultées car elles sont plus à même de fournir les différentes exigences propres au processus métier.

À ce stade, plusieurs types de stakeholders peuvent être identifiés. Il y a d'abord celui étant à l'initiative et est donc le responsable du projet. Ce projet a évidemment un coût qui devra être pris en charge. Celui ayant pour rôle de financer le projet est plus communément appelé le sponsor. Lors du développement du projet, certaines personnes auront également la charge de réguler et d'imposer des contraintes qui devront être respectées. La solution, une fois implémentée, sera utilisée par une ou des personnes, appelées utilisateurs. Il est aussi intéressant d'identifier le type de personnes à qui bénéficiera la mise en place de cette solution. Il est à noter qu'il est tout à fait possible qu'une même personne ait plusieurs des rôles identifiés ci-dessus [Institute, 2015].

La table 3.2 identifie les différentes personnes qui vont endosser les rôles détaillés précédemment pour le cas d'étude du théâtre Medallion. Les noms et prénoms des personnes sont purement fictifs. On se rend alors compte que seule la direction, ainsi que les pouvoirs

publics, sont des parties prenantes exécutives. Le reste est composé de stakeholders opérationnels. La solution bénéficiera principalement aux clients (spectateurs). Et de manière secondaire, au personnel du Medallion qui l'utilisera.

| Rôles          | Fonctions                   | Identités                    |
|----------------|-----------------------------|------------------------------|
| Responsable    | Direction                   | Alain Dubois                 |
| Sponsor        | Pouvoirs publics (culture)  | Julie Père                   |
| Utilisateurs 1 | Guichet                     | Nicole Steun, Caroline Steun |
| Utilisateurs 2 | Programmation               | Jan Peters                   |
| Régulateurs    | Direction, Pouvoirs publics | Alain Dubois, Julie Père     |
| Bénéficiaires  | Clients                     | /                            |

TABLE 3.2 – Identification des rôles des parties prenantes pour le projet du théâtre Medallion

### 3.3 Système actuel

Les recommandations et besoins d'une organisation émanent d'une situation particulière. Analyser la situation, c'est donc devoir en comprendre les problèmes, les différentes causes ou facteurs provoquant cette situation. Nombreuses sont les organisations n'étudiant pas de manière assez approfondie les problèmes car souvent déjà trop focalisées sur la solution. Par conséquent, beaucoup de projets, en finalité, ne se retrouvent pas alignés avec les besoins de l'entreprise. On passe alors à côté de beaucoup d'opportunités, sans compter toutes les ressources gaspillées [Spradlin, 2012].

En identifiant et analysant les problèmes, le business analyst sera en mesure de pouvoir concevoir une solution résolvant le problème et étant en concordance avec les besoins et exigences du client [Rooney and Heuvel, 2004]. Pour entamer cette analyse, il est nécessaire de passer par une phase d'élicitation afin de recueillir des informations pertinentes.

Tout d'abord, dans l'énoncé du cas d'étude, se trouve la retranscription d'une conversation entre un client et un membre du guichet du Medallion qui sera analysée. Ensuite, nous tâcherons d'identifier et d'analyser le processus de réservation d'une place pour une performance. Ce processus sera modélisé via BPMN.

Les informations pertinentes pouvant être retenues de la discussion entre le client et le responsable du guichet sont présentées dans la figure 3.3 sous forme d'une analyse « 5 Whys ». Cette analyse est fréquemment utilisée dans le domaine du Lean Management

et notamment dans le Toyota Production System. Cet outil permet d'identifier le problème source à une situation d'échec. On interroge alors cinq fois la raison de l'existence d'une telle situation (pourquoi?). De cette manière, l'analyse évolue en commençant par l'identification d'une problématique de haut niveau pour se diriger au fur et à mesure vers l'identification d'une problématique de plus bas niveau et ainsi de suite, jusqu'à identifier le réel problème auquel fait face l'organisation [Murugaiah et al., 2010, Institute, 2015].

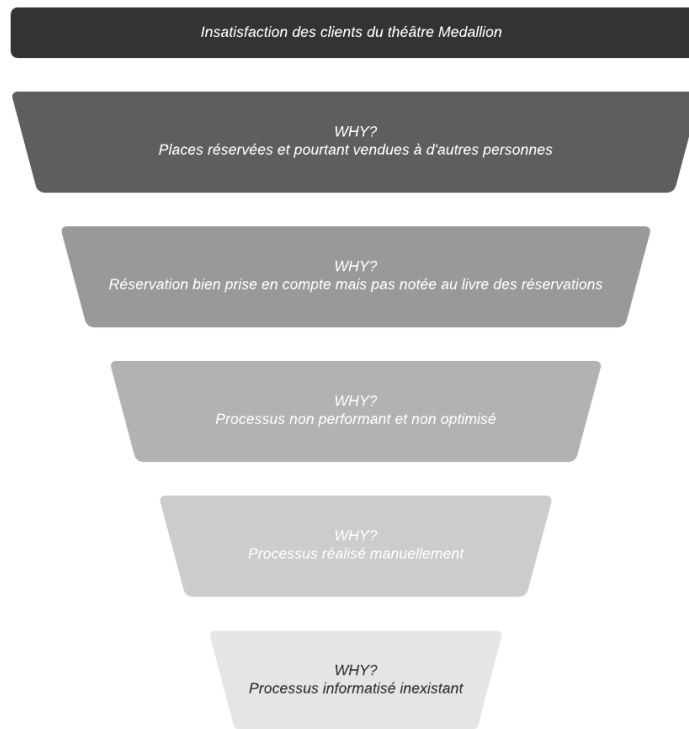


FIGURE 3.3 – Analyse des 5 Whys du théâtre Medallion

Étant donné que le processus de réservation des tickets au Medallion est une cause de l'insatisfaction des clients, il est intéressant d'analyser plus en profondeur ce processus métier.

Un processus métier (business process en anglais), est défini comme « *a collection of inter-related events, activities and decision points that involve a number of actors and objects, and that collectively lead to an outcome that is of value to at least one customer.* » [Dumas et al., 2013, p. 5] (un ensemble d'événements, d'activités et de points de décision interdépendants impliquant un certain nombre d'acteurs et d'objets, et qui conduisent collectivement à un résultat ayant de la valeur pour au moins un client [notre traduction]).

Un processus se compose alors d'événements et d'activités. Un événement est quelque chose qui arrive automatiquement et qui n'a pas de durée dans le temps. Par exemple, dans



le cas du théâtre Madallion, l'arrivée d'un client qui souhaite réserver une performance est considérée comme un événement. Une activité, quant à elle, représente une unité de travail. Elle a donc une durée dans le temps. Par exemple, inscrire le nom d'un client sur un carton de réservation. Lors du processus, il est possible qu'un acteur doive prendre une décision. Cette décision est choisie à la suite de l'évaluation d'une situation à un moment donné. La décision aura pour effet d'influencer le processus et son résultat. Le processus métier comporte également différents acteurs comme des personnes ou des logiciels, des objets physiques (par exemple, un document) et immatériels comme un e-mail [Dumas et al., 2013].

Le processus a pour objectif de fournir un résultat. Ce résultat peut être positif ou négatif [Dumas et al., 2013]. Bien entendu, dans notre contexte, il va de soi qu'il est souhaité que ce résultat soit positif. En effet, un résultat positif délivre de la valeur pour différents acteurs impliqués dans le processus. En figure 3.4, les différentes composantes d'un business process sont représentées sous forme d'un diagramme.

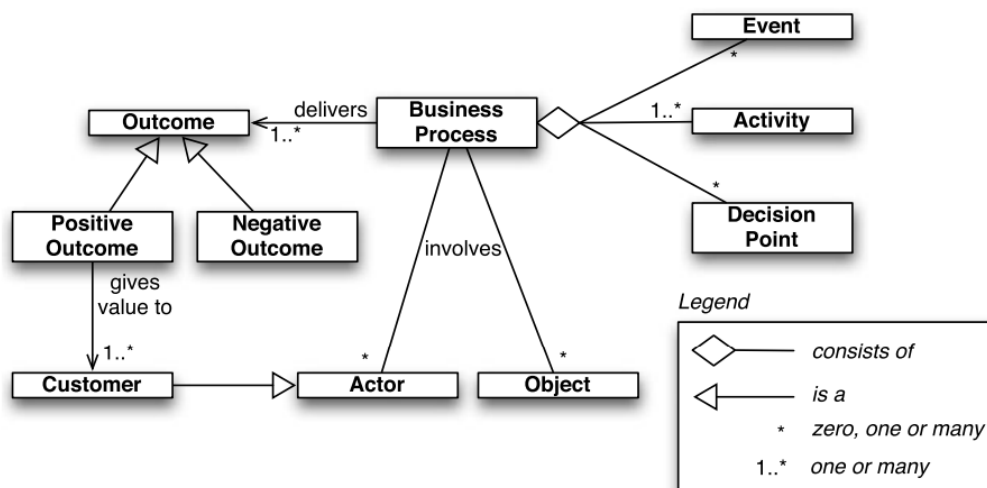


FIGURE 3.4 – Les différentes composantes d'un processus métier. Reproduit de « Introduction to business process management » par M. Dumas et al., *Fundamentals of business process management*, 2013, p. 6

Pour analyser le processus, nous présentons en figure 3.6 une modélisation de ce dernier via BPMN. Il nous permettra d'identifier les problèmes et lacunes présents lors du déroulement de la réservation d'une place pour un spectacle. BPMN (Business Process Model and Notation) est une notation graphique permettant de standardiser un processus sous forme d'un diagramme. Le choix de BPMN pour la modélisation des processus métiers se justifie en considérant qu'il est un des grands standards. De plus, sa notation est simple et facile à comprendre. BPMN détient également la norme ISO/IEC 19510 [OMG, 2021].

Une partie des notations de base, couramment utilisées, sont présentées dans la figure 3.5.

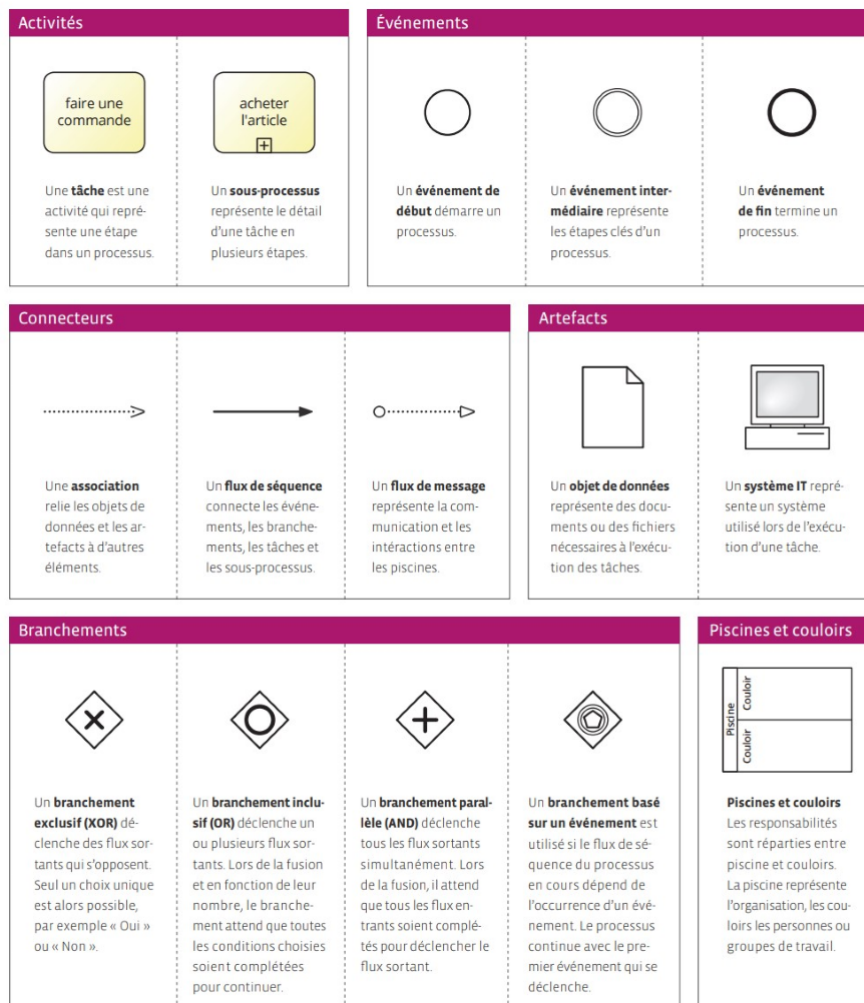


FIGURE 3.5 – Notations de base BPMN 2.0 [Signavio, 2012]

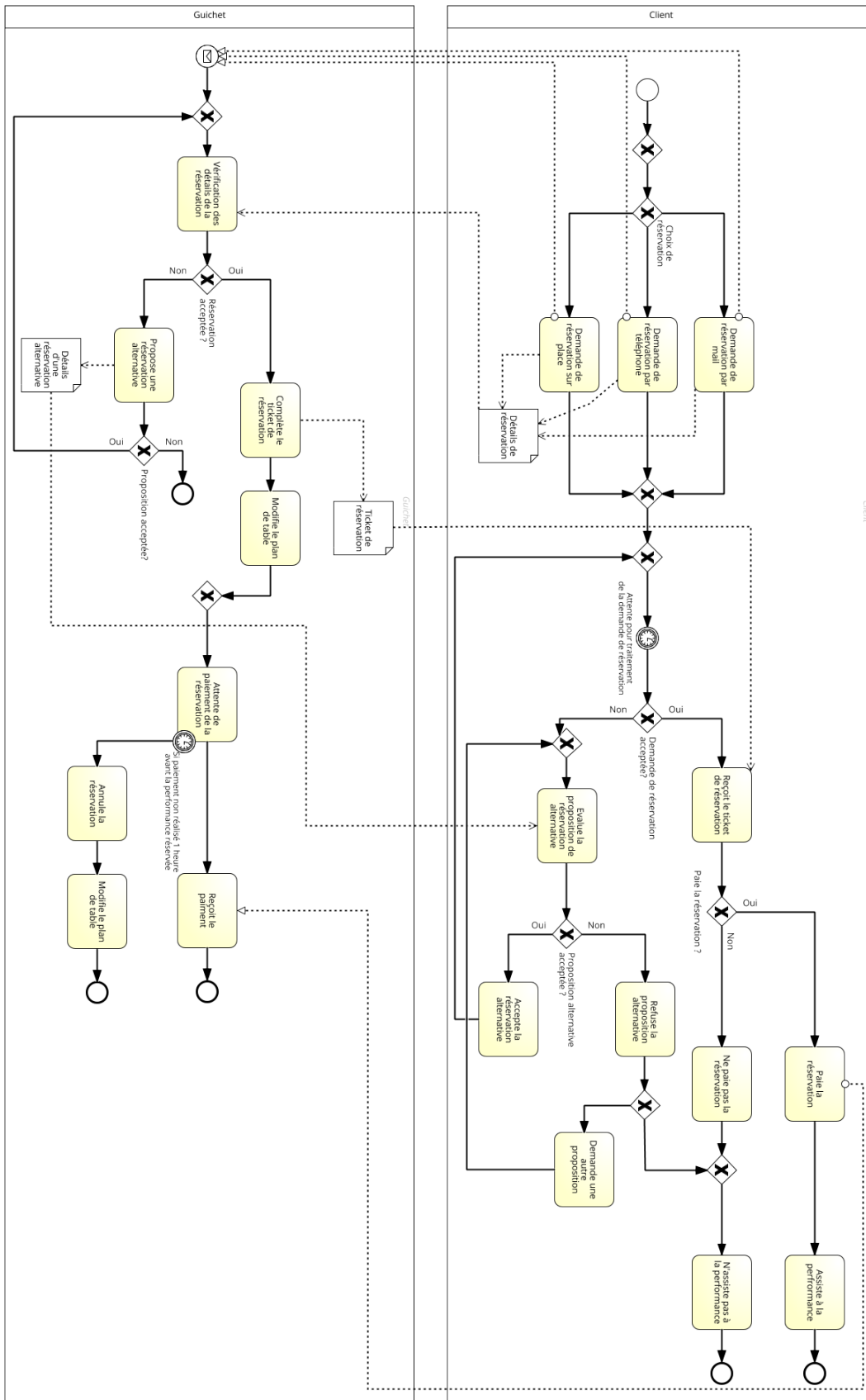


FIGURE 3.6 – Modélisation du processus de réservation d’un ticket pour une performance au théâtre Medallion.

En analysant le processus, nous pouvons facilement dresser une liste des problèmes susceptibles de provoquer une insatisfaction de la clientèle lors de la réservation :

- La vérification des demandes de réservation introduites par les clients se fait de manière manuelle. Cela a pour conséquences :
  1. La possibilité d'erreurs humaines lors de la vérification ;
  2. Un temps de vérification particulièrement long.
- Deux personnes du personnel sont chargées de prendre en charge les demandes de réservation via les différents canaux (courriel, téléphone, sur place). Il est donc possible que deux demandes soient traitées de manière simultanée. Si la coordination n'est pas optimale entre les deux employés, il peut y avoir un risque qu'une même place soit réservée au nom de deux personnes pour une même performance à une même date et à une même heure.
- Si la demande de réservation (réalisée par e-mail) ne peut être acceptée, l'employé va automatiquement proposer une réservation alternative. Entre la proposition formulée par l'employé, l'évaluation de la proposition par le client et l'acceptation de cette proposition, se déroule un laps de temps non négligeable. Par conséquent, une fois l'accord donné par le client, l'employé doit une nouvelle fois procéder à une vérification.
- La réservation, une fois acceptée, n'implique pas un paiement immédiat. La réservation est considérée comme annulée si le montant de la réservation n'a pas été réglé au maximum une heure avant l'heure de début de la performance dont la réservation fait l'objet. Il est donc possible de se retrouver dans un scénario où certains sièges se retrouvent vides au moment de la performance.
- Les canaux disponibles pour réaliser une demande de réservation sont nombreux. C'est une charge de travail non négligeable pour les employés.
- Le processus, en majorité manuel, le rend particulièrement long jusqu'à arriver au résultat.

### 3.3.1 Conclusion

De manière à conclure cette analyse et à dresser un bilan des problèmes identifiés, nous proposons une synthèse sous forme d'un diagramme de Fishbone ou diagramme d'Ishikawa. Ce diagramme permet de représenter de manière plus visuelle les différentes causes relatives à un effet. Il présente une ligne horizontale menant jusqu'à l'effet ou la situation problématique. Des lignes verticales sont disposées autour de la ligne horizontale et représentent les différentes causes ou facteurs responsables de cet effet [Barsalou, 2014].

De manière générale, les facteurs sont répartis en cinq catégories, appelés « les 5 M », à savoir :

- La main d'œuvre : elle peut être une raison de l'effet. Par exemple, parfois dû à un manque de qualification ;
- Le matériel : la présence ou l'inexistence de tout matériel peut également être une cause de cet effet ;
- Le milieu : l'environnement dans lequel se déroule le processus peut avoir un impact ;
- La mesure : toute mesure défectueuse lors de la réalisation d'un processus peut également être une cause ;
- La méthode : tout ce qui est relatif à la mauvaise manière de réaliser un processus [Barsalou, 2014].

Ensuite, en dessous de ces grandes catégories sont présentées les sous-catégories permettant de détailler davantage les facteurs responsables de la situation ou effet analysé. Le lecteur trouvera en figure 3.7 une représentation générique du diagramme de Fishbone tel qu'expliqué plus en haut.

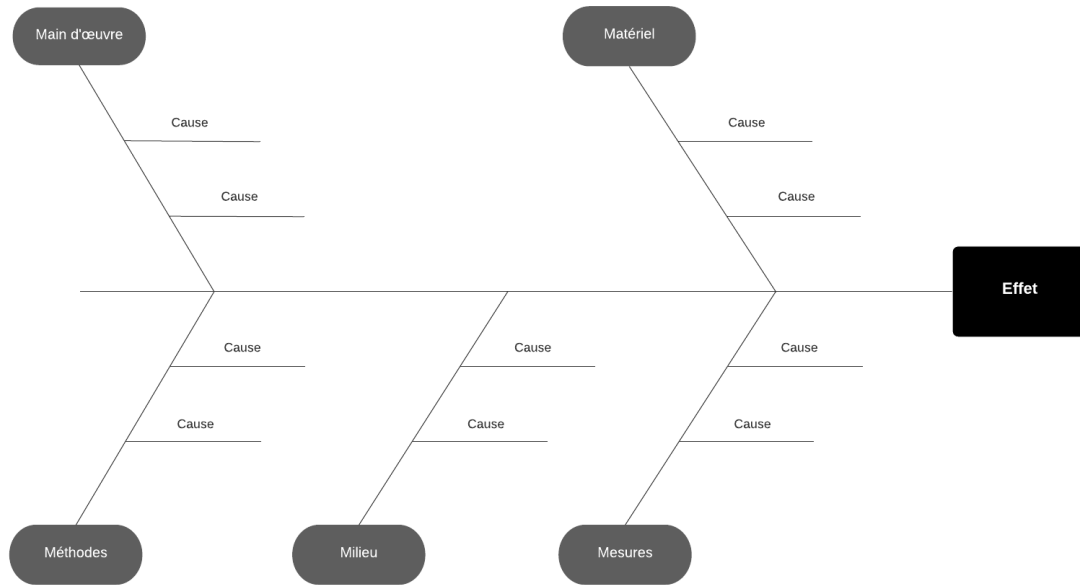


FIGURE 3.7 – Représentation du diagramme de Fishbone

En figure 3.8, le diagramme résume les différents facteurs tendant à expliquer l'insatisfaction client lors du processus de réservation de tickets au théâtre Medallion. Ce diagramme se veut d'un assez haut niveau. L'objectif recherché ici est d'offrir un support visuel.

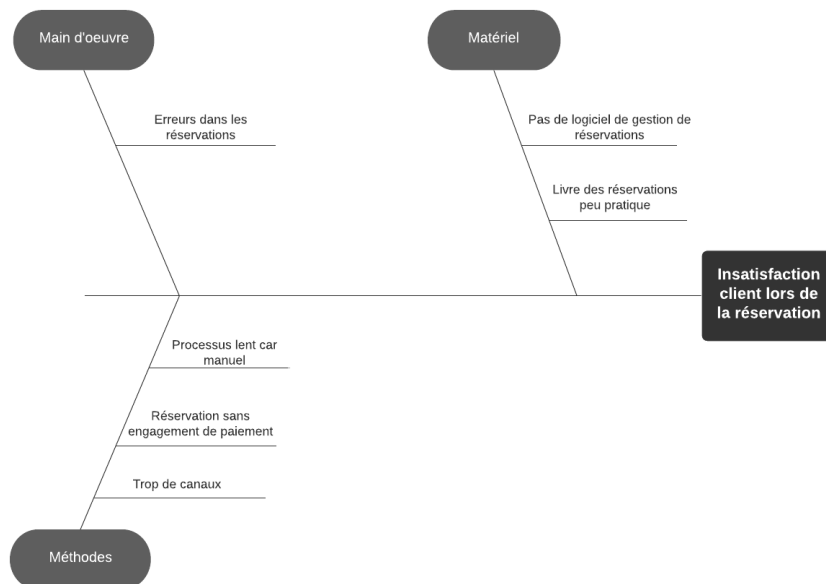


FIGURE 3.8 – Présentation des différentes causes de l'insatisfaction client dans la réservation de tickets au théâtre Medallion

### 3.4 Analyse du système futur

Une fois que les causes ont été identifiées, l'analyste métier doit identifier et déterminer une série de ressources qui permettraient de solutionner les causes [Institute, 2015].

Il existe différentes méthodes pour les déterminer. Dans ce travail, nous utiliserons la « capability table » (tableau des possibilités/capacités, en français). Ce tableau est présenté en trois colonnes : les problèmes ou limitations actuelles, les causes sources et, enfin, les ressources nécessaires pour solutionner ces dernières. Ce tableau offre donc l'avantage et la possibilité de déterminer une (ou plusieurs) ressource(s) spécifique(s) pour chaque cause. [Institute, 2015].

La table 3.3 présente les différentes solutions proposées pour le théâtre Medallion. Ces solutions sont les exigences métiers du client. On remarque que l'automatisation d'une partie du processus de réservation des tickets résout une majorité des causes sources. La suggestion d'un système informatisé émise par le client est donc cohérente. On constate aisément qu'une automatisation d'une partie du processus permettrait d'éradiquer les causes existantes, responsables de l'insatisfaction client.

| Problème              | Causes  | Ressources   |
|-----------------------|---|--|
| Insatisfaction client | Lenteur du processus de réservation                   | Automatisation d'une partie du processus                                     |
|                       | Réservation sans engagement de paiement               | Mise en place d'un processus de réservation engagement de paiement           |
|                       | Canaux de réservation trop nombreux et trop superflus | Réservation uniquement sur place et sur un site de vente de tickets en ligne |
|                       | Erreur dans les prises de réservations                | Automatisation de la gestion du livre des réservation                        |
|                       | Livre des réservations non-pratique et non-performant | Automatisation de la gestion du livre des réservation                        |

TABLE 3.3 – Identification des solutions pour le théâtre Medallion

### 3.5 Analyse des livrables

Après avoir déterminé les ressources requises, il est essentiel d'identifier les écarts entre les ressources actuelles et requises. De cette manière, l'identification des écarts permettra d'identifier des livrables pouvant être mis en place de manière à combler l'écart des ressources. Afin d'ajouter de la valeur pour le client, il est fréquent de proposer des alternatives. Le client les évaluera suivant différents critères, grâce à une étude de faisabilité, afin de ne retenir que la meilleure [Institute, 2015].

La table 3.4 identifie les livrables retenus par le théâtre Medallion afin de combler l'écart des ressources. On remarque que, dans la quasi-totalité des cas, il faut partir d'une feuille blanche. Dans les ressources informatiques requises, aucune n'existe actuellement.

Pour la suite et afin de limiter la longueur ce travail, nous nous focaliserons uniquement sur le développement du logiciel de réservation de tickets.

| Problème(s)           | Causes  | Ressources   | Livrables pour compléter les écarts  |
|-----------------------|---|--|--|
| Insatisfaction client | Lenteur du processus de réservation                   | Automatisation d'une partie du processus                                     | Développement d'un logiciel de réservation de tickets                                    |
|                       | Réservation sans engagement de paiement               | Mise en place d'un processus de réservation avec engagement de paiement      | Intégration du processus de paiement au processus de réservation (process redesign)      |
|                       | Canaux de réservation trop nombreux et trop superflus | Réservation uniquement sur place et sur un site de vente de tickets en ligne | Suppression du canal téléphonique et e-mail. Développement d'un site de vente de tickets |
|                       | Erreur dans les prises de réservations                | Automatisation de la gestion du livre des réservations                       | Développement d'un logiciel de réservation de tickets                                    |
|                       | Livre des réservations non-pratique et non-performant | Automatisation de la gestion du livre des réservations                       | Mise en place d'une base de données hébergée dans un cloud                               |

TABLE 3.4 – Identification des écarts pour le théâtre Medallion



## 3.6 Modélisation de l'environnement actuel et futur avec $i^*$

Une solution alternative ou complémentaire à l'analyse du système actuel et futur réalisée précédemment pourrait être l'utilisation du cadre de modélisation  $i^*$  afin de représenter l'environnement du système actuel et futur.

$i^*$  a été développé dans le but de modéliser et analyser l'environnement d'une organisation et ses systèmes d'information de manière semi-formelle. On distingue deux types de modèles : le modèle de dépendance stratégique et le modèle de relation stratégique. Le premier décrit les relations de dépendance entre les différents acteurs du domaine et le dernier met en lumière les différents intérêts des acteurs et la manière dont ces derniers sont traités [Yu, 1997].

Le concept d' $i^*$  est caractérisé par l'intentionnalité de l'acteur. Les acteurs ont des intentions comme des objectifs. C'est grâce à la relation de dépendance entre les acteurs que les objectifs sont atteints [Yu, 1997]. Cette dépendance joue un rôle crucial dans réalisation des objectifs.

### 3.6.1 Concepts de base

#### Modèle de dépendance stratégique

Le modèle de dépendance stratégique est constitué d'un ensemble de nœuds et de liens. Les nœuds représentent les acteurs et les liens les relations. On définit un acteur comme toute entité effectuant des actions dans le but d'atteindre un objectif. En fonction de la nature de la relation, un acteur peut être caractérisé comme le dépendant de l'autre acteur avec qui il est en relation et appelé alors le « dependeur ». Inversement, l'autre acteur sera alors responsable auquel le dependeur dépend et appelé le « dependee ». Le « dependum » caractérise l'objet de la relation [Yu, 1995].

L'objet de la relation ou dependum peut être de quatre types [Yu, 1995] et sont illustrés en figure 3.9 :

- Un but : condition ou état qu'un acteur veut atteindre ;
- Une qualité : condition ou état qu'un acteur veut atteindre mais à contrario de l'objectif, qui n'est pas précisément défini et a une connotation subjective ;
- Une tâche : une activité réalisée par un acteur ;
- Une ressource : une entité qui peut être physique ou informationnelle.



FIGURE 3.9 – Les objets de relations dans i\*

Par conséquent, entre acteurs il est possible de représenter quatre types de relations [Yu, 1995] entre le dependeur et le dependee, modélisés en figure 3.10.

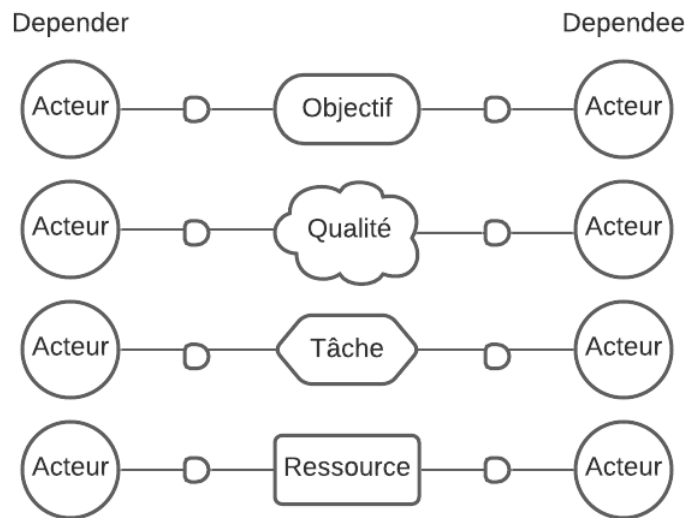


FIGURE 3.10 – Les relations de dépendance dans i\*

## Modèle de relation stratégique

Le modèle de relation stratégique se compose également d'un ensemble de nœuds et de liens. Ce modèle permet de présenter les intentions internes des relations entre les acteurs [Yu, 1995]. Les nœuds sont les mêmes que ceux présentés en figure 3.9.

On distingue deux types de liens :

- Le lien moyen-fin : il décrit une relation entre une fin (un objectif, une tâche, une ressource, ...) et le moyen par lequel elle est réalisée. Le moyen se décrit à l'aide d'une tâche ou d'un ensemble de tâches ;
- Le lien de décomposition des tâches : une tâche est composée d'un ensemble de composantes. Ces composantes sont reliées à la tâche à l'aide du lien de décomposition des tâches [Yu, 1995].

On retrouve quatre types de liens moyens-fins présentés en figure 3.11 :

- Le lien objectif-tâche : la fin est un objectif et le moyen une tâche ;
- Le lien ressource-tâche : la fin est une ressource et le moyen une tâche ;
- Le lien qualité-tâche : la fin est un soft-goal (qualité) et le moyen une tâche ;
- Le lien qualité-qualité : permet de hiérarchiser un ensemble de qualités [Yu, 1995].

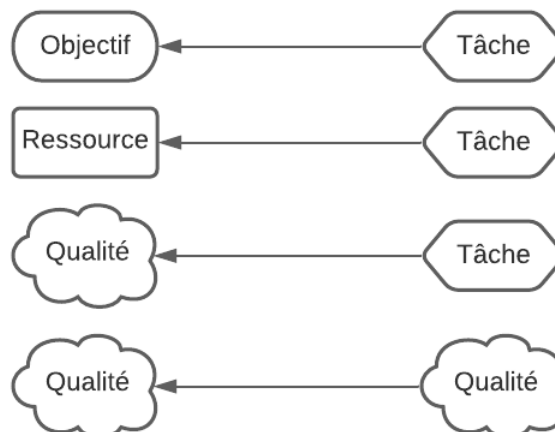


FIGURE 3.11 – Les types de liens moyens-fins dans  $i^*$

Il y a quatre types de liens de décomposition des tâches présentés en figure 3.12 :

- Le sous-objectif : étant, comme son nom l'indique, un sous-objectif que comprend la tâche ;
- La sous-tâche : dans la même logique que précédemment, une sous-tâche que comprend la tâche ;
- La ressource nécessaire : relie la tâche à une composante qui concerne l'utilisation d'une ressource ;
- La qualité nécessaire : relie la tâche à une composante qui concerne une manière dont la tâche doit être réalisée [Yu, 1995].

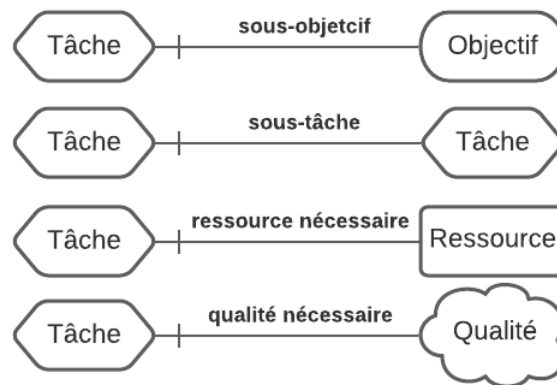


FIGURE 3.12 – Les types de liens de décomposition dans  $i^*$

### 3.6.2 Processus d'utilisation

Lors du développement d'un système d'information, Yu (1995) propose de commencer par utiliser le modèle de dépendance stratégique et de modéliser l'environnement d'abord sans le système d'information (situation actuelle) et ensuite avec le système d'information (situation future). En faisant de la sorte, il sera possible d'identifier les parties prenantes qui seront affectées par le changement et également de comprendre comment leurs intérêts se retrouveront impactés.

Toutefois, le modèle de dépendance stratégique n'offre qu'une vue externe des relations entre les acteurs. Le modèle de relation stratégique permet alors de comprendre plus en profondeur les processus internes des différents acteurs et de se rendre compte dans quelle mesure ces processus permettent d'atteindre les objectifs souhaités. Yu (1995) propose, de la même manière qu'auparavant, de réaliser un premier modèle sans le système d'information et, ensuite, avec. Réaliser un premier modèle sans le système d'information permettra de comprendre comment les activités se déroulent sans celui-ci. Il donnera donc des indications et perspectives sur les capacités et fonctions que devra assumer le futur système.

### 3.6.3 Application au cas d'étude

#### Diagrammes de dépendance stratégique

En analysant la figure 3.13 et ensuite la figure 3.14, on constate que le guichet et le programmeur, futurs utilisateurs du système, se retrouveront impactés par ce changement. De manière indirecte, le client sera impacté. Il n'y a pas de grands changements au niveau des intérêts. Cela s'explique par le fait que le guichet reste l'intermédiaire entre le client et le système de réservation.

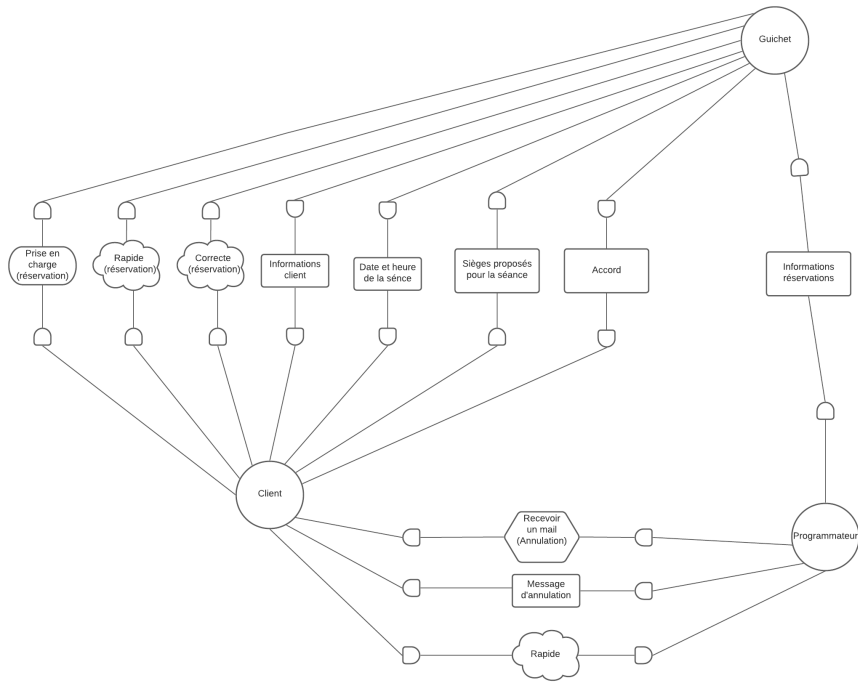


FIGURE 3.13 – Modèle de dépendance stratégique sans le système de réservation informatisé

### Diagrammes de relation stratégique

On voit également, en analysant la figure 3.15, et ensuite la figure 3.16, que le processus ne change pas de manière significative. Cela est toujours dû au fait que le guichet joue le rôle intermédiaire. Le changement se trouve surtout dans la réalisation des qualités telles que celles relatives aux efforts, à la rapidité et à la réservation correcte. Lorsque la réservation est prise en charge manuellement, nous pouvons remarquer un impact négatif sur ces diverses qualités. Un fois le système informatisé mis en place, il n'en est rien.

Le choix de garder le guichetier comme rôle intermédiaire entre le client et le logiciel est intentionnel. En effet, nous considérons que le théâtre Medallion souhaite garder un service d'assistance aux personnes venant réserver les tickets.

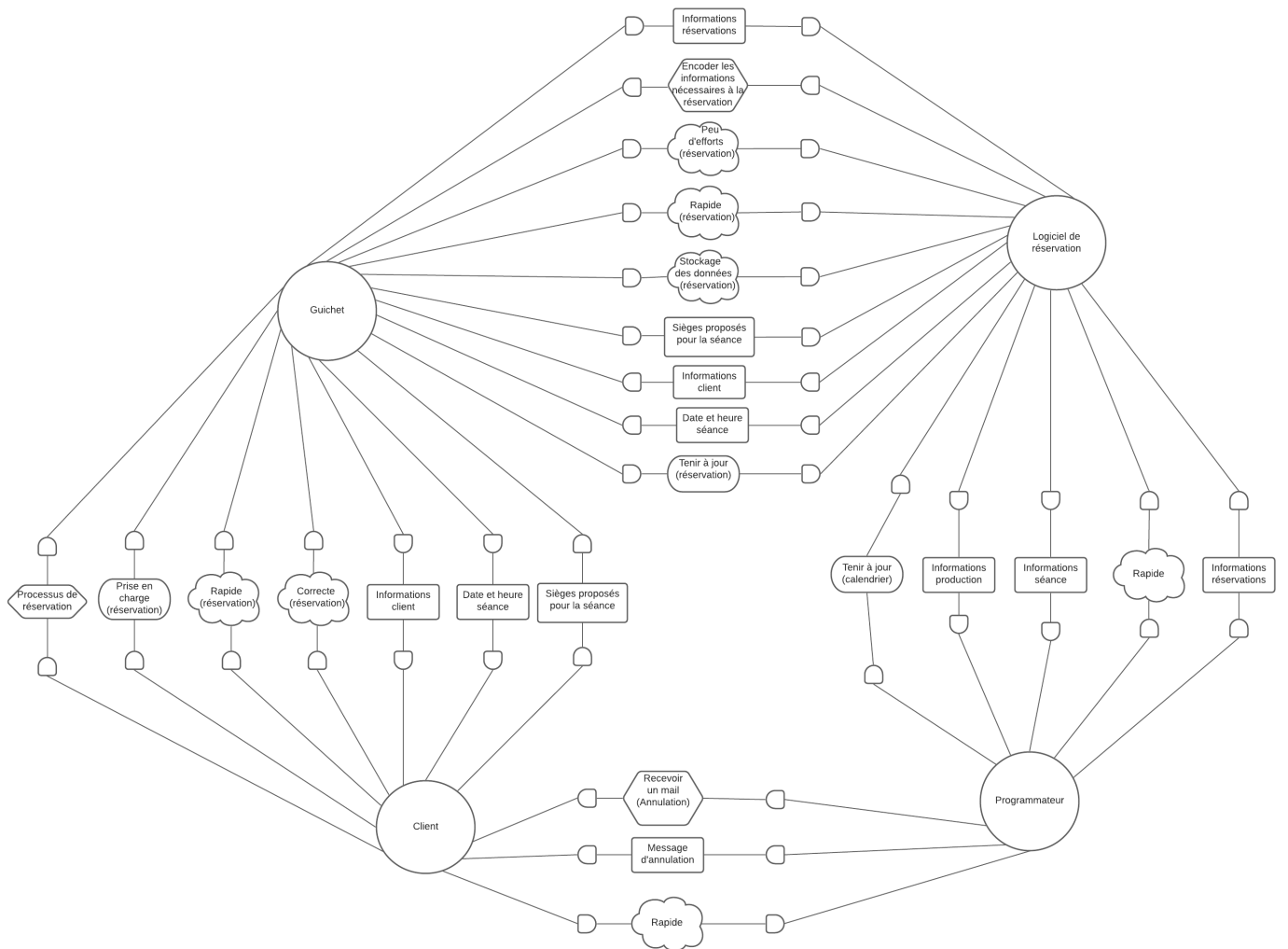


FIGURE 3.14 – Modèle de dépendance stratégique avec le système de réservation informatisé

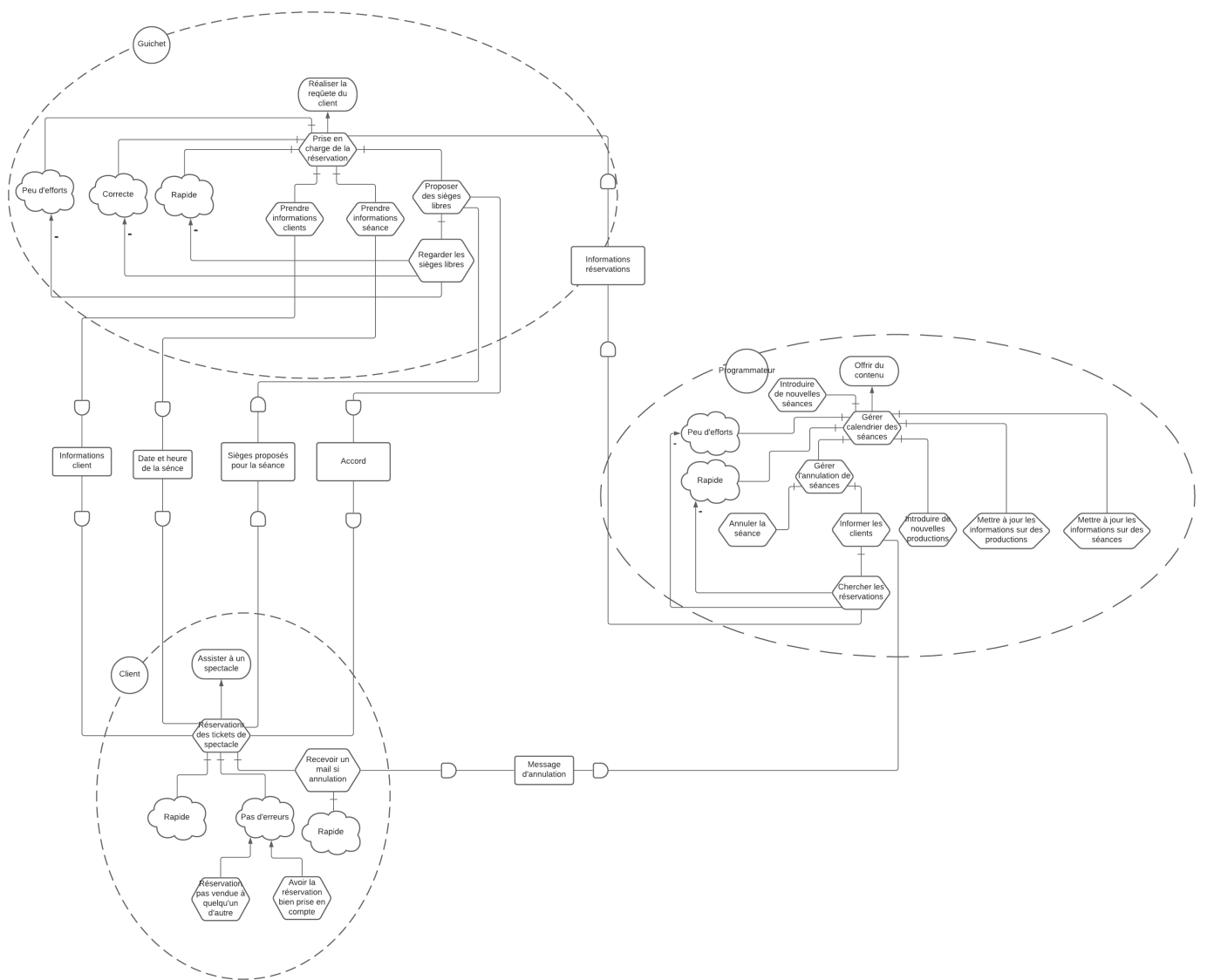


FIGURE 3.15 – Modèle de relation stratégique sans le système de réservation informatisé



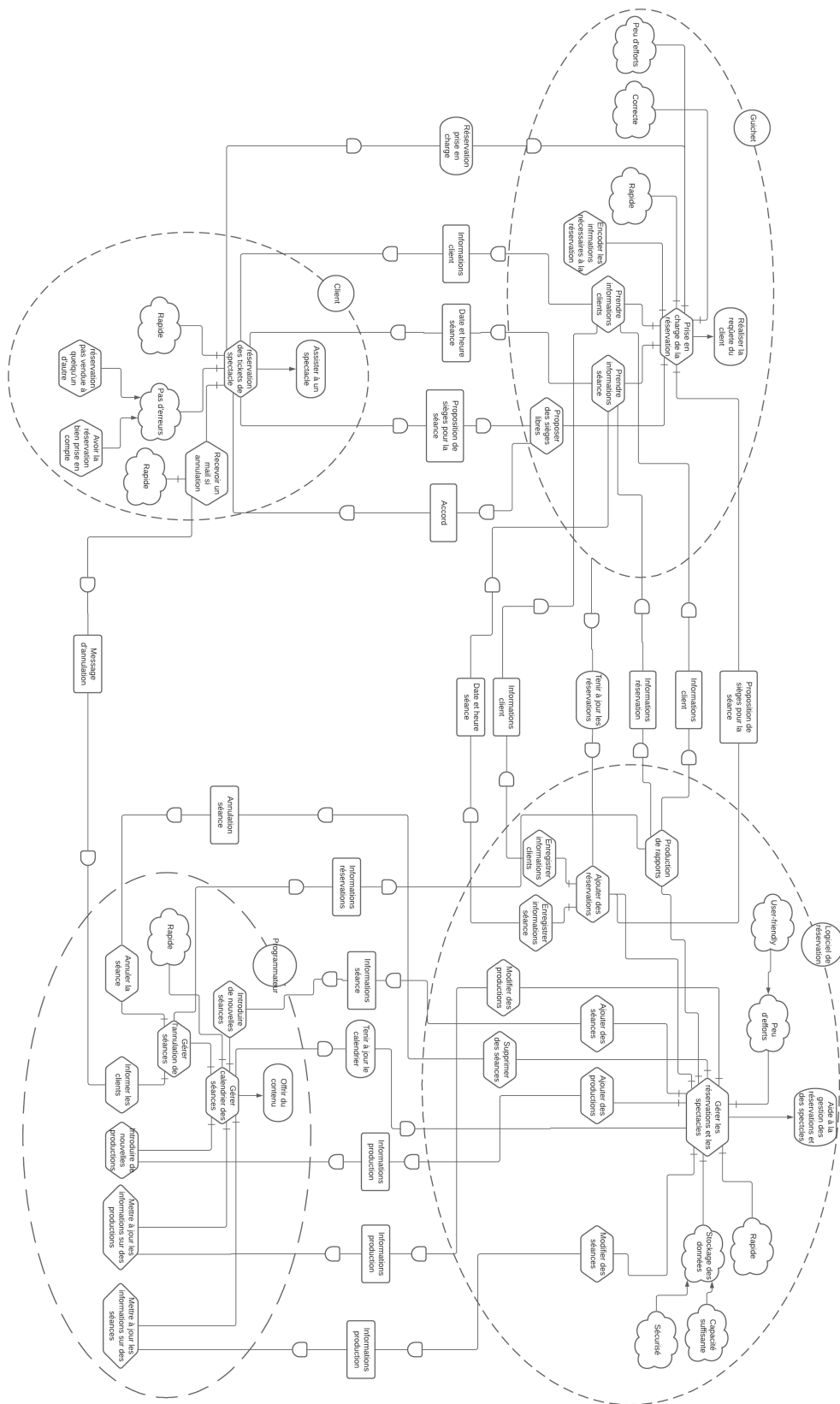


FIGURE 3.16 – Modèle de relation stratégique avec le système de réservation informatisé

# Chapitre 4

## Analyse de la solution

Cette partie se concentre sur les activités d'analyse du système de réservation. L'objectif de cette partie est d'éliciter, spécifier, documenter et valider les fonctionnalités et comportements que doit comprendre un logiciel dans un langage technique compris par les informaticiens. Bien entendu, ces fonctionnalités doivent correspondre aux exigences et attentes émises par l'utilisateur au début de cette analyse. L'activité centrale pour le business analyst dans cette phase est alors de documenter les exigences fonctionnelles dans un livrable appelé « Spécification des exigences logicielles (SRS) ». Ce livrable décrit de manière aussi complète que possible le comportement attendu du logiciel [Beatty et Wiegers, 2013].

Cette phase sera alors déclinée comme suit :

- Analyse du domaine du système ;
- Analyse des exigences relatives aux données ;
- Analyse des exigences fonctionnelles ;
- Analyse des exigences non fonctionnelles ;
- Analyse des exigences relatives à l'interface externe du système.

### 4.1 Les exigences logicielles

Il existe encore beaucoup de débats entre les praticiens du secteur autour de la définition d'une exigence logicielle. On peut aisément l'expliquer par le fait qu'il existe plusieurs types d'exigences n'étant pas de la même nature. De plus, ces exigences sont émises par diverses personnes dont le degré de connaissance technique varie. Par conséquent, une même exigence peut être vue comme d'un haut niveau technique pour quelqu'un, par exemple, un développeur. Alors que pour une autre personne, elle peut être perçue comme une exigence de bas niveau comme pour un utilisateur totalement néophyte en la matière

[Beatty et Wieggers, 2013].

Une définition complète des exigences est proposée par Ian Sommerville et Pete Sawyer : « *Requirements are a specification of what should be implemented. They are descriptions of how the system should behave, or of a system property or attribute. They may be a constraint on the development process of the system.* » (Cité dans Beatty et Wieggers, 2013, p. 51) (Les exigences sont une spécification de ce qui doit être mis en œuvre. Ce sont des descriptions de la façon dont le système doit se comporter, ou d'une propriété ou d'un attribut du système. Elles peuvent constituer une contrainte pour le processus de développement du système [notre traduction]).

Les exigences logicielles incluent une dimension temporelle. Elles peuvent être actuelles, déjà existantes. Elles décrivent alors les capacités du système actuel. Elles peuvent également concerner un futur. Plus ce futur est proche, plus l'exigence est considérée comme prioritaire comme fonctionnalité à inclure dans le logiciel [Beatty et Wieggers, 2013].

#### 4.1.1 Taxonomie des exigences logicielles

De manière générale, on divise les exigences logicielles en trois types : les exigences fonctionnelles, les exigences non fonctionnelles et les contraintes [Glinz, 2007]. Les exigences fonctionnelles sont des exigences qui « *specifies a function that a system or system component must be able to perform.* » [IEEE, 1990, p.35] (spécifie une fonction qu'un système ou un composant du système doit être capable d'exécuter [notre traduction]). Les exigences non fonctionnelles sont toutes les propriétés qu'un système doit posséder ou les contraintes qu'il doit respecter [Beatty et Wieggers, 2013]. Les contraintes sont l'ensemble des restrictions imposées qui limitent l'espace de la solution et sont en dehors de ce qui est considéré comme nécessaire pour répondre aux exigences fonctionnelles et non fonctionnelles [Glinz, 2007].

À cette taxonomie, il sera également ajouté :

- Les exigences relatives aux données du système (data requirements) : ces exigences feront référence à l'aspect des données du système ;
- Les exigences relatives à l'interface externe du système : ces exigences traiteront les caractéristiques de l'interface utilisateur, software et hardware, que doit comprendre le système [Beatty et Wieggers, 2013].

La figure 4.1 présente la taxonomie des exigences logicielles.

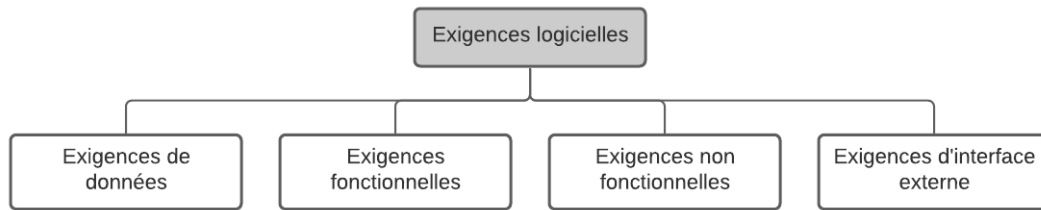


FIGURE 4.1 – Taxonomie des exigences logicielles

## 4.2 Affinement des parties prenantes

Au fur et à mesure que le projet avance, l'identification des parties prenantes doit être réaffinée et approfondie. En effet, l'analyse des stakeholders doit être vue comme un processus itératif où il est possible de découvrir de nouvelles parties prenantes au fil du temps. Lors de la phase de planification, il est pertinent de reconduire cette analyse et également d'analyser les rôles et responsabilités que les personnes auront tout le long de ce projet. Cette seconde analyse permettra un meilleur traçage et suivi [Institute, 2015].

Dans le cas où les stakeholders seraient en grand nombre, il est recommandé de les regrouper par leurs caractéristiques communes afin de simplifier l'analyse [Institute, 2015]. Pour ce faire, une technique existante est celle des personas. Cette technique consiste en la création de personnes caractéristiques et fictives. Elle permet donc de « stéréotyper » un groupe commun de stakeholders afin d'en comprendre leurs buts et activités qui aideront à développer la solution de manière conforme aux attentes et objectifs [Pruitt et Grudin, 2003]. Dans notre cas d'étude, le nombre de stakeholders restant raisonnable, il ne sera pas nécessaire de faire appel à cette technique.

Dans notre travail et étant donné que le projet est le développement d'une solution informatisée, l'identification des nouveaux stakeholders se composera alors de tous les acteurs du domaine IT qui joueront un rôle actif dans le développement et l'implémentation de la solution. Ces personnes peuvent être vues comme des parties prenantes à part entière : ce sont les acteurs du projet.

Les acteurs sont alors toutes les personnes impliquées dans l'analyse, la documentation, la conception et l'implémentation de la solution [Institute, 2015]. Dans un projet orienté technologie de l'information, il est possible d'avoir de nombreux et variés corps de métiers dédiés à son développement tant ce domaine a évolué et est devenu vaste et complexe [Wallace, 2020]. Dans le cadre de ce travail, nous nous limiterons à l'identification et l'explication des rôles fondamentaux. L'équipe de projet sera constituée d'un chef de projet spécialisé IT, d'un business analyst, d'un architect analyst, d'un designer UI, de

programmeurs ainsi qu'un testeur.

Dans un premier temps, le business analyst aura pour tâche de transcrire des besoins business en fonctionnalités que devra posséder le système informatique. Ces fonctionnalités seront alors traduites dans un langage plus technique, sous une forme fonctionnelle. Ensuite, le technical architect ou system architect prendra le relais pour présenter le design de la solution. Cette forme se veut plus technique et de moins haut niveau que celle produite par le business analyst. Il aura en outre la charge de définir les composantes que comportera la solution. Le designer UI est la personne responsable de la conception de l'interface du logiciel. Il a pour objectif de la rendre claire et ergonomique. Par la suite, le logiciel sera développé par des développeurs [Wallace, 2020, Dufour, s. d.]. Une fois développée, le testeur aura pour mission de vérifier si la solution est conforme aux exigences fonctionnelles et non fonctionnelles du client [Dufour, s.d.].

À cette étape, nous allons également définir les différentes responsabilités des parties prenantes. Cette analyse sera faite à l'aide d'une matrice RACI que le lecteur trouvera en table 4.1. Cette matrice de responsabilité linéaire reprend les différentes étapes du projet. Pour chaque étape et pour chaque partie prenante identifiée, on indiquera leur niveau de rôle à l'aide d'une lettre [Paul et al., 2014] :

- R : désigne la personne responsable de la mise en œuvre et le développement de la tâche ;
- A : désigne l'approbateur de l'activité. C'est à lui que revient le rôle décisionnel ;
- C : désigne toute personne consultée lors de la réalisation d'une tâche. Elle fournira des informations utiles et nécessaire à la réalisation de la tâche ;
- I : désigne les personnes tenues informées de l'avancement du projet. Ce type de rôle est en grande partie passif.

|                   | Responsable | Sponsor | Chef de Projet IT | Business analyst | Architect analyst | Designer UI | Développeurs | Testeur | Régulateur | Utilisateurs |
|-------------------|-------------|---------|-------------------|------------------|-------------------|-------------|--------------|---------|------------|--------------|
| Budget            | I           | R       | C                 |                  |                   |             |              |         |            |              |
| Analyse           | I           | I       | C                 | R                | C                 | I           | I            | I       | A          | C            |
| Design/Conception | I           |         | C                 | C                | R                 | I           | C            | I       | A          |              |
| Interface design  | I           |         | C                 | C                | C                 | R           | I            | I       | A          | C            |
| Implémentation    | I           |         | C                 | I/C              | C                 | I/C         | R            | I       | I          |              |
| Tests             | I           | I       | C                 | C                | C                 |             | C            | R       | I          |              |

TABLE 4.1 – RACI matrice du projet Medallion

## 4.3 Analyse du domaine

Étant donné que la solution à implémenter est, en réalité, un système d'information, nous savons que la mise en place d'une base de données sera nécessaire.

Le développement d'une vue logique de la base de données en question sera donc requise. Toutefois, cette vue logique n'est pas une pure représentation du domaine réel. Ceci s'explique en partie par les contraintes et pratiques liées aux systèmes de gestion de bases de données. Cet écart peut alors être problématique dans la communication autour du design de la base de données auprès d'utilisateurs complètement néophytes en la matière [Chen, 1976].

### 4.3.1 Le diagramme entité-association

Le diagramme d'entité-association permet de réduire cet écart en proposant une présentation de la structure des informations dans le domaine réel. Ce schéma, plus « fidèle » au monde réel, sera plus facile à comprendre par les utilisateurs. Par la suite, le business analyst pourra partir de ce modèle comme base pour développer la vue logique de la base de données. En passant par cette étape intermédiaire, on s'assure alors que les exigences en matière de données sont bien conformes [Chen, 1976].

En résumé, l'apport du diagramme d'entité-association est multiple :

- Il permet de représenter le domaine d'un point de vue entité-association ;
- Il assure une communication optimale avec les utilisateurs ;
- Il permet un bon alignement de la base de données du système par rapport aux exigences des utilisateurs.

### 4.3.2 Concepts de base

Le modèle entité-association est un modèle conceptuel. L'objectif de ce dernier est de représenter la structure des informations présentes au sein d'un domaine. Ce modèle se présente sous forme d'un diagramme [Song and Chen, 2009].

La figure 4.2 présente le diagramme de manière générique. Il est composé d'un ensemble d'entités liées entre elles par des relations d'associations. On retrouve, dans chaque entité, les différents attributs qui la composent. Dans les entités, les attributs de types identifiants se distinguent des attributs descriptifs grâce au soulignement du nom de l'attribut. Les associations sont également caractérisées par les cardinalités [Song and Chen, 2009].

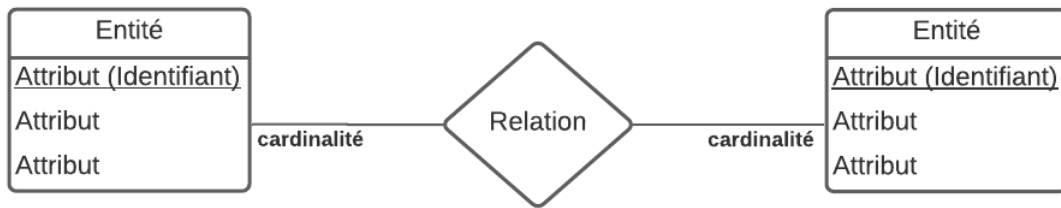


FIGURE 4.2 – Diagramme entité-association générique

### 4.3.3 Application au cas d'étude

Le diagramme entité-association réalisé pour le théâtre Medallion est présenté en figure 4.3. Il est composé de cinq entités : catégorie de siège, siège, client, séance et production.

On remarque que certaines classes ont des identifiants composés d'un ensemble d'attributs. En effet, une instance de la classe client est identifiée à l'aide de l'ensemble des attributs nom, prénom et numéro de téléphone. Une instance de la classe séance est identifiée à l'aide de la date et de l'heure. Une production est identifiée par son nom et son type car certaines productions de différents types peuvent avoir le même nom.

L'attribut « adresse » se trouve être un attribut composite. Il est composé des attributs suivants : numéro, rue, code postal, localité et pays.

On peut s'apercevoir que le modèle comprend une association ternaire entre les entités : siège, séance et client.

Concernant les cardinalités, deux « N » sont en rouges. Ces deux valeurs doivent être plus petites ou égales à 602, soit le nombre de places disponibles dans la salle du théâtre du Medallion. En effet, une catégorie de sièges peut posséder, au plus, 602 places. Dans la même logique, pour une séance, un maximum de 602 places peut être réservé.

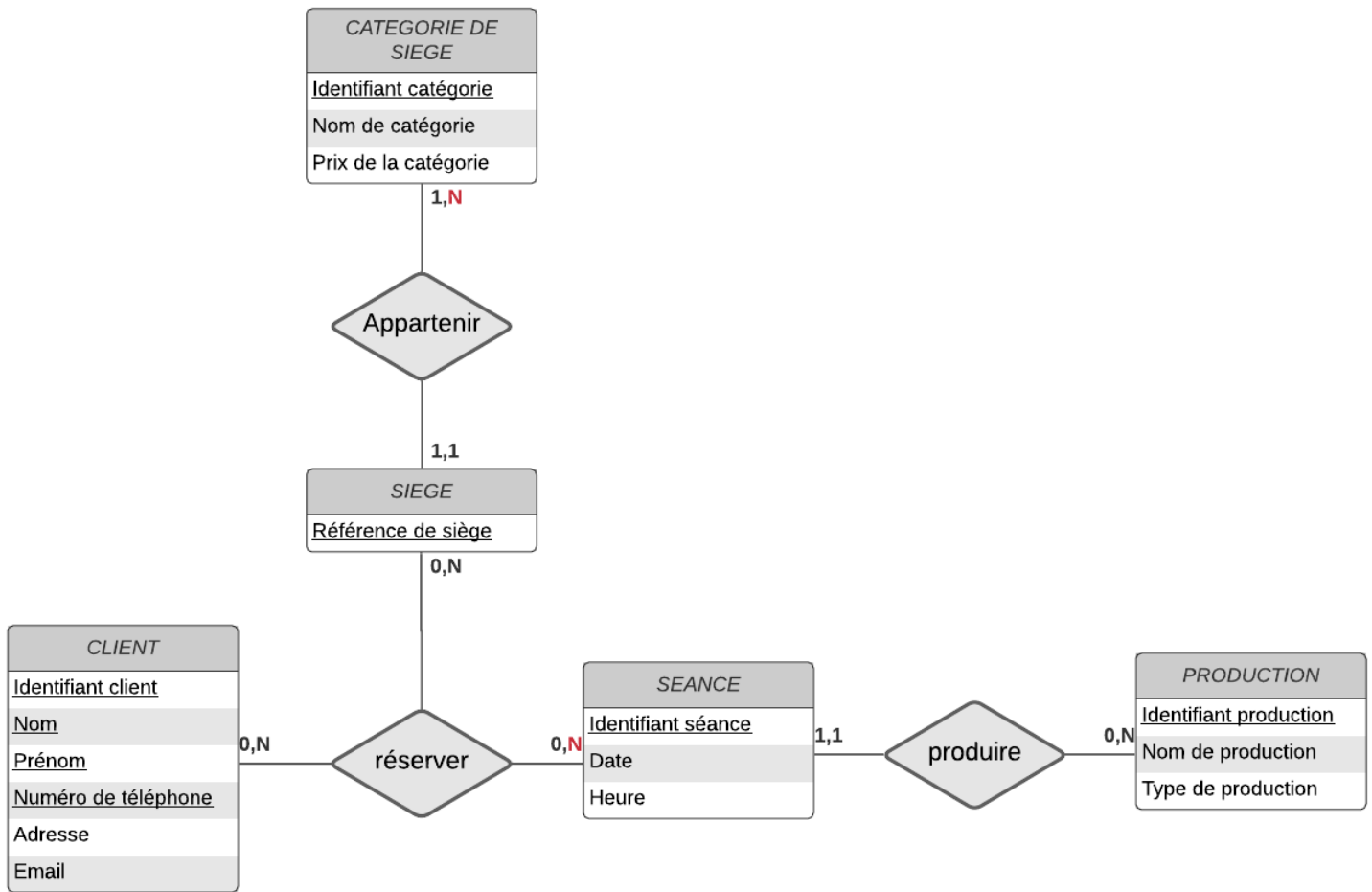


FIGURE 4.3 – Diagramme entité-association du théâtre Medallion

## 4.4 Analyse des exigences de données

Précédemment, nous avons constaté que le domaine de notre cas d'étude comportait un certain nombre d'informations organisées selon une structure représentée en figure 4.3. Ces informations devront être stockées dans une base de données. Il est alors primordial de pouvoir représenter la vue logique qu'aura cette dernière. Le modèle d'entité-association est une bonne base dans la conception logique de la base de données mais ce modèle ne respecte pas toutes les propriétés du modèle relationnel.

### 4.4.1 Diagramme relationnel

#### Concepts de base

Le diagramme relationnel, présenté de manière générique en figure 4.4, permet de représenter les données d'une base de données établie sur le modèle relationnel. Ce schéma est donc d'ordre logique puisqu'il respecte les contraintes du modèle relationnel [Hainaut, 2015]. Ainsi, ce schéma est composé :



- De types d'entités qui représentent les tables de la base de données relationnelle ;
- Au moins d'un attribut pour tous les types d'entités représentés, qui est, en réalité, une colonne de la table ;
- Les entités sont reliées entre elles [Hainaut, 2015].

Afin d'être conforme au modèle relationnel, le schéma doit respecter des contraintes :

- Un attribut doit être monovalué et atomique. Il peut être facultatif ou obligatoire ;
- Il n'existe que des contraintes relatives aux identifiants et attributs de référence ;
- Le schéma ne comporte aucune autre construction ;
- Les noms des objets doivent respecter la syntaxe SQL [Hainaut, 2015].

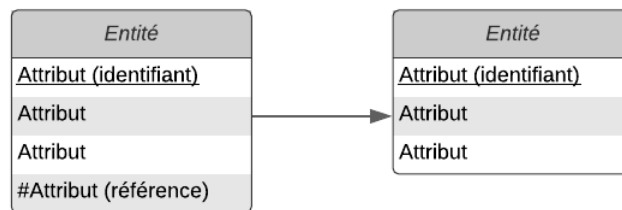


FIGURE 4.4 – Diagramme relationnel générique

## Processus d'utilisation

Afin de réaliser ce diagramme, il est recommandé de partir du modèle conceptuel préalablement réalisé. Dans notre cas, ce modèle est représenté par le diagramme entité-association. Il faudra ensuite :

- Représenter les types d'entités ;
- Représenter les relations ;
- Normaliser les relations ;
- Supprimer les généralisations [Chung, 2021].

## Application au cas d'étude

Le diagramme relationnel s'appliquant au cas d'étude est présenté en figure 4.5. La base de données comportera, non pas cinq, mais six tables, comme on aurait pu le croire au vu du diagramme entité-association. Il a également été pris la liberté d'ajouter des identifiants. Ce choix se justifie par le fait qu'il est plus pratique de travailler avec des identifiants sous un type spécifique en système et gestion des bases de données. Cela n'enlève en rien le fait qu'il soit toujours possible d'identifier une instance d'une classe à l'aide d'anciens identifiants spécifiés dans le diagramme d'entité-association.

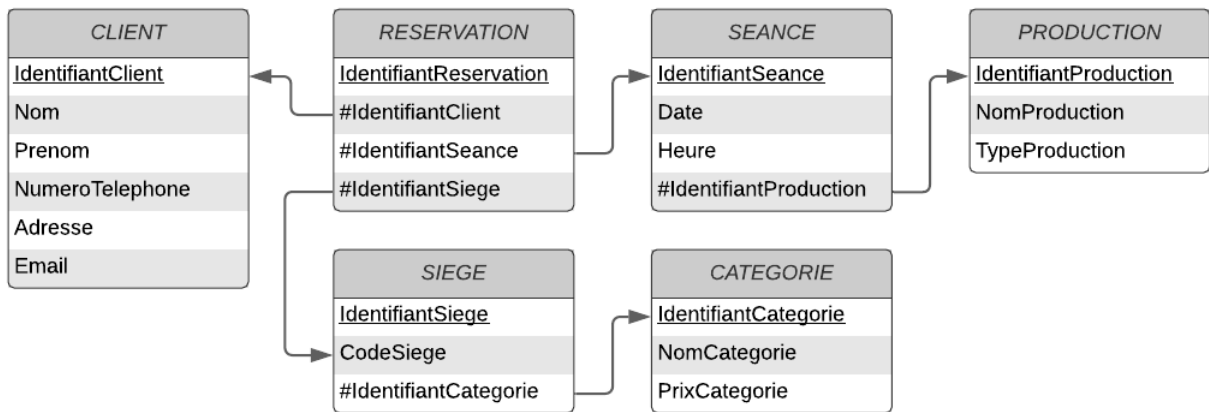


FIGURE 4.5 – Diagramme relationnel du théâtre Medallion

#### 4.4.2 Data dictionary

Il est également conseillé d'établir un document qui répertorie les données du système, appelé data dictionary. Il offre des informations complémentaires au diagramme relationnel comme une description des tables, des associations, des types des attributs. De plus, ce document augmente la possibilité de réutiliser ce livrable dans d'autres projets [Beatty et Wiegers, 2013]. Il n'existe pas de standard dans la réalisation de data dictionary. Pour réaliser le data dictionary du cas d'étude, nous utiliserons la forme de documentation proposée par Faulkner (2021)<sup>1</sup> dans le cadre du cours d'ICT Business analysis.

#### Application au cas d'étude

La table 4.2 documente les différentes classes existantes. On remarque que le data type de l'attribut « heure » est booléen. Cela est tout à fait normal car, selon les règles métiers indiquées dans l'énoncé, par jour, il n'y a que deux séances : l'une l'après-midi, l'autre le soir aux heures fixes.

La table 4.3 décrit les associations identifiées dans le diagramme entité-association (figure 4.3). Les lettres en rouges ont la même signification qu'expliqué lors de la réalisation du diagramme entité-association.

1. Source dérivée de WebCampus (non accessible au public)

| Classe      | Description                                  | Attribut               | Type    | Identifiant            |
|-------------|--|------------------------|---------|------------------------|
| Client      | Personne souhaitant réserver une séance      | IdentifiantClient      | Integer | IdentifiantClient      |
|             |  | Nom                    | String  |                        |
|             |  | Prénom                 | String  |                        |
|             |  | NumeroTelephone        | String  |                        |
|             |  | Adresse                | String  |                        |
|             |  | Email                  | String  |                        |
| Reservation | Réservation pour une séance de théâtre       | IdentifiantReservation | Integer | IdentifiantReservation |
|             |  | IdentifiantClient      | Integer |                        |
|             |  | IdentifiantSiege       | Integer |                        |
|             |  | IdentifiantSeance      | Integer |                        |
| Seance      | Pièce de théâtre à une date et heure précise | IdentifiantSeance      | Integer | IdentifiantReservation |
|             |  | Date                   | Date    |                        |
|             |  | Heure                  | Boolean |                        |
|             |  | IdentifiantProduction  | Integer |                        |
| Production  | Groupe chargé de la réalisation de la séance | IdentifiantProduction  | Integer | IdentifiantProduction  |
|             |  | NomProduction          | String  |                        |
|             |  | Type                   | Boolean |                        |
| Siege       | Place dans la salle de théâtre               | IdentifiantSiege       | Integer | IdentifiantSiege       |
|             |  | CodeSiege              | String  |                        |
|             |  | IdentifiantCategorie   | Integer |                        |
| Categorie   | Catégorie du siège                           | IdentifiantCategorie   | Integer | IdentifiantCategorie   |
|             |  | NomCategorie           | String  |                        |
|             |  | PrixCategorie          | Float   |                        |

TABLE 4.2 – Data dictionary pour les classes

| Association | Description  | Entités impliquées |
|-------------|--|--------------------|
| Appartenir  | Un siège appartient à une catégorie de sièges  | Categorie (1,1)    |
|             |  | Siege (1,N)        |
| Réserver    | Un client peut réserver plusieurs séances et plusieurs sièges. Une séance peut comprendre des sièges réservés. | Client (0,N)       |
|             |  | Seance (0,N)       |
|             |  | Siege (1,N)        |
| Produire    | Une production produit des séances   | Seance (0,N)       |
|             |  | Production (1,1)   |

TABLE 4.3 – Data dictionary pour les associations

## 4.5 Analyse des exigences fonctionnelles

### 4.5.1 UML

Pour réaliser cette partie, nous utiliserons le langage de modélisation unifié (UML).

UML est une norme de langage de modélisation unifié orienté objet mis en place par l'Object Management Group (OMG). Il se compose d'un lot de diagrammes permettant de décrire et modéliser des aspects des systèmes informatiques conçus et développés selon le paradigme orienté objet. On distingue deux grands types de diagrammes en UML : les digrammes de structure et les diagrammes de comportement [Fowler, 2004]. Les diagrammes de premiers types ont pour fonction de représenter l'aspect statique du système alors que ceux du second type représentent l'aspect dynamique [Gabay et Gabay, 2008].

Dans ce travail, ne sont présentés qu'exclusivement les diagrammes et les concepts de base utilisés dans le cas d'étude et ce, de manière à limiter l'étendue du travail. Pour l'aspect statique, il y aura alors le diagramme de cas d'utilisation et le diagramme de classes. Pour l'aspect comportement, nous utiliserons le diagramme d'activités et le diagramme de séquences.

Comme expliqué plus avant, la modélisation offre de nombreux avantages, mais le choix de l'utilisation d'UML ne se limite pas à cela. Un autre point, encore non abordé jusqu'ici et qui optimise une bonne communication, est l'utilisation de standards, compris et reconnus par l'ensemble d'une communauté. C'est cette caractéristique qu'offre UML et c'est

également la raison de sa création [Fowler, 2004]. En effet, au début des années 90, il existait une cinquantaine de manières d’analyser de concevoir des systèmes en orienté objet. La création d’UML s’est alors inscrite dans une démarche d’offrir un ensemble d’outils unifiés qui formeraient alors les standards de l’analyse et de la conception orientée objet [Gabay et Gabay, 2008]. Dès lors, UML est considéré comme standard en modélisation orientée objet.

#### 4.5.2 Processus

Il est important de rappeler qu’UML n’est pas une méthode mais juste une norme de langage. Il n’existe pas standards dans la manière d’utiliser UML [Gabay et Gabay, 2008]. Les manière d’utiliser UML sont nombreuses et laissent place à de nombreux débats. C’est pourquoi un processus unifié (Unified Process) fut développé par les auteurs d’UML. Il permet de décrire le processus en rapport avec UML [Fowler, 2004]. L’UP se caractérise par son fondement à partir du diagramme des cas d’utilisation qui sera un moyen de structurer, guider et encadrer l’analyse. Ce processus se veut itératif et incrémental [Gabay et Gabay, 2008]. Une partie de ce processus sera utilisée pour réaliser l’analyse fonctionnelle de notre cas d’étude. Il comportera les étapes suivantes :

- Élaboration du diagramme de cas d’utilisation ;
- Priorisation des cas d’utilisation ;
- Élaboration des scénarios des cas d’utilisation (textuelle et graphique) ;
- Élaboration du diagramme de classes fonctionnel ;
- Élaboration des digrammes de séquences fonctionnels pour les scénarios des cas d’utilisation.

### 4.5.3 Élaboration du digramme de cas d'utilisation

#### Concepts de base

Le diagramme de cas d'utilisation permet de représenter un système par l'ensemble des cas d'utilisation de celui-ci. Ces derniers sont, en réalité, les besoins exprimés par les utilisateurs de ce système. Ce schéma est, dès lors, composé de différents acteurs ayant chacun un certain nombre de cas d'utilisation. La figure 4.6 représente un diagramme de cas d'utilisation générique [Gabay et Gabay, 2008]. Ces derniers décrivent des interactions typiques entre les utilisateurs et le système. Enfin, ce diagramme doit rester simple et d'un haut niveau [Fowler, 2004].

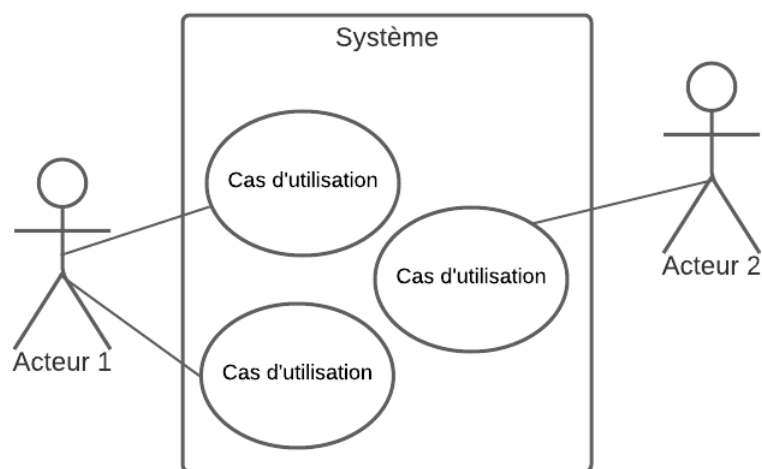


FIGURE 4.6 – Diagramme des cas d'utilisation générique

Dans ce diagramme, on autorise la réutilisation de certains cas d'utilisation à l'aide de relations entre ces derniers. On retrouve trois types de relations présentés en figure 4.7 :

- La relation d'inclusion (« includes ») : une instance du cas d'utilisation A contiendra le cas d'utilisation B ;
- La relation d'extension (« extends ») : une instance du cas d'utilisation A peut, dans certains cas, contenir le cas d'utilisation B ;
- La relation de généralisation : il est possible de réaliser des relations d'héritage entre les cas d'utilisation et d'obtenir des cas d'utilisation parents et enfants [Gabay et Gabay, 2008].

L'utilisation de ce diagramme s'avère être un excellent outil pour identifier les besoins exprimés par les utilisateurs. Il a l'avantage d'être facilement compréhensible par ces derniers. Le diagramme peut également être utilisé comme moyen d'organisation et de structure du développement du système. En effet, il permet de guider et d'encadrer le

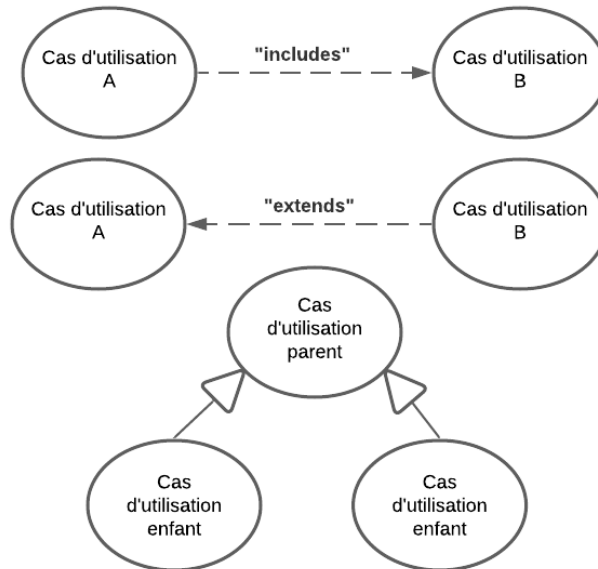


FIGURE 4.7 – Relations des cas d'utilisation

processus d'analyse du système en se basant sur les cas d'utilisation identifiés [Gabay et Gabay, 2008].

### Application au cas d'étude

La figure 4.8 présente le diagramme de cas d'utilisation pour notre cas d'étude. Le lecteur pourra remarquer que les cas d'utilisation se retrouvent également dans le modèle de relation stratégique d' $i^*$  (figure 3.16). Se baser sur le modèle  $i^*$  pour réaliser le diagramme des cas d'utilisation permet de s'assurer que les besoins exprimés par les utilisateurs restent bien alignés sur les objectifs du système et permet donc de définir un périmètre d'application dans l'identification des cas d'utilisation.

Dans le diagramme, les cas d'utilisation grisés sont des cas ayant été personnellement suggérés mais pas clairement explicités dans les consignes du cas d'étude (voir appendice 1).

On note que la réservation d'une place peut potentiellement demander à ajouter ou modifier un compte. En effet, un client doit posséder un compte pour réserver des places. Ajouter une production peut potentiellement déboucher sur l'ajout d'une séance. Aussi, modifier une production, supprimer une production ou modifier une séance peut éventuellement demander à annuler une séance. En effet, une réservation porte sur une séance et une production. Si les informations venaient à être supprimées ou modifiées alors que certains clients avaient réservé cette séance, il faudrait procéder à l'annulation de cette dernière.

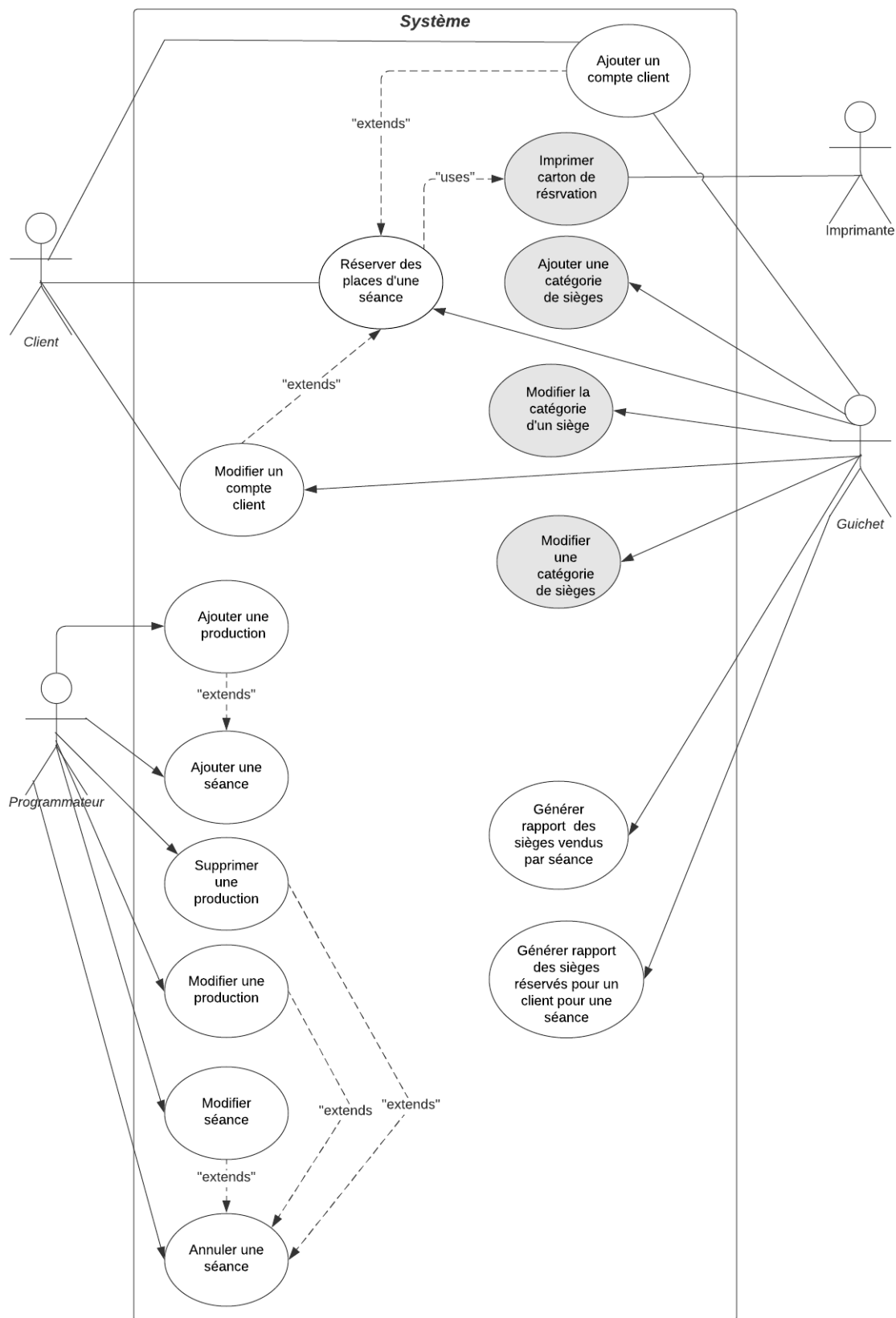


FIGURE 4.8 – Diagramme des cas d'utilisation du théâtre Medallion



#### 4.5.4 Priorisation des cas d'utilisation

L'ensemble des cas d'utilisation précédemment identifiés définissent, à un haut niveau, les grandes fonctionnalités du système. Cependant, ces besoins se distinguent selon différentes caractéristiques :

- Le degré d'importance de cette fonctionnalité ;
- Le niveau d'allocation des ressources nécessaires (temps, technicité, budget, ...) pour développer cette fonctionnalité [Firesmith, 2004].

Par conséquent, il est primordial de prioriser les cas d'utilisation afin de délivrer une valeur métier optimisée. Pour ce faire, il existe de nombreuses techniques de priorisation. Pour ce travail, la technique MoSCoW sera présentée et utilisée pour notre cas d'étude.

#### Concepts de base

MoSCoW est proposée par le Project Management Institute (2015). Cette technique consiste à classer les besoins dans quatre catégories :

- Must have : fonctionnalité élémentaire dépendant de la réussite du projet ;
- Should have : fonctionnalité importante mais qui ne dépend pas de la réussite du projet ;
- Could have : fonctionnalité qui n'est pas importante et qui n'impacte en aucun cas la réussite du projet et qui peut alors être aisément écartée du projet ;
- Won't have : fonctionnalité non réalisée pour ce projet [Institute, 2015].

#### Application au cas d'étude

La table 4.4 définit la priorisation des cas d'utilisation. Tous les cas d'utilisation « must have » sont ceux ayant été explicitement cités dans l'énoncé du cas d'étude. Le reste des cas d'utilisation, grisés dans la figure 4.8, sont ceux suggérés personnellement. Ceux-ci se retrouvent alors dans la catégorie « should have », exception faite pour le cas d'utilisation « imprimer carton de réservation » considéré comme une fonctionnalité peu importante (could have). Pour la suite du travail, nous ne prendrons en considération que les cas d'utilisation de catégorie « must have ».

| Catégories  | Cas d'utilisation  |
|-------------|--|
| Must have   | Ajouter un compte client   |
|             | Réserver des places d'une séance                                   |
|             | Modifier un compte client  |
|             | Ajouter une production   |
|             | Ajouter une séance   |
|             | Modifier une production  |
|             | Modifier séance  |
|             | Annuler une séance   |
|             | Générer rapport des sièges vendus par séance                       |
|             | Générer rapport des sièges réservés pour un client pour une séance |
| Should have | Ajouter une catégorie de sièges                                    |
|             | Modifier la catégorie d'un siège                                   |
|             | Modifier une catégorie de sièges                                   |
|             | Supprimer une production   |
| Could have  | Imprimer carton de réservation                                     |

TABLE 4.4 – Priorisation des cas d'utilisation pour le théâtre Medallion

#### 4.5.5 Élaboration des scénarios des cas d'utilisation

A ce stade, les cas d'utilisation restent très généraux. Il convient de détailler ces différents cas de manière à identifier et analyser le comportement que doit avoir le système.

##### Description textuelle

La description textuelle consiste simplement à détailler dans un langage informel les interactions entre les acteurs et le système. La description d'un cas d'utilisation peut être articulée en plusieurs scénarios : un scénario principal, où tout se déroule normalement, et des scénarios alternatifs décrivant le comportement que le système doit avoir dans des cas particuliers.

Il n'existe pas de standard particulier à respecter pour décrire les cas d'utilisation. Toutefois, la description doit comprendre :

- Un objectif : qui est le résultat souhaité derrière la réalisation du cas ;
- Les acteurs concernés par ce cas d'utilisation ;
- Les conditions particulières requises avant l'exécution du cas, s'il y en a (préconditions) ;
- Les conditions particulières requises après l'exécution du cas (postconditions) ;
- Le scénario principal ;
- Les scénarios alternatifs, s'il y en a.

Bien qu'une description textuelle des différents scénarios des cas d'utilisation identifiée soit peu formelle, elle représente une bonne base. Cette description offre une première structure pour capter les informations lors des discussions des scénarios avec les utilisateurs. Ce support est facilement compréhensible et permet de passer en revue le flux d'activités avec les utilisateurs [Beatty et Wiegers, 2013].

## **Application au cas d'étude**

Le lecteur pourra trouver en appendice 2 les différents scénarios des cas d'utilisation retenus (must have).

## **Les digrammes d'activités**

Le diagramme d'activités permet présenter visuellement le flux logique des différents scénarios. Comme le spécifie la figure 4.9, le diagramme d'activités se compose d'un ensemble d'actions, d'états et de transitions entre ces états et actions. On considère une activité comme un ensemble d'actions. Une action se caractérise par le fait qu'elle modifie l'état du système. Un état, lui, se caractérise par une situation d'attente. Lorsqu'une action est achevée, une transition est déclenchée de manière à passer à l'action suivante [Gabay et Gabay, 2008].

Il est possible de retrouver des nœuds dans les digrammes d'activités. Dans notre cas d'étude, un seul type de nœud sera utilisé. Il est appelé le nœud de décision. Celui-ci permet de se diriger vers un flux parmi plusieurs sous le respect d'une condition précise [Gabay et Gabay, 2008].

L'intérêt d'utiliser ces diagrammes s'explique par le fait qu'ils permettent de traduire les scénarios détaillés textuellement dans un langage semi-formel. De ce fait, on réduit le problème d'informalité rencontré avec la description textuelle [Beatty et Wiegers, 2013].

Les diagrammes d'activités ont également un autre avantage. Un digramme permet de décrire l'ensemble des scénarios d'un use case, rendant ainsi la présentation plus condensée et visuelle étant donné sa présentation graphique [Fowler, 2004].

Un autre point qui a également son importance est qu'il est possible, dans les digrammes d'activités, de spécifier lorsqu'un objet est affecté par le flux d'activités. Étant donné que nous suivons une approche orientée objet, il est intéressant de s'apercevoir à quel moment les objets sont concernés. Dans un diagramme d'activités, ces objets sont représentés par un rectangle où le nom d'objet est souligné.



FIGURE 4.9 – Les éléments généraux du diagramme d'activités

### Application au cas d'étude

Le lecteur trouvera en appendice 3 l'ensemble des diagrammes d'activités qui s'appliquent au cas d'étude.

### 4.5.6 Élaboration du diagramme de classes fonctionnel

Dans la modélisation UML, le diagramme de classes représente un réel pivot dans le développement des exigences logicielles car il présente graphiquement le système à développer [Gabay et Gabay, 2008].

À l'image du paradigme de programmation orienté objet, le diagramme présente un ensemble de classes liées entre elles par des associations. Cette classe se compose d'attributs étant des propriétés similaires à un groupe d'objets et des opérations étant des comportements que partagent un même groupe d'objets. On appelle également objet, l'instance de la classe. Comme dans le diagramme d'entité-association, on retrouve les cardinalités que l'on appellera, dans ce cas-ci, des multiplicités. Elles permettent de spécifier à un domaine de valeurs le nombre d'instances possibles d'une autre classe pour une association [Gabay et Gabay, 2008, Fowler, 2004].

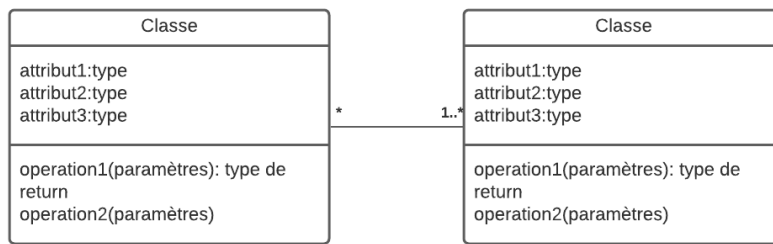


FIGURE 4.10 – Les classes du digramme de classes

Il est possible de déclarer la visibilité des attributs et méthodes. Il y a alors la visibilité publique, déclarée par le symbole « + » qui autorise toutes les classes à utiliser l'attribut ou la méthode. On déclare la visibilité protégée, qui n'autorise que les classes descendantes à utiliser les attributs ou opérations par le symbole « # ». Enfin, la visibilité privée est déclarée par le symbole « - » afin de n'autoriser que la classe elle-même d'utiliser les fonctionnalités.

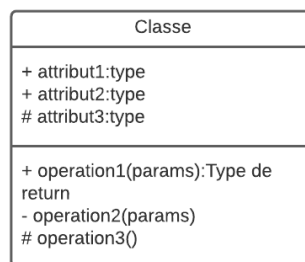


FIGURE 4.11 – Visibilité des attributs et méthodes

Il est aussi possible de spécifier les contraintes sur les attributs, les associations et les relations entre ces deux derniers. Pour ce faire, il suffit de les ajouter dans le diagramme sous cette forme : {contrainte}.

## Processus

Dans un premier temps, nous repartirons du digramme relationnel pour construire le diagramme de classes. L'objectif ici est de construire un diagramme de classes fonctionnel qui décrit en profondeur la couche métier. Le diagramme de classes de conception, son apport et la différence avec le diagramme de classes fonctionnel sera discuté plus tard.

## Application au cas d'étude

Le diagramme de classe fonctionnel du théâtre Medallion est présenté en figure 4.12.

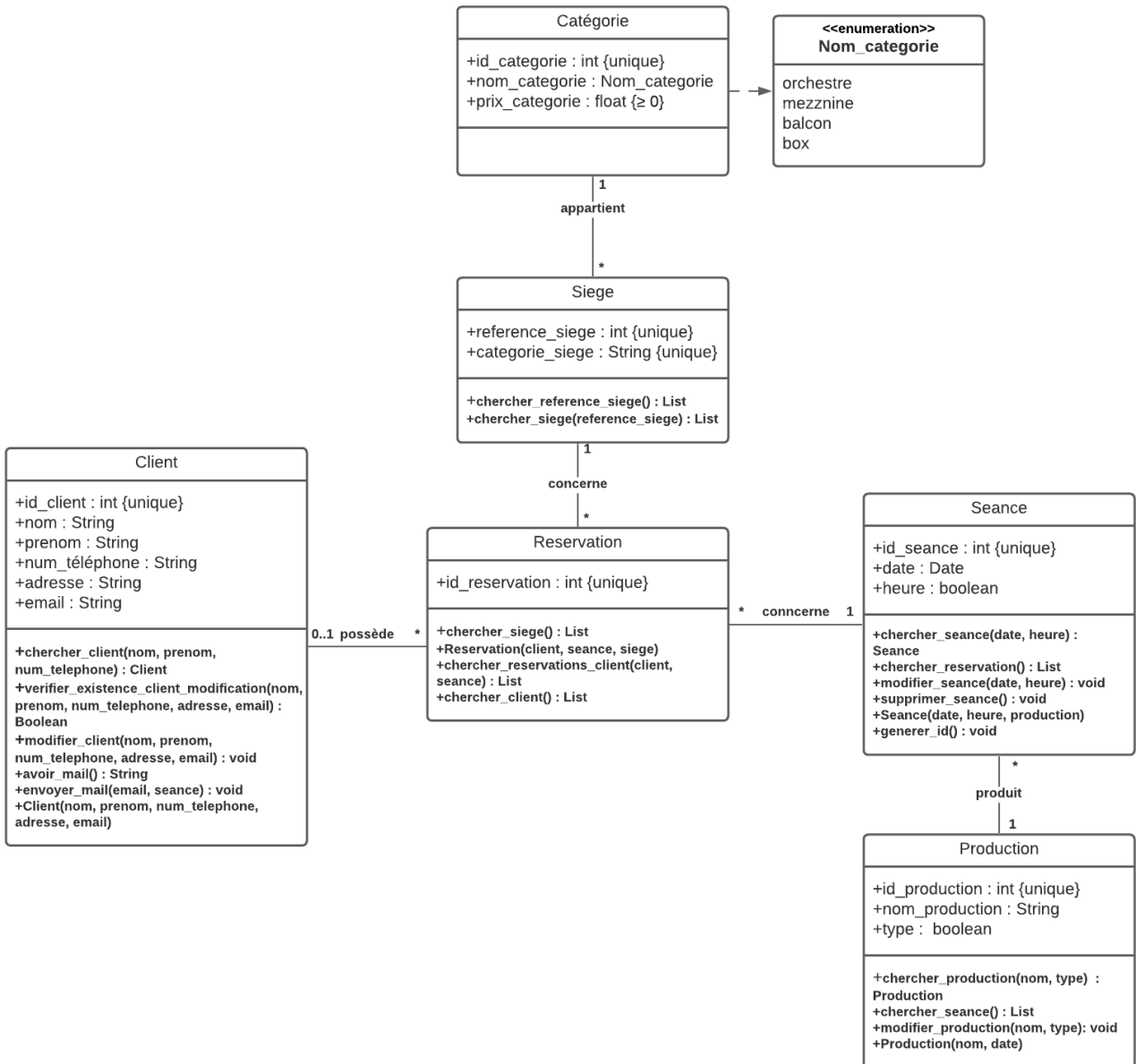


FIGURE 4.12 – Diagramme de classes fonctionnel du système de réservation du théâtre Medallion

## 4.5.7 Élaboration des diagrammes de séquences

### Concepts de base

Le diagramme de séquences permet de montrer graphiquement et de manière chronologique les interactions ayant lieu entre les objets lors des différents scénarios des cas d'utilisation. Il existe alors un diagramme de séquences par scénario [Fowler, 2004].

Comme le présente la figure 4.13, on retrouve l'acteur et les différents objets. Une particularité de ce diagramme est le concept de lignes de vie. Les lignes de vies, représentées par des rectangles sur les lignes, dépeignent l'ensemble des opérations exécutées par l'objet.

Les objets s'envoient des messages sous forme d'opérations. Ces messages peuvent être synchrones ou asynchrones. Dans notre cas d'étude, nous n'utiliserons que les messages synchrones. Le retour d'information peut être spécifié à l'aide de messages de retours représentés par les flèches en pointillé. De manière générale, ces retours d'informations ne sont pas spécifiés car implicites [Gabay et Gabay, 2008 ; Fowler, 2004].

Il est également possible d'indiquer la création d'un nouvel objet et la destruction de celui-ci lorsque l'on veut que cela soit explicite. Un exemple générique est donné dans la figure 4.13

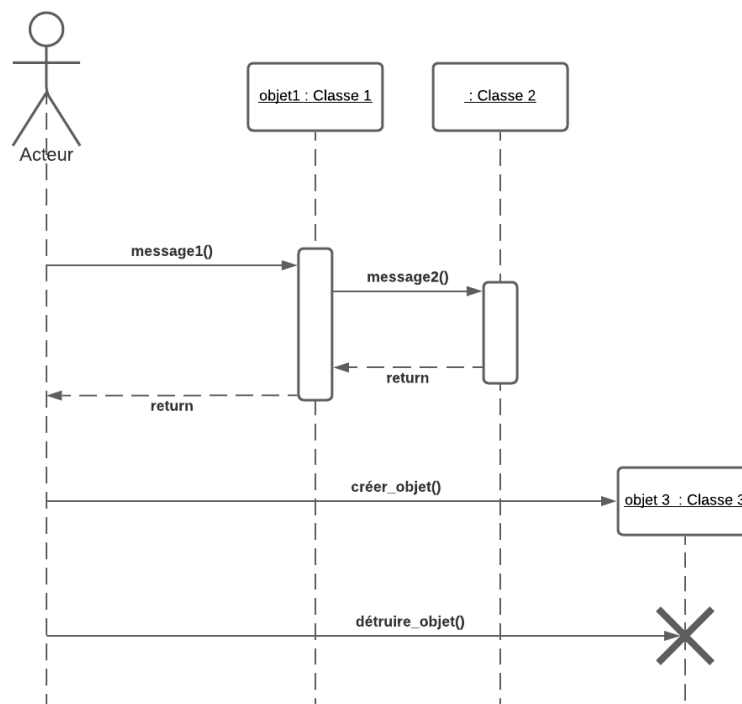


FIGURE 4.13 – Diagramme de séquences générique

L'intérêt d'utiliser les diagrammes de séquences s'inscrit dans un but de complétion du diagramme de classes. Premièrement, le diagramme de séquences permet d'offrir une vue dynamique des interactions entre les classes. Deuxièmement, développer les diagrammes de séquences permet de s'assurer de la complétude des méthodes et attributs dans les diagrammes de classes [Gabay et Gabay, 2008, Fowler, 2004].

### Application au cas d'étude

Afin de limiter le travail, seuls des diagrammes de séquences des scénarios principaux des cas d'utilisation ont été effectués. Exception faite au scénario alternatif de l'annulation de séance. L'ensemble de ces diagrammes sont présentés en appendice 4.

## 4.5.8 Documentation des règles métiers

### Taxonomie des règles métier

Comme déjà expliqué auparavant, les règles métiers ne sont pas considérées comme des exigences en tant que telles, mais des exigences peuvent découler de l'existence de ces règles métiers. Il est alors conseillé de les documenter [Beatty et Wiegers, 2013]. Pour notre cas d'étude, une simple documentation sous forme de tables sera suffisante.

Karl Wieger et Joy beatty (2013) propose une taxonomie des règles métiers. Elle en comprend cinq types :

- Faits : les faits sont des déclarations considérées comme vraies et actées à un certain moment dans le temps ;
- Contraintes : les contraintes sont l'ensemble des règles propres à l'activité métier qui contraignent le champ d'action et de possibilité du système informatique ;
- Déclencheurs : ce sont l'ensemble de règles qui provoquent le déclenchement d'actions spécifiques lorsqu'une condition particulière est atteinte ;
- Dérivation : ce sont l'ensemble des données transformées depuis des données de base ayant respecté un processus ou calcul défini par les règles métiers.

### Application au cas d'étude

Une partie des règles métiers de faits ont déjà été documentées dans le data dictionary, précédemment créé et disponible en figure 4.2 et 4.3 et peuvent donc être réutilisées comme documentation pour cette section. Les reste des règles métiers sont présentées dans la table 4.5.

| ID  | Description   | Type        |
|-----|---|-------------|
| RM1 | Le prix d'une catégorie de sièges doit être supérieur à 0€  | Contrainte  |
| RM2 | Un client doit posséder un compte client pour réserver des places de spectacle                          | Contrainte  |
| RM3 | Chaque jour de l'année, il y a deux séances à heures fixes. La première à 14h00, la seconde à 20h00.    | Fait        |
| RM4 | Il existe deux types de productions : concert ou théâtre  | Fait        |
| RM5 | Il existe quatre types de catégories de sièges ayant des prix différents (voir table 3.1)               | Fait        |
| RM6 | Les sièges ont un code unique spécifique (voire table 3.1)  | Fait        |
| RM7 | Si une séance est annulée, les clients ayant réservé des places pour cette séance sont avertis par mail | Déclencheur |

TABLE 4.5 – Règles métier du théâtre Medallion



## 4.6 Analyse des exigences non fonctionnelles

Pour rappel, les exigences non fonctionnelles sont toutes les propriétés qu'un système doit posséder ou contraintes qu'il doit respecter [Beatty et Wieggers, 2013]. Ces exigences non fonctionnelles varient par leur nature. Elles peuvent être externes ou internes.

Ces exigences non fonctionnelles ont une influence sur la décision d'architecture et de conception qui vont être prises. Par conséquent, ce type d'exigences n'est pas à négliger car il peut être très coûteux de devoir revoir la conception d'un système n'atteignant pas les exigences non fonctionnelles [Beatty et Wieggers, 2013].

### 4.6.1 Les exigences non fonctionnelles externes

Les exigences externes sont l'ensemble des caractéristiques observables lorsque le système est en marche [Beatty et Wieggers, 2013]. Ces exigences diffèrent par leur nature. De cette manière, on trouve des exigences qui détaillent :

- Disponibilité ;
- Installation ;
- Intégrité ;
- Interopérabilité ;
- Performance ;
- Fiabilité ;
- Robustesse ;
- Sécurité. [Beatty et Wieggers, 2013]

### 4.6.2 Les exigences non fonctionnelles internes

Les exigences internes ne sont pas observables. Elles concernent l'ensemble des règles et propriétés relatives à l'architecture et la conception du système, propres aux développeurs. Ces exigences ne sont pas à négliger non plus car elles peuvent avoir un impact sur la perception de la qualité du système du client [Beatty et Wieggers, 2013]. Ces exigences diffèrent par leur nature. De cette manière, on trouve des exigences qui détaillent :

- Efficience ;
- Modificabilité ;
- Portabilité ;
- Réutilisabilité ;
- Scalabilité ;
- Vérification. [Beatty et Wieggers, 2013]

### 4.6.3 Processus

Une manière de réaliser l'analyse des exigences non fonctionnelles est de partir de la taxonomie de cette dernière, présentée dans les sections précédentes. En fonction du type de projet, plus ou moins de types de nature d'exigences non fonctionnelles seront retenus. Les différentes natures d'exigences seront alors priorisées. Il s'agira ensuite de passer par une phase d'élicitation de manière à recueillir les attentes des utilisateurs et autres parties prenantes concernées concernant les différents attributs requis. Pour terminer, ces exigences doivent être spécifiées et documentées [Beatty et Wieggers, 2013]. Il est à noter qu'il est également intéressant de se baser sur les modèles i\* développés ci-avant. Ces derniers livrent déjà des indications quant aux exigences non fonctionnelles sous la forme de qualités, identifiées dans les modèles.

### 4.6.4 Outil

Il est difficile de spécifier de manière formelle et non ambiguë des exigences non fonctionnelles tant elles peuvent avoir un caractère très subjectif et flou [Beatty et Wieggers, 2013]. Une manière de spécifier les exigences non fonctionnelles de manière détaillée est d'utiliser le langage Planguage. Ce langage comprend un ensemble de mots clés permettant de décrire avec précision des exigences [Gilb, 2005]. L'ensemble des mots clés est tellement vaste qu'il est impossible de le présenter dans sa totalité dans ce travail. Par conséquent, ne seront présentés dans la table 4.6 uniquement les mots-clés utilisés pour l'application de notre cas d'étude.

| Mot-clé  | Description  |
|----------|--|
| TAG      | Identifiant qui permet de référencer une exigence                            |
| AMBITION | Il décrit le niveau de performance attendu de manière qualitative            |
| SCALE    | Il définit l'unité de mesure de la performance                               |
| METER    | Il définit la méthode d'évaluation de la performance                         |
| GOAL     | Il fixe l'objectif que doit atteindre la performance de manière quantitative |

TABLE 4.6 – Une partie des mots clés existant en Planguage

### 4.6.5 Application au cas d'étude

Les mots clés présentés plus avant permettent, entre autres, de spécifier les exigences non fonctionnelles concernant les performances du système. Les exigences non fonctionnelles requises pour le système de réservation ne seront pas spécifiées dans ce travail. Dans la table 4.7, il est présenté un unique exemple de spécification de la performance requise en termes de temps de réponse pour la génération des rapports des sièges vendus par séances

et des sièges réservés pour un client pour une séance spécifique.

| Mot-clé  | Description   |
|----------|---|
| TAG      | Performance.Rapport.TempsdeRéponse  |
| AMBITION | Temps de réponse rapide pour la génération de rapports  |
| SCALE    | Écart de temps en secondes entre le moment où l'utilisateur clique sur le bouton "OK" et le moment où le rapport est affiché à l'écran. |
| METER    | Test de chronométrage effectué sur 40 tests.  |
| GOAL     | Temps de réponse inférieur ou égal à 3 secondes   |

TABLE 4.7 – Spécification des performances requises en termes de temps de réponse pour la génération des rapports

## 4.7 Analyse des exigences relatives aux interfaces externes du système

Cette analyse permet d'assurer que le futur système communiquera correctement avec les utilisateurs et conformément à leurs exigences en la matière. On distingue trois types d'interfaces externes sur lesquelles des exigences peuvent être appliquées :

- L'interface utilisateur : il est question ici de s'intéresser aux caractéristiques nécessaires pour l'interface utilisateur (couleurs, résolution, standards visuels, ...);
- L'interface software : on décrit ici les connections entre le logiciel à développer et les autres composants logiciels (base de données, librairies, ...);
- L'interface hardware : on décrit les propriétés de chaque interface entre les composants software et hardware, s'il y en a [Beatty et Wiegers, 2013].

### 4.7.1 Application au cas d'étude

Pour notre cas d'étude, nous nous intéresserons seulement aux exigences relatives à l'interface utilisateur. Dans l'énoncé du cas, il est demandé de réaliser un système « user-friendly ». De manière à satisfaire les besoins d'utilisation, il a été établi une série de mock-up qui définit le prototype de la future interface utilisateur. Ces mock-up sont disponibles en appendice 5.

# Chapitre 5

## Conception de la solution

Une fois l'ensemble de la couche métier analysée, il faut ensuite pouvoir définir les spécifications de l'architecture technique du système. On entre alors dans une phase de conception du système qui présente et communique la manière dont le système devra être développé [TP026B, 2017].

La partie conception est généralement laissée à l'architect analyst qui travaillera alors de manière étroite avec le business analyst [Satzinger et al., 2015]. Toutefois, nous soutenons le fait que la conception, ou du moins une partie, peut être réalisée par le business analyst. En effet, force est de constater que les formations qui débouchent sur l'analyse business mettent de plus en plus l'accent sur le développement d'une culture et de connaissances techniques dans le domaine de l'IT. De plus, nous préférons voir le business analyst et l'architect analyst non pas comme un métier en tant que tel mais plutôt comme un rôle au sein d'un projet IT. Dès lors, une même personne pourra endosser ces deux rôles si tant est qu'elle possède assez de connaissances techniques et/ou une certaine expérience.

Par conséquent, une partie portant sur la réalisation de la conception du système sera traitée. Elle débouchera sur la réalisation de deux livrables : un diagramme de classes de conception accompagnée des diagrammes de séquences des scénarios principaux.

### 5.1 Processus

Pour développer la conception du système, il est possible de suivre les étapes du processus unifié (UP) dédiées à cette phase. Nous ne réaliserons que deux étapes dédiées à la conception du processus :

- Élaboration du diagramme de classes de conception ;
- Élaboration des diagrammes de séquences pour les scénarios principaux [Fowler, 2004].

Pour réaliser ces deux activités, il va de soi que nous continuerons à utiliser UML.

### 5.1.1 Élaboration du diagramme de classes de conception

Le diagramme de classes fonctionnel développé précédemment ne décrit pas la manière dont le logiciel va être développé. En effet, ce diagramme se basait sur le diagramme relationnel. Cependant, la logique derrière un modèle relationnel n'est pas la même que le paradigme de la programmation orientée objet. Il convient donc de redéfinir ce diagramme et de l'adapter à une logique orientée objet [Gabay et Gabay, 2008].

#### Concepts de base

Le diagramme de classes comporte d'autres caractéristiques qui n'ont pas été présentées avant. Il est possible de représenter des relations spécifiques comme, la relation d'héritage [Gabay et Gabay, 2008], telle que le montre la figure 5.1.

Comme le présente la figure 5.2, il est également possible de représenter les interfaces des classes.

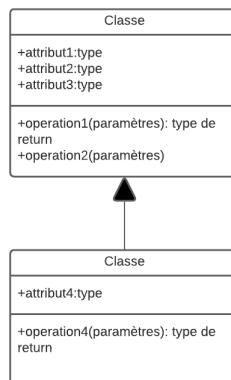


FIGURE 5.1 – Relation d'héritage dans un diagramme de classes

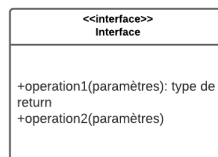


FIGURE 5.2 – Interface dans un diagramme de classes



## 5.1.2 Élaboration des diagrammes de séquences

L'intérêt et le principe des diagrammes de séquences restent les mêmes que lorsqu'utilisés pour l'analyse fonctionnelle.

### Application au cas d'étude

Le lecteur trouvera en appendice 6 les diagrammes de séquences relatifs aux diagrammes de classes de conception.

# Conclusion

L'objectif de ce mémoire a été l'identification et le développement d'un cadre méthodologique pour la business analysis. Le tout, proposé avec une étude de cas en parallèle.

Dans les différentes étapes et dans chaque chapitre, il a été question d'expliquer l'intérêt et la pertinence du travail réalisé. Il a également été présenté et utilisé un grand nombre d'outils, principalement de modélisation. Selon l'étape, ce travail s'est basé sur les pratiques et méthodes reconnues comme étant des standards dans la pratique.

La conclusion principale qui pourrait être faite à travers ce mémoire est que le rôle joué par le business analyst dans le bon développement des systèmes d'information est capital. On comprend très aisément qu'il est très risqué d'ignorer, ne serait-ce qu'une étape du travail du business analyst, s'il on souhaite que le système soit développé conformément aux exigences et attentes de l'entreprise.

Une deuxième conclusion pourrait être celle du constat que l'analyse business ne doit pas se réduire à l'analyse fonctionnelle du système. La phase précoce des exigences, qui précède cette analyse, joue également un rôle significatif dans la bonne réalisation d'un projet. Il ne faut pas oublier qu'un système est développé dans le but d'atteindre des objectifs business. Le système doit donc être aligné sur ces objectifs.

Toutefois, ce mémoire souffre de certaines limitations. En effet, étant donné que l'objectif de ce dernier était d'offrir un cadre méthodologique complet, il n'a pas été possible de parcourir tous les chapitres en profondeur. C'est le cas notamment pour l'analyse des exigences non fonctionnelles, l'analyse des exigences concernant les interfaces et la phase de conception. Ces parties méritent davantage de profondeur dans l'étude qui leur a été faite. Un travail dédié spécifiquement à l'étude et l'analyse des exigences non fonctionnelles serait souhaitable.

Aussi, il n'a pas été possible de confronter ce cadre méthodologique et plus particulièrement les outils de modélisation avec d'autres alternatives de manière à mettre en lumière les avantages et inconvénients de ces derniers.



# Bibliographie

- [Barsalou, 2014] Barsalou, M. A. (2014). *Root Cause Analysis : A Step-By-Step Guide to Using the Right Tool at the Right Time*. CRC Press. Google-Books-ID : qpTSBQAA-QBAJ.
- [Beatty et Wiegers, 2013] Beatty, J. and Wiegers, K. (2013). *Software Requirements Third Edition*. Microsoft Press, Washington, s4carlisle publishing services edition. original-date : 10/03/2021.
- [Borgida et al., 2009] Borgida, A. T., Chaudhri, V., Giorgini, P., and Yu, E. (2009). *Conceptual modeling : foundations and applications : Essays in honor of John Mylopoulos*, volume 5600. Springer Science & Business Media.
- [Chen, 1976] Chen, P. P.-S. (1976). The entity-relationship model – toward a unified view of data. *ACM transactions on database systems (TODS)*, 1(1) :9–36.
- [Chung, 2021] Chung, S. S. (2021). Translation of er-diagram into relational schema [diapositives].
- [Dufour, s. d.] Dufour, M. (s. d. ). Les acteurs d'un projet informatique. <http://www.fonctionnel.net/profession/acteurs-projet-informatique>. Consulté le 3 mars 2021.
- [Dumas et al., 2013] Dumas, M., La Rosa, M., Mendling, J., and Reijers, H. A. (2013). Introduction to business process management. In *Fundamentals of business process management*, pages 1–31. Springer.
- [Faulkner, 2021] Faulkner, S. (2021). Information system engineering life cycle [diapositives].
- [Firesmith, 2004] Firesmith, D. (2004). Prioritizing requirements. *J. Object Technol.*, 3(8) :35–48.
- [Fowler, 2004] Fowler, M. (2004). *UML distilled : a brief guide to the standard object modeling language*. Addison-Wesley Professional.
- [Fox, 2017] Fox, T. L. (2017). Teaching Note : Systems Analysis, Design and Development Case Study :Medallion Theater - Ticket Sales System. *Journal of the International Academy for Case Studies*.
- [Freeman, 2010] Freeman, R. E. (2010). *Strategic management : A stakeholder approach*. Cambridge university press.

- [Gabay et Gabay, 2008] Gabay, J. and Gabay, D. (2008). *UML 2 Analyse et conception : Mise en œuvre guidée avec études de cas*. Dunod.
- [Gilb, 2005] Gilb, T. (2005). *Competitive engineering : a handbook for systems engineering, requirements engineering, and software engineering using Planguage*. Elsevier.
- [Glinz, 2007] Glinz, M. (2007). On non functional requirements. In *15th IEEE International Requirements Engineering Conference (RE 2007)*, pages 21–26. IEEE.
- [Hainaut, 2015] Hainaut, J.-L. (2015). *Bases de données-3e éd. : Concepts, utilisation et développement*. Dunod.
- [Hong et Fauvel, 2013] Hong, Y. C. and Fauvel, C. (2013). Criticisms, variations and experiences with business model canvas.
- [Cernosek et Naiburg , 2004] Cernosek, G., & Naiburg, E. (2004). *The value of modeling*. IBM White Paper.
- [IBM, 2020] IBM. (2020). Document de vision pour un projet d'exigences. <https://www.ibm.com/docs/fr/elm/7.0.1?topic=requirements-vision-document>. Consulté le 17 mai 2021
- [IEEE, 1990] IEEE (1990). Ieee standard glossary of software engineering terminology. *IEEE Std 610.12-1990*, pages 1–84.
- [IIBA, 2015] IIBA (2015). *Babok : A Guide to the Business Analysis Body of Knowledge*. Number vol. 3. International Institute of Business Analysis.
- [Institute, 2015] Institute, P. M. (2015). *Business Analysis for Practitioners : A Practice Guide*. None edition, USA.
- [Isaias et Issa, 2015] Isaias, P. and Issa, T. (2015). *Introduction to Information Systems Models and Methodologies*, pages 1–19. Springer New York: New York, NY.
- [Jureta et al., 2008] Jureta, I. J., Faulkner, S., and Schobbens, P.-Y. (2008). Clear justification of modeling decisions for goal-oriented requirements engineering. *Requirements Engineering*, 13(2), pages 87-115.
- [Leau et al., 2012] Leau, Y. B., Loo, W. K., Tham, W. Y., and Tan, S. F. (2012). Software development life cycle agile vs traditional approaches. In *International Conference on Information and Network Technology*, 37, pages 162–167.
- [Milani, 2019] Milani, F. (2019). *Introduction to Business Analysis*, pages 1–29. Springer International Publishing, Cham.
- [Murugaiah et al., 2010] Murugaiah, U., Benjamin, S. J., Marathamuthu, M. S., and Muthaiyah, S. (2010). Scrap loss reduction using the 5-whys analysis. *International Journal of Quality & Reliability Management*.
- [Mylopoulos et al., 2001] Mylopoulos, J., Chung, L., Liao, S., Wang, H., & Yu, E. (2001). Exploring alternatives during requirements analysis. *IEEE Software*, 18(1), 92-96.
- [Nuseibeh et Easterbrook, 2000] Nuseibeh, B. and Easterbrook, S. (2000). Require-

- ments engineering : a roadmap. In *Proceedings of the Conference on the Future of Software Engineerig*, pages 35–46
- [OMG, 2021] OMG, O. M. G. (2021). Business Process Model & Notation™ (BPMN™) | Object Management Group.
- [Osterwalder et Pigneur, 2010] Osterwalder, A. and Pigneur, Y. (2010). *Business Model Generation : A Handbook for Visionaries, Game Changers, and Challengers*. John Wiley & Sons. Google-Books-ID : UzuTAAwAAQBAJ.
- [Pandey et al., 2010] Pandey, D., Suman, U., and Ramani, A. (2010). An effective requirement engineering process model for software development and requirements management. In *2010 International Conference on Advances in Recent Technologies in Communication and Computing*, pages 287–291.
- [Paul et al., 2014] Paul, D., Cadle, J., and Yeates, D. (2014). *Business analysis Third Edition*. BCS Learning & Development Ltd, bcs, the chartered institute for it edition. original-date : 18/03/21.
- [Pruitt et Grudin, 2003] Pruitt, J. and Grudin, J. (2003). Personas : practice and theory. In *Proceedings of the 2003 conference on Designing for user experiences*, pages 1–15.
- [Rooney and Heuvel, 2004] Rooney, J. J. and Heuvel, L. N. V. (2004). Root cause analysis for beginners. *Quality progress*, 37(7), pages 45–56.
- [Ruparelia, 2010] Ruparelia, N. B. (2010). Software development lifecycle models. *SIG-SOFT Softw. Eng. Notes*, 35(3), pages 8–13.
- [Satzinger et al., 2015] Satzinger, J., Jackson, R., and Burd, S. (2015). *Systems Analysis and Design in a Changing World*. USA, course technology inc ; 7e édition edition.
- [Signavio, 2012b] Signavio (2012b). Commandez dès maintenant votre Mémo BPMN 2.0gratuit. Signavio. <https://www.signavio.com/fr/telechargements/en-bref/memo-bpmn-2-0/>. Consulté le 1 mars 2021.
- [Song and Chen, 2009] Song, I.-Y. and Chen, P. P. (2009). *Entity Relationship Model*, pages 1003–1009. Springer US, Boston, MA.
- [Spradlin, 2012] Spradlin, D. (2012). Are You Solving the Right Problem ? *Harvard Business Review*.
- [TP026B, 2017] TP026B, R. (2017). Rational unified process. URL : [https://www.ibm.com/developerworks/rational/library/content/03July/10\\_00/1251/1251\\_bestprac-tices\\_TP026B.pdf](https://www.ibm.com/developerworks/rational/library/content/03July/10_00/1251/1251_bestprac-tices_TP026B.pdf) Cited October, zv, 18.
- [Van Lamsweerde, 2001] Van Lamsweerde, A. (2001). Goal-oriented requirements engineering : A guided tour. In *Proceedings fifth ieee international symposium on requirements engineering*, pages 249–262. IEEE.
- [Wallace, 2020] Wallace, S. (2020). Project Structure and Organisation, <https://www.epmbook.com/structure.htm>. Consulté le 4 mars 2021.

[Yu, 1997] Yu, E. S. (1997). Towards modelling and reasoning support for early-phase requirements engineering. In *Proceedings of ISRE'97 : 3rd IEEE International Symposium on Requirements Engineering*, pages 226–235. IEEE.

[Yu, 1995] Yu, E. S.-K. (1995). *Modelling Strategic Relationships for Process Reengineering*. PhD thesis, University of Toronto, Toronto.

La Business Analyse | IIBA Chapitre France. <https://france.iiba.org/fr/la-business-analyse>. Consulté le 17 mai 2021.

# Appendices

## Appendice 1 – Enoncé du cas d'étude

### **SYSTEMS ANALYSIS, DESIGN, AND DEVELOPMENT CASE STUDY: MEDALLION THEATRE – TICKET SALES SYSTEM**

**Terry L. Fox, University of Mary Hardin-Baylor**

#### **CASE DESCRIPTION**

*The primary purpose of this case study is for Systems Analysis and Design, Systems Development, and Database courses. Students examine realistic dialog and Interview Notes, as well as existing documents. For Systems Analysis and Design courses, the students should be able to follow this realistic and fairly common case study of a small organization and conduct the planning, analysis, and design phases of the System Development Life Cycle (SDLC), using either a traditional or object-oriented approach. Deliverables would include process and data diagrams and modeling, and user interface designs, and should require approximately 14-17 hours to complete, outside normal class time. In System Development courses, e.g., capstone courses for a computer information systems major, students can use this case study to not only analyze and design a solution, but actually develop the solution using various windows or web-based tools. The entire project should require approximately 22-28 hours to complete. For Database courses, this case could be used to illustrate database design techniques, resulting in the creation of appropriate data models and physical database designs. This should require approximately 10-12 hours to complete. The case study is of moderate difficulty – ranging from a four to five, and is designed for junior and senior level students, but could also be used for graduate courses.*

#### **CASE SYNOPSIS**

*Dr. Thomas Waggoner, an information systems professor at the local university, is at the Will Call window at the Medallion Theatre, trying to pick up tickets he had reserved. However, do to an oversight which turns out to be rather frequent, his tickets were sold to another patron. Fortunately for Dr. Waggoner and his wife, who are celebrating their wedding anniversary, the box office manager finds two box seats which had not been claimed. In talking with the box office manager, Dr. Waggoner starts thinking that he could perhaps help the theatre avoid this type of problem in the future. His students could design and build a system to help keep track of ticket sales, and hopefully help the theatre become more efficient.*

#### **INTRODUCTION**

Dr. Thomas Waggoner waited at the Will Call window for the evening's theatre production at the Medallion. He had reserved center orchestra seating tickets for he and his wife for this showing of Les Miz' more than two months ago, and for some reason the box office couldn't find his reservation and had sold his tickets to another patron.

"I am not sure what could have happened to your tickets. But we have already sold the seats you said you reserved – D18 and D19," said the young lady behind the glass. "We are sold out for tonight's performance."

“That’s very disappointing, particularly considering we are celebrating our anniversary tonight and have been looking forward to this for a long time,” responded Dr. Waggoner. “Are you very sure there are no seats left?”

Overhearing the conversation between the ticket clerk and Dr. Waggoner, the Box Office Manager moved over to the window.

“Dr. Waggoner?” she asked. “Actually I remember taking your phone reservation. I am so sorry. I wrote down your reservation on the phone list, but evidently it somehow wasn’t transferred to our master seating chart. I wish I could say this is the only time this has ever happened, but unfortunately it happens at least once a week. As it turns out, there are two box seats available tonight. The patrons who have those season tickets are not able to attend tonight - you can have those tickets. And, because it was our mistake, the tickets are on us – no charge.”

“Fair enough,” replied Dr. Waggoner. “What could have been a lousy evening looks like it will work out even better than planned!”

“Thank you for understanding. I hope we can figure out a better process so this doesn’t happen again.”

“Well, if you are interested, I think I could probably help you with that. I teach several systems analysis and development courses at the university, and I think my students could develop a computer system which can automate much of what you are currently doing manually. I can meet with you later this week to start talking about your processes and information needs,” offered Dr. Waggoner.

“That would be wonderful! Thank you so much for your understanding, and your offer to help us. Enjoy the show!”

Later that week Dr. Waggoner met with the Box Office Manager to develop an overall understanding of their business processes, the information they maintain, and the reporting needed. Dr. Waggoner compiled this information and presented it to a group of his students. The detailed requirements are listed below.

### **The Medallion Theater: Detailed Requirements**

The Medallion Theater Ticket Office keeps track of patrons, productions, performances, and seats. The Medallion likes to keep track of their patrons so they can notify them of upcoming events, and identify those who frequently support the theatre. Information kept on patrons include first and last name, street address, city, state, zip code, cell phone number, and email address. For the purposes of a computerized system, a unique patron number will need to be created.

With regard to productions, the Medallion maintains information on the name and type of the production, such as play or concert. With respect to performances, the theatre tracks the date of the performance and whether it is a matinee or evening performance. A production can, and usually does, have multiple performances, and can have both a matinee and evening performance on the same date.

The theater has a total of 602 seats. Each seat is available for each performance. The seats are in four categories, and the seat numbers are assigned as shown in the following table:

| <u>Seating Category</u> | <u>Seats</u>   | <u>Price</u> |
|-------------------------|--|--------------|
| Orchestra               | A1-A30, B1-B30, C1-C30<br>D1-D30, E1-E30, F1-F30         | \$65.00      |
| Mezzanine               | G1-30, H1-30, I1-30, J1-30<br>K1-30, L1-30, M1-30, N1-30 | \$55.00      |
| Balcony                 | AA1-30, BB1-30, CC1-30<br>DD1-28, EE1-24, F1-24          | \$40.00      |
| Box                     | X1-16  | \$85.00      |

When a patron calls or stops by to reserve a ticket, their name, address, and contact information is recorded, if they are not already in the files. The patron can then select the performance, including the date and whether it is matinee or evening. The patron then selects their seats from those available for a particular performance. The ticket thus identifies a specific seat for a specific performance.

Appropriate and user-friendly data entry/edit screens need to be created to enter and edit information for patrons and performances, as well as ticket sales for a specified seat selection.

The system should generate a report of seats sold or available for a particular performance, and should be able to show the seats purchased by a patron (based on name or patron number) for a specific performance.

# Appendice 2 – Scénarios des cas d’utilisation

## Ajouter un compte client

### Scénario principal

| <b>Précondition</b> : La personne n’a pas encore de compte client au théâtre Medallion  |  |  |
|---|--|--|
| <b>Postcondition</b> : La personne a un compte client   |  |  |
| Client  | Guichet  | Système  |
| La personne demande la création d’un compte client.<br>Elle indique son nom, prénom, adresse (rue, numéro, ville, code postal, pays), numéro de téléphone et son adresse email. | L’employé au guichet introduit les coordonnées de la personne. | Le système vérifie qu’il n’existe pas déjà un compte client existant.<br>Le système attribue un identifiant client unique au nouveau compte client.<br>Le système enregistre le nouveau compte client.<br>Le système indique que le nouveau compte client a bien été enregistré. |

### Scénario alternatif : Le client a déjà un compte client

| <b>Précondition</b> : Les coordonnées de la personne correspondent déjà à un compte client  |  |   |
|---|--|---|
| <b>Postcondition</b> : Le compte client n’est pas créé  |  |   |
| Client  | Guichet  | Système   |
| La personne demande au guichet d’être enregistrée en tant que client.<br>Elle indique son nom, prénom, adresse (rue, numéro, ville, code postal, pays), numéro de téléphone et son adresse email. | L’employé au guichet introduit les coordonnées de la personne. | Le système vérifie qu’il n’existe pas déjà un compte client existant.<br>Le système identifie un compte client déjà existant.<br>Le système signale qu’il existe déjà un compte client. |



# Ajouter un compte client

## Scénario principal

**Précondition :** La personne dont il faut modifier les coordonnées possède déjà un compte client au théâtre Medallion

**Postcondition :** Les coordonnées du client sont modifiées

| Client  | Guichet  | Système   |
|---|--|---|
| La personne demande au guichet de modifier ses coordonnées enregistrées. Le client indique son nom, prénom. | L'employé au guichet introduit le nom, prénom.           | Le système vérifie qu'il existe bien au moins un compte client correspondant au nom et prénom introduits. Le système affiche les coordonnées du ou des comptes existants correspondant au nom et prénom introduits. |
|   | L'employé sélectionne le compte correspondant au client. |   |
| Le client indique la ou les informations à modifier.  | L'employé introduit la ou les nouvelles informations.    | Le système enregistre les nouvelles coordonnées. Le système indique que les modifications ont bien été modifiées.   |

## Scénario alternatif : La personne n'a pas de compte client

**Précondition :** La personne dont il faut modifier les coordonnées du compte client ne possède pas de compte client.

**Postcondition :** Aucun compte client n'a été modifié.

| Client  | Guichet  | Système   |
|---|--|---|
| La personne demande au guichet de modifier ses coordonnées enregistrées. Le client indique son nom, prénom. | L'employé au guichet introduit le nom, prénom. | Le système vérifie qu'il existe bien au moins un compte client correspondant au nom et prénom introduits. Le système indique qu'il n'existe pas de compte client correspondant au nom et prénom introduits. |

# Ajouter une production

## Scénario principal

| <b>Précondition</b> : La nouvelle production à ajouter n'est pas déjà enregistrée.                              |  |
|---|--|
| <b>Postcondition</b> : Une nouvelle production est enregistrée.   |  |
| Programmeur   | Système  |
| Le programmeur introduit le nom de la production ainsi que le type de production (pièce de théâtre ou concert). | Le système vérifie qu'il n'existe pas déjà une production enregistrée au nom et type introduits.<br>Le système attribue un identifiant unique à la production.<br>Le système enregistre la nouvelle production.<br>Le système indique que la nouvelle production a bien été enregistrée. |

## Scénario alternatif : La production existe déjà

| <b>Précondition</b> : La production à ajouter existe déjà.  |   |
|---|---|
| <b>Postcondition</b> : Une nouvelle production est enregistrée.   |   |
| Programmeur   | Système   |
| Le programmeur introduit le nom de la production ainsi que le type de production (pièce de théâtre ou concert). | Le système vérifie qu'il n'existe pas déjà une production enregistrée au nom et type introduits.<br>Le système indique qu'une production correspondant au nom et type introduit est déjà enregistrée. |

# Ajouter une séance

## Scénario principal

| <b>Précondition</b> : L'heure et la date de la séance ne correspondent pas à une heure et une date d'une séance déjà enregistrée. La production existe. |   |
|---|---|
| <b>Postcondition</b> : La nouvelle séance est enregistrée.  |   |
| Programmeur   | Système   |
| Le programmeur introduit le nom de la production et le type de la production.   | Le système vérifie qu'il existe une production enregistrée au nom et type introduits.   |
| Le programmeur introduit le nom de la production et le type de la production.   | Le système vérifie qu'il n'existe pas déjà une séance enregistrée à la date et à l'heure introduites.<br>Le système attribue un identifiant unique à la séance.<br>Le système enregistre la séance.<br>Le système indique que la séance a bien été enregistrée. |

## Scénario alternatif : Une autre séance est déjà enregistrée

**Précondition** : L'heure et la date de la séance correspondent à une heure et une date d'une séance déjà enregistrée. La production existe.

**Postcondition** : La nouvelle séance n'est pas enregistrée.

| Programmeur   | Système  |
|---|--|
| Le programmeur introduit le nom de la production et le type de la production. |  |
| Le programmeur introduit le nom de la production et le type de la production. | Le système vérifie qu'il n'existe pas déjà une séance enregistrée à la date et à l'heure introduites.<br>Le système indique qu'il existe déjà une séance enregistrée à la date et à l'heure introduites. |

## Scénario alternatif : La production n'existe pas

**Précondition** : La production n'existe pas.

**Postcondition** : La nouvelle séance n'est pas enregistrée.

| Programmeur   | Système   |
|---|---|
| Le programmeur introduit le nom de la production et le type de la production. | Le système vérifie qu'il existe une production enregistrée au nom et type introduits.<br>Le système indique qu'aucune production n'est enregistrée au nom et type introduits. |

# Modifier une production

## Scénario principal

**Précondition** : La production dont il faut modifier les informations est déjà enregistrée. Il n'y a aucune réservation enregistrée pour une séance de cette production.

**Postcondition** : Les nouvelles informations de la production sont enregistrées.

| Programmeur  | Système  |
|--|--|
| Le programmeur introduit le nom de la production ainsi que le type de production (pièce de théâtre ou concert) dont il désire modifier les informations. | Le système vérifie qu'il existe bien une production au nom indiqué et au type indiqué.<br>Le système vérifie qu'il n'y a aucune réservation pour une séance de cette production.   |
| L'employé au guichet introduit le nouveau nom et/ou le nouveau type de la production.  | Le système vérifie qu'il n'existe pas déjà une production du même nom et du même type qui est enregistrée.<br>Le système enregistre les nouvelles informations.<br>Le système indique que les informations ont bien été modifiées. |

### Scénario alternatif : Il y a au moins une réservation pour une séance de cette production

| <b>Précondition</b> : La production dont il faut modifier les informations est déjà enregistrée. Il y a au moins une réservation enregistrée pour une séance de cette production. |   |
|---|---|
| <b>Postcondition</b> : Les nouvelles informations de la production sont enregistrées.   |   |
| Programmeur   | Système   |
| Le programmeur introduit le nom de la production ainsi que le type de production (pièce de théâtre ou concert) dont il désire modifier les informations.                          | Le système vérifie qu'il existe bien une production au nom indiqué et au type indiqué.<br>Le système vérifie qu'il n'y a aucune réservation pour une séance de cette production.<br>Le système indique qu'il y a de réservations enregistrées pour des séances de cette performance |

### Scénario alternatif : la production n'existe pas

| <b>Précondition</b> : La production dont il faut modifier les informations n'est pas enregistrée dans le système.  |   |
|--|---|
| <b>Postcondition</b> : Les nouvelles informations ne sont pas enregistrées.  |   |
| Programmeur  | Système   |
| Le programmeur introduit le nom de la production ainsi que le type de production (pièce de théâtre ou concert) dont il désire modifier les informations. | Le système vérifie qu'il existe bien une production au nom indiqué et au type indiqué.<br>Le système indique qu'il n'existe pas de production enregistrée au nom et au type introduits. |

### Scénario alternatif : Les nouvelles informations correspondent à une production déjà existante

| <b>Précondition</b> : La production dont il faut modifier les informations est déjà enregistrée. Il n'y a aucune réservation enregistrée pour une séance de cette production. Les nouvelles informations correspondent à une production déjà enregistrée. |  |
|---|--|
| <b>Postcondition</b> : Les nouvelles informations de la production ne sont pas enregistrées.  |  |
| Programmeur   | Système  |
| Le programmeur introduit le nom de la production ainsi que le type de production (pièce de théâtre ou concert) dont il désire modifier les informations.  | Le système vérifie qu'il existe bien une production au nom indiqué et au type indiqué.<br>Le système vérifie qu'il n'y a aucune réservation pour une séance de cette production.                         |
| L'employé au guichet introduit le nouveau nom et/ou le nouveau type de la production.   | Le système vérifie qu'il n'existe pas déjà une production du même nom et du même type qui est enregistrée.<br>Le système indique qu'il existe déjà une production enregistrée au nom et type introduits. |

# Modifier une séance

## Scénario principal

| <b>Précondition</b> : Aucun siège n'a déjà été vendue pour la séance à modifier. Les nouvelles informations (heure et/ou date) ne correspondent pas à celles d'une séance déjà enregistrée. |   |
|---|---|
| <b>Postcondition</b> : Les informations de la séance sont mises à jour.   |   |
| Programmeur   | Système   |
| Le programmeur introduit la date et l'heure de la séance qu'il souhaite modifier.   | Le système vérifie qu'il existe bien une séance enregistrée à la date et à l'heure introduites.<br>Le système vérifie qu'il n'y a aucun siège déjà réservé pour la séance.  |
| L'employé au guichet introduit le nouveau nom et/ou la nouvelle date et/ou la nouvelle heure de la séance.  | Le système vérifie qu'il n'existe pas déjà une séance enregistrée à la date et à l'heure introduites.<br>Le système enregistre les nouvelles informations.<br>Le système indique que la séance a bien été modifiée. |

## Scénario alternatif : L'heure, la date nouvellement introduite(s) de la séance correspondent déjà à une autre séance enregistrée.

| <b>Précondition</b> : Aucune siège n'a déjà été vendue pour la séance à modifier. Les nouvelles informations (heure et/ou date) correspondent à une séance déjà enregistrée. |   |
|--|---|
| <b>Postcondition</b> : Les informations de la séance ne sont pas modifiées.  |   |
| Programmeur  | Système   |
| Le programmeur introduit le nom, la date et l'heure de la séance qu'il souhaite modifier.  | Le système vérifie qu'il existe bien une séance enregistrée au nom, à la date et à l'heure introduites.<br>Le système vérifie qu'il n'y a aucun siège déjà réservé pour la séance enregistrée au nom, à la date et à l'heure introduites.   |
| L'employé au guichet introduit le nouveau nom et/ou la nouvelle date et/ou la nouvelle heure de la séance en question.   | Le système vérifie qu'il n'existe pas déjà une séance enregistrée à la date et à l'heure nouvellement introduites.<br>Le système indique qu'il existe déjà une séance enregistrée à la date et à l'heure nouvellement introduites.<br>Le système indique le nom de la séance déjà enregistrée à la date et à l'heure introduites. |

## Scénario alternatif : Au moins un siège a été réservé pour la séance à modifier

| <b>Précondition</b> : Une ou plusieurs sièges ont déjà été réservés pour la séance encore non modifiée. |  |
|---|--|
| <b>Postcondition</b> : Les informations de la séance ne sont pas mises à jour.                          |  |
| Programmeur   | Système  |
| Le programmeur introduit le nom, la date et l'heure de la séance qu'il souhaite modifier.               | Le système vérifie qu'il existe bien une séance enregistrée au nom, à la date et à l'heure introduits.<br>Le système vérifie qu'il n'y a aucun siège déjà réservé pour la séance enregistrée au nom, à la date et à l'heure introduits.<br>Le système indique qu'une ou plusieurs sièges ont déjà été vendue(s). |

## Scénario alternatif : La séance à modifier n'existe pas

| <b>Précondition</b> : La séance à modifier n'existe pas                                   |   |
|---|---|
| <b>Postcondition</b> : Les informations de la séance ne sont pas mises à jour.            |   |
| Programmeur   | Système   |
| Le programmeur introduit le nom, la date et l'heure de la séance qu'il souhaite modifier. | Le système vérifie qu'il existe bien une séance enregistrée au nom, à la date et à l'heure introduits.<br>Le système indique qu'il n'existe aucune séance enregistrée au nom, date et heure introduits. |

# Annuler une séance

## Scénario principal

| <b>Précondition</b> : La séance à annuler existe. Il n'y a aucune réservation d'enregistrée pour la séance à annuler. |   |
|---|---|
| <b>Postcondition</b> : La séance est supprimée.   |   |
| Programmeur   | Système   |
| Le programmeur introduit la date et l'heure de la séance qu'il souhaite annuler.                                      | Le système vérifie qu'il existe bien une séance enregistrée à la date et à l'heure introduits.<br>Le système vérifie qu'il n'y a aucun siège déjà réservé pour la séance enregistrée au nom, à la date et à l'heure introduits.<br>Le système supprime la séance.<br>Le système indique que la séance a bien été supprimée. |

## Scénario alternatif : Au moins un siège a été réservée pour la séance à annuler

| <p><b>Précondition</b> : La séance à annuler existe. Il y a au moins une réservation enregistrée pour la séance à annuler.</p> <p><b>Postcondition</b> : La séance est supprimée. Un message automatique d'annulation est envoyé aux clients ayant réalisé une réservation.</p> |  |
|---|--|
| Programmeur   | Système  |
| Le programmeur introduit la date et l'heure de la séance qu'il souhaite annuler.  | Le système vérifie qu'il existe bien une séance enregistrée à la date et à l'heure introduites.<br>Le système vérifie qu'il n'y a aucun siège déjà réservé pour la séance enregistrée au nom, à la date et à l'heure introduits.   |
|   | Le système indique que des réservations ont déjà été effectuées.<br>Le système envoie un message d'annulation de la séance à l'adresse électronique des clients ayant effectué une réservation.<br>Le système indique qu'une notification d'annulation de séances a été envoyée aux clients ayant réservées des sièges pour la séance annulée.<br>Le système supprime la séance.<br>Le système indique que la séance a bien été supprimée. |

## Scénario alternatif : La séance à annuler n'existe pas

| <p><b>Précondition</b> : La séance à annuler n'existe pas.</p> <p><b>Postcondition</b> : Aucune séance n'est supprimée.</p> |  |
|---|--|
| Programmeur   | Système  |
| Le programmeur introduit la date et l'heure de la séance qu'il souhaite annuler.  | Le système vérifie qu'il existe bien une séance enregistrée à la date et à l'heure introduites.<br>Le système indique qu'il n'y a aucune séance enregistrée au nom, à la date et à l'heure introduits. |

# Annuler une séance

## Scénario principal

| <b>Précondition</b> : Le client possède un compte client. La séance n'est pas sold-out.    |  |  |
|--|--|--|
| <b>Postcondition</b> : La réservation est enregistrée.                                     |  |  |
| Client   | Guichet  | Système  |
| Le client indique son nom, prénom et son numéro de téléphone.                              | L'employé au guichet introduit le nom, prénom et numéro de téléphone.  | Le système vérifie qu'il existe bien un compte client enregistré au nom, prénom et numéro de téléphone introduits.   |
| Le client indique le nom, la date et l'heure de la séance à laquelle il souhaite assister. | L'employé au guichet introduit le nom, la date et l'heure de la séance à laquelle le client souhaite assister.   | Le système vérifie qu'il existe bien une séance enregistrée au nom, à la date et à l'heure introduits.<br>Le système vérifie qu'il reste encore des sièges disponibles.<br>Le système indique les sièges encore disponibles. |
| Le client indique le ou les références(s) des sièges qui l'intéresse.                      | L'employé au guichet sélectionne la ou les références(s) des sièges qui intéresse le client parmi les sièges disponibles restantes affichées par le système. | Le système attribue un identifiant unique à la réservation.<br>Le système enregistre la réservation.<br>Le système indique que la réservation est bien enregistrée.  |

## Scénario alternatif : Les places disponibles ne satisfont pas le client

| <b>Précondition</b> : Le client possède un compte client. La séance n'est pas sold-out.    |  |  |
|--|--|--|
| <b>Postcondition</b> : La requête de réservation est annulée.                              |  |  |
| Client   | Guichet  | Système  |
| Le client indique son nom, prénom et son numéro de téléphone.                              | L'employé au guichet introduit le nom, prénom et numéro de téléphone.  | Le système vérifie qu'il existe bien un compte client enregistré au nom, prénom et numéro de téléphone introduits.   |
| Le client indique le nom, la date et l'heure de la séance à laquelle il souhaite assister. | L'employé au guichet introduit le nom, la date et l'heure de la séance à laquelle le client souhaite assister. | Le système vérifie qu'il existe bien une séance enregistrée au nom, à la date et à l'heure introduits.<br>Le système vérifie qu'il reste encore des sièges disponibles.<br>Le système indique les sièges encore disponibles. |
| Le client indique que les sièges restants ne lui conviennent pas.                          | L'employé annule la réservation.   | Le système annule la requête.  |



### Scénario alternatif : Le client n'a pas de compte client

| <b>Précondition</b> : Le client ne possède pas déjà un compte client enregistré. |   |  |
|--|---|--|
| <b>Postcondition</b> : Aucune réservation n'est enregistrée                      |   |  |
| Client   | Guichet   | Systeme  |
| Le client indique son nom, prénom et son numéro de téléphone.                    | L'employé au guichet introduit le nom, prénom et numéro de téléphone. | Le système vérifie qu'il existe bien un compte client enregistré au nom, prénom et numéro de téléphone introduits.<br>Le système indique qu'aucun compte client n'est enregistré au nom, prénom et numéro de téléphone introduits. |

### Scénario alternatif : La séance est sold-out.

| <b>Précondition</b> : Le client possède déjà un compte client enregistré. La séance dont la réservation fait l'objet est sold-out. |   |   |
|--|---|---|
| <b>Postcondition</b> : Aucune réservation n'est enregistrée.   |   |   |
| Client   | Guichet   | Systeme   |
| Le client indique son nom, prénom et son numéro de téléphone.  | L'employé au guichet introduit le nom, prénom et numéro de téléphone.   | Le système vérifie qu'il existe bien un compte client enregistré au nom, prénom et numéro de téléphone introduits.  |
| Le client indique le nom, la date et l'heure de la séance à laquelle il souhaite assister.   | L'employé au guichet introduit le le nom, la date et l'heure de la séance à laquelle le client souhaite assister. | Le système vérifie qu'il existe bien une séance enregistrée au nom, à la date et à l'heure introduits.<br>Le système vérifie qu'il reste encore des sièges disponibles.<br>Le système indique que la séance est sold-out. |

### Scénario alternatif : La séance n'existe pas.

| <b>Précondition</b> : Le client possède déjà un compte client enregistré. La séance dont la réservation fait l'objet n'existe pas. |  |  |
|--|--|--|
| <b>Postcondition</b> : Aucune réservation n'est enregistrée.   |  |  |
| Client   | Guichet  | Système  |
| Le client indique son nom, prénom et son numéro de téléphone.  | L'employé au guichet introduit le nom, prénom et numéro de téléphone.  | Le système vérifie qu'il existe bien un compte client enregistré au nom, prénom et numéro de téléphone introduits.   |
| Le client indique le nom, la date et l'heure de la séance à laquelle il souhaite assister.   | L'employé au guichet introduit le nom, la date et l'heure de la séance à laquelle le client souhaite assister. | Le système vérifie qu'il existe bien une séance enregistrée au nom, à la date et à l'heure introduits.<br>Le système indique qu'il n'existe pas de séance enregistrée au nom, à la date et à l'heure introduits. |

## Rapport des sièges réservés pour une séance

### Scénario principal

| <b>Précondition</b> : La séance existe.                        |   |
|--|---|
| <b>Postcondition</b> : Le système affiche les sièges réservés. |   |
| Guichet  | Système   |
| L'employé au guichet introduit la date et l'heure de la séance | Le système vérifie qu'il existe bien une séance enregistrée au nom, date et heure introduits.<br>Le système affiche les références des sièges réservés, le nom et prénom des clients qui ont réservé les sièges pour la séance en question ainsi que le nom de la production. |

### Scénario alternatif : La séance n'existe pas.

| <b>Précondition</b> : La séance existe.                        |  |
|--|--|
| <b>Postcondition</b> : Le système affiche les sièges réservés. |  |
| Guichet  | Système  |
| L'employé au guichet introduit la date et l'heure de la séance | Le système vérifie qu'il existe bien une séance enregistrée au nom, date et heure introduits.<br>Le système indique qu'il n'existe pas de séance enregistrée au nom, date et heure introduits. |

## Scénario alternatif : Aucune réservation

| <b>Précondition</b> : La séance existe. Il n'y a pas de réservations. |   |
|---|---|
| <b>Postcondition</b> : Le système affiche les sièges réservés.        |   |
| Guichet   | Système   |
| L'employé au guichet introduit la date et l'heure de la séance        | Le système vérifie qu'il existe bien une séance enregistrée à la date et heure introduites.<br>Le système indique qu'il n'y a aucune réservation enregistrée pour cette séance. |

# Rapport des sièges réservés pour une séance pour un client

## Scénario principal

| <b>Précondition</b> : La séance existe. Le client existe.  |  |
|--|--|
| <b>Postcondition</b> : Le système affiche les sièges réservés.   |  |
| Guichet  | Système  |
| L'employé au guichet introduit la date, l'heure de la séance, le nom, prénom et numéro de téléphone du client. | Le système vérifie qu'il existe bien une séance enregistrée au nom, date et heure introduits.<br>Le système vérifie qu'il existe bien un client enregistré au nom, prénom et numéro de téléphone introduits.       |
|  | Le système vérifie que le client à une réservation enregistrée pour la séance.<br>Le système affiche les références des sièges réservés pour le client et la séance en question ainsi que le nom de la production. |

## Scénario alternatif : La séance n'existe pas.

| <b>Précondition</b> : La séance n'existe pas.  |  |
|--|--|
| <b>Postcondition</b> : Aucun rapport n'est généré.   |  |
| Guichet  | Système  |
| L'employé au guichet introduit la date, l'heure de la séance, le nom, prénom et numéro de téléphone du client. | Le système vérifie qu'il existe bien une séance enregistrée au nom, date et heure introduits.<br>Le système indique qu'il n'existe pas de séance enregistrée au nom, date et heure introduits. |

### Scénario alternatif : Le client n'existe pas.

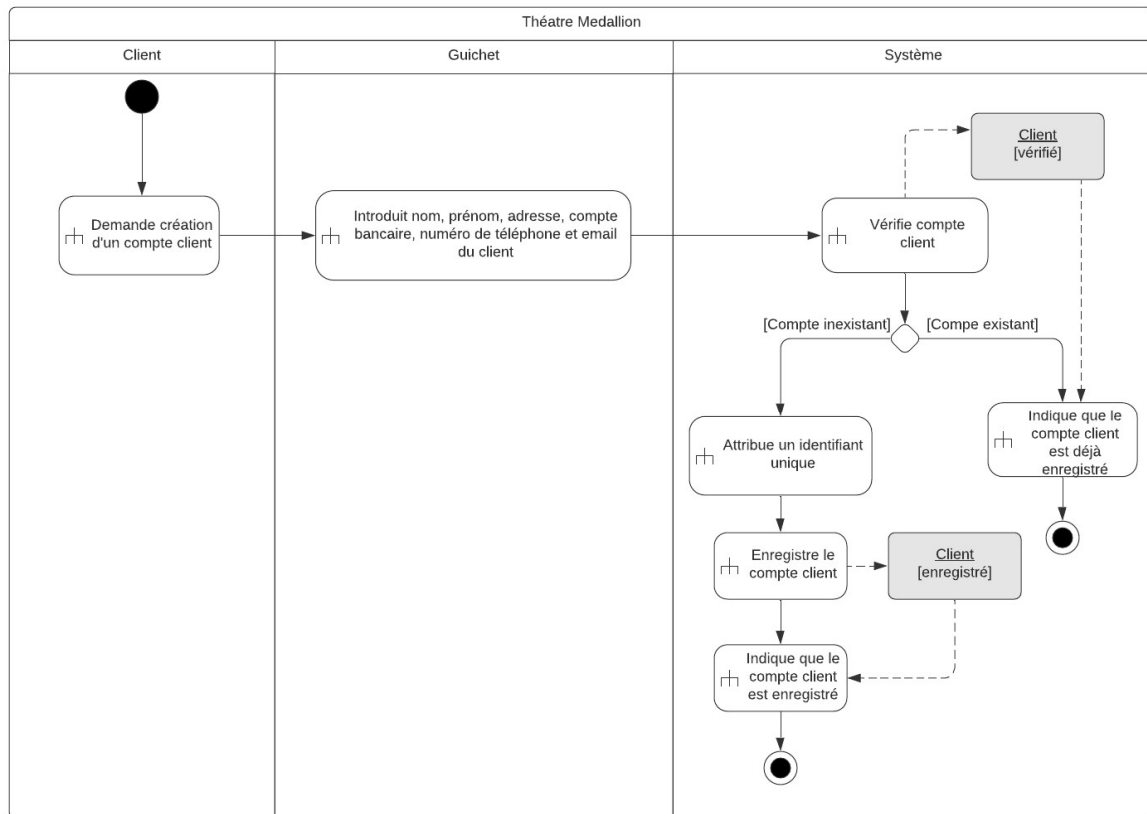
| <b>Précondition</b> : Le client n'existe pas.  |  |
|--|--|
| <b>Postcondition</b> : Aucun rapport n'est généré.   |  |
| Guichet  | Système  |
| L'employé au guichet introduit la date, l'heure de la séance, le nom, prénom et numéro de téléphone du client. | Le système vérifie qu'il existe bien une séance enregistrée au nom, date et heure introduits.<br>Le système vérifie qu'il existe bien un client enregistré au nom, prénom et numéro de téléphone introduits.<br>Le système indique qu'il n'existe pas de client enregistré au nom, prénom et numéro de téléphone introduits. |

### Scénario alternatif : Aucune réservation

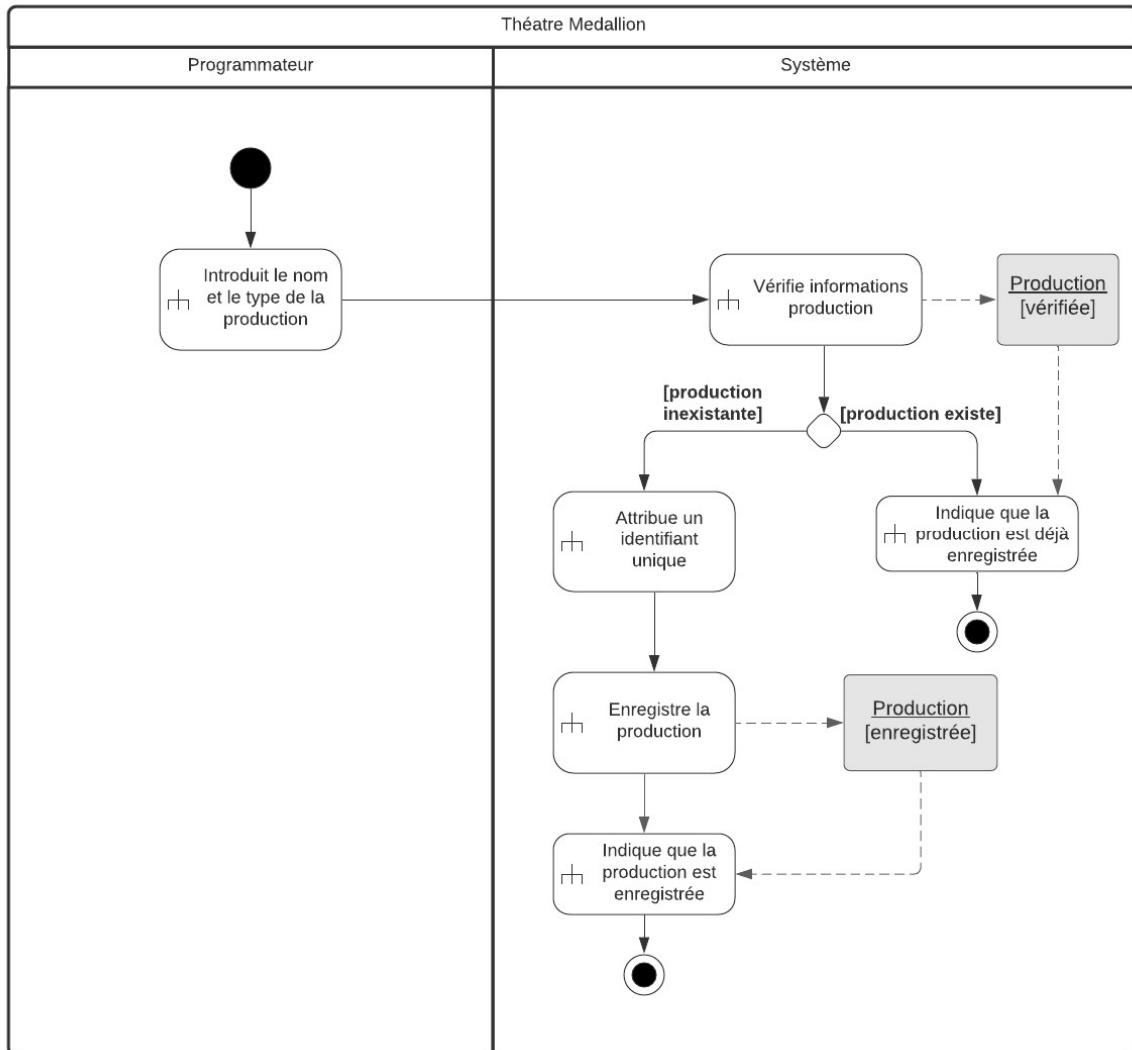
| <b>Précondition</b> : Aucune réservation réalisée pour le client.  |  |
|--|--|
| <b>Postcondition</b> : Aucun rapport n'est généré.   |  |
| Guichet  | Système  |
| L'employé au guichet introduit la date, l'heure de la séance, le nom, prénom et numéro de téléphone du client. | Le système vérifie qu'il existe bien une séance enregistrée au nom, date et heure introduits.<br>Le système vérifie qu'il existe bien un client enregistré au nom, prénom et numéro de téléphone introduits. |
|  | Le système vérifie que le client à une réservation enregistrée pour la séance.<br>Le système indique que le client n'a aucune réservation pour cette séance.   |

# Appendice 3 – Diagrammes d’activités

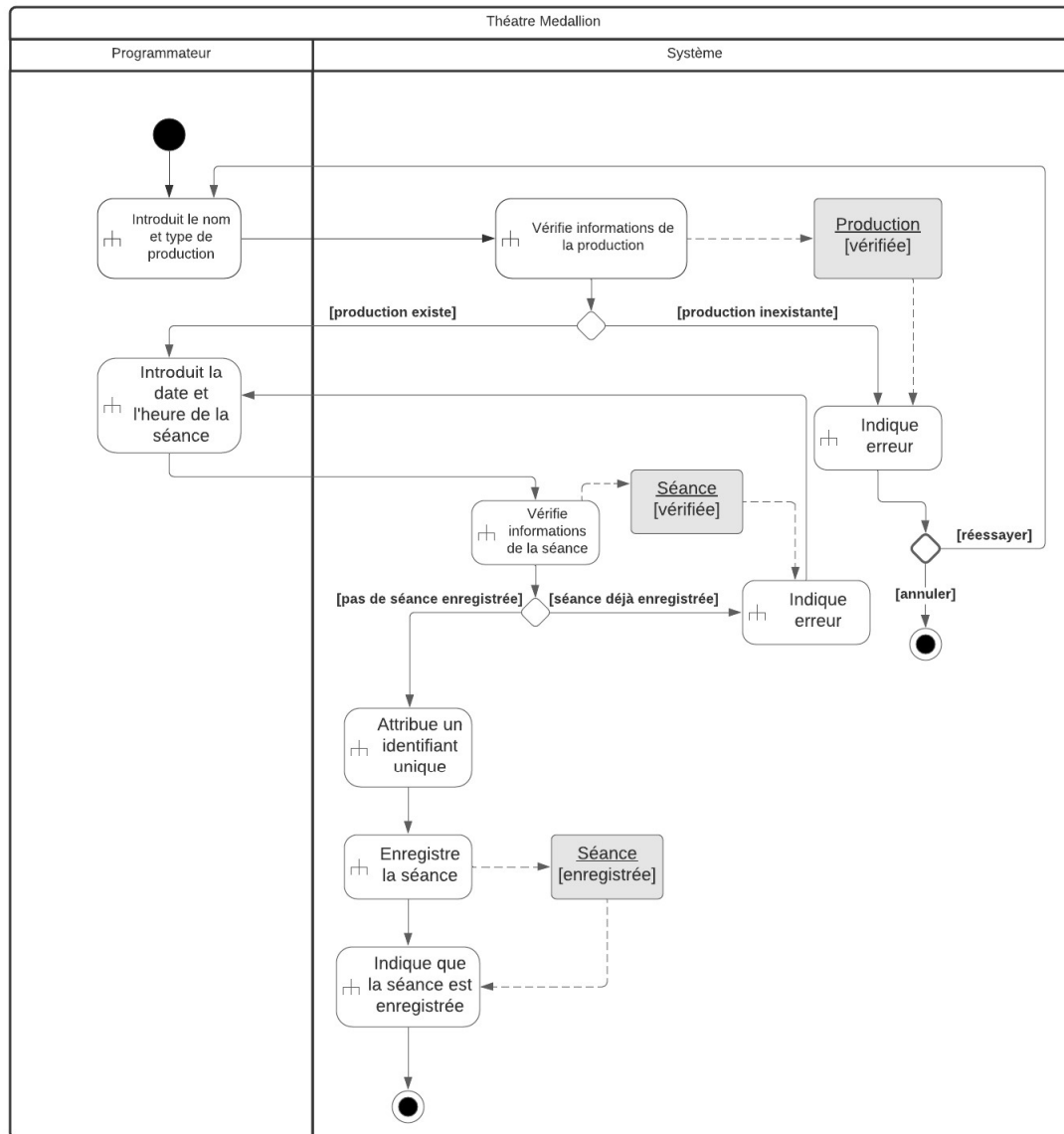
## Ajouter un compte client



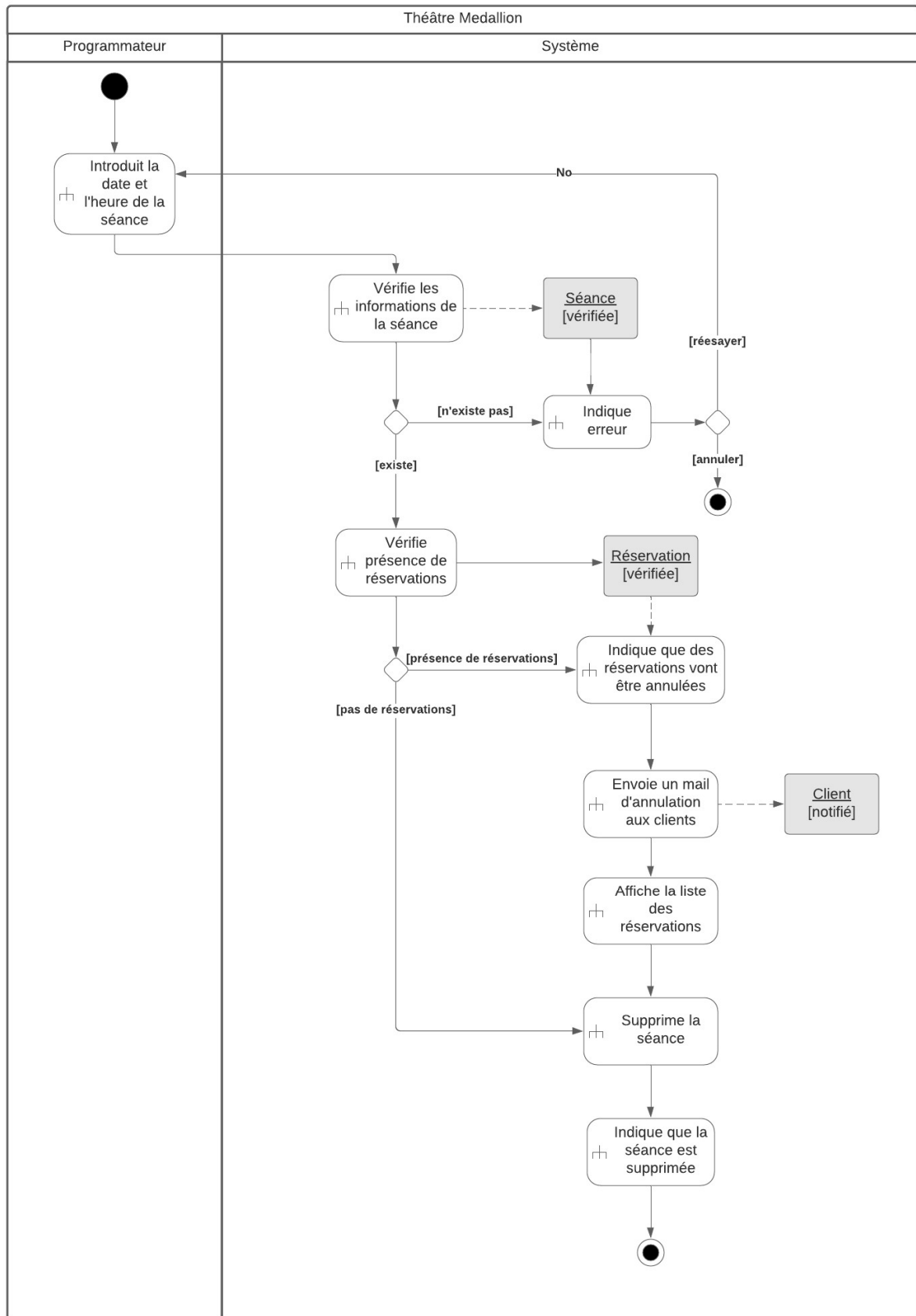
# Ajouter une production



# Ajouter une séance

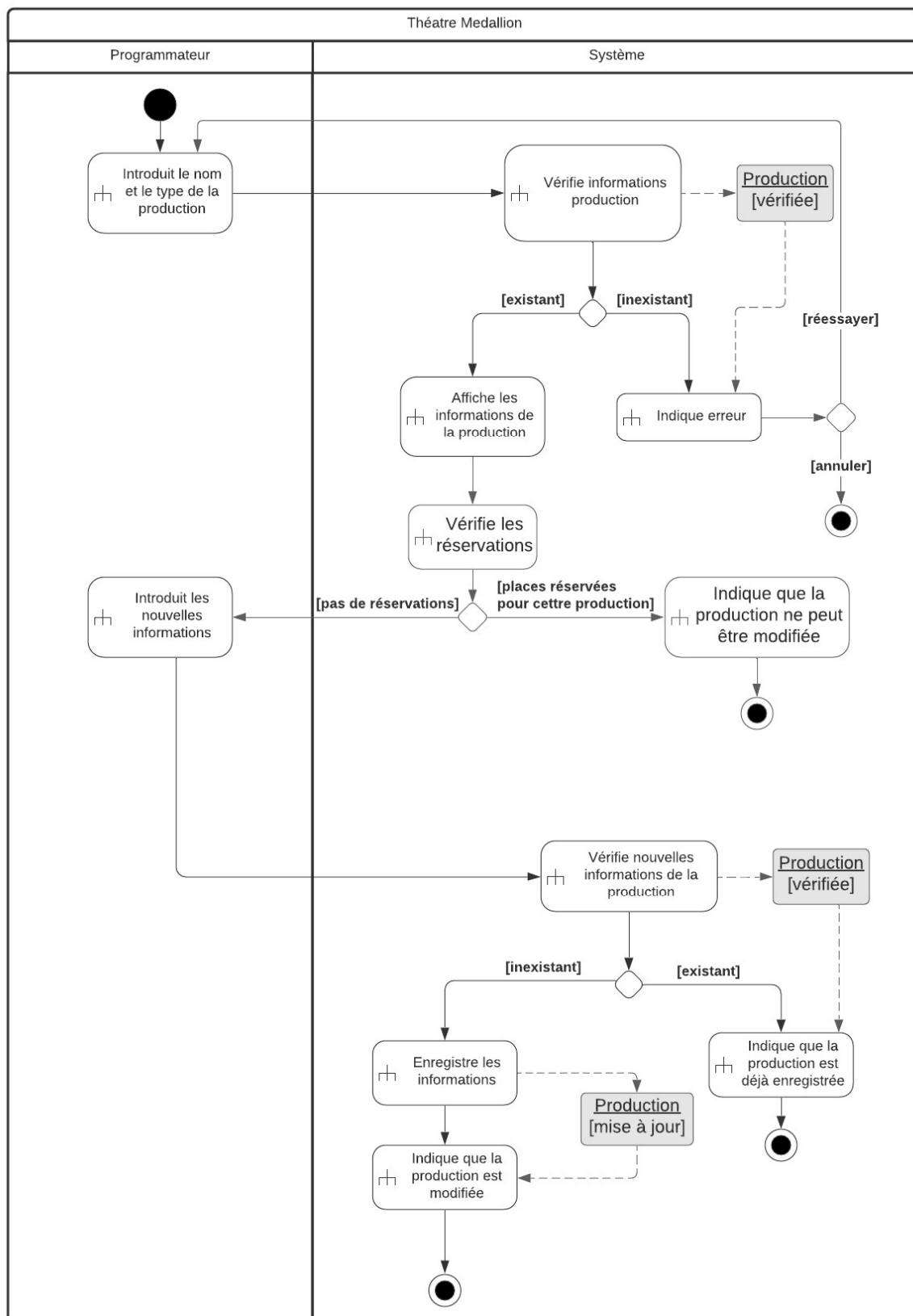


# Annuler une séance

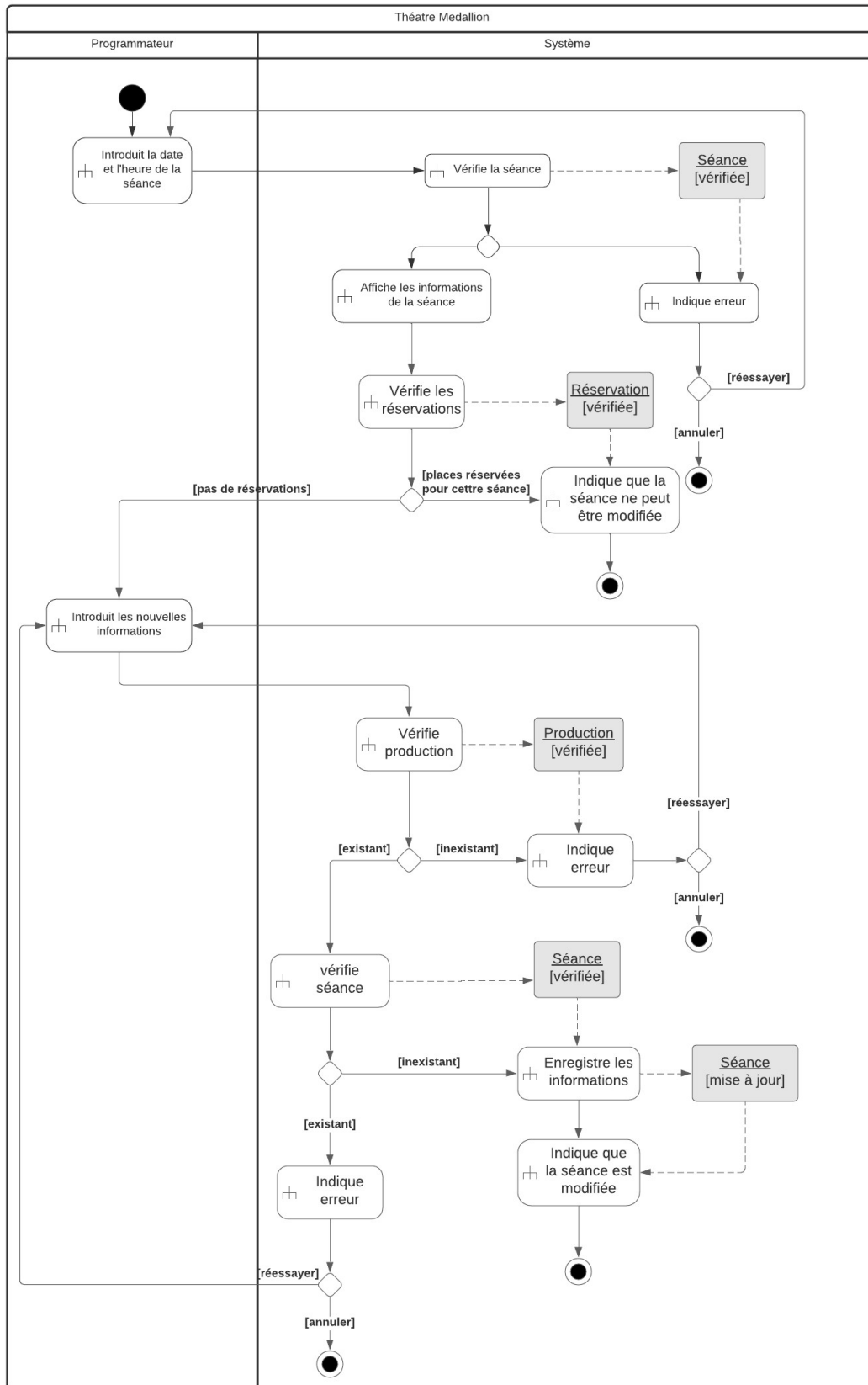




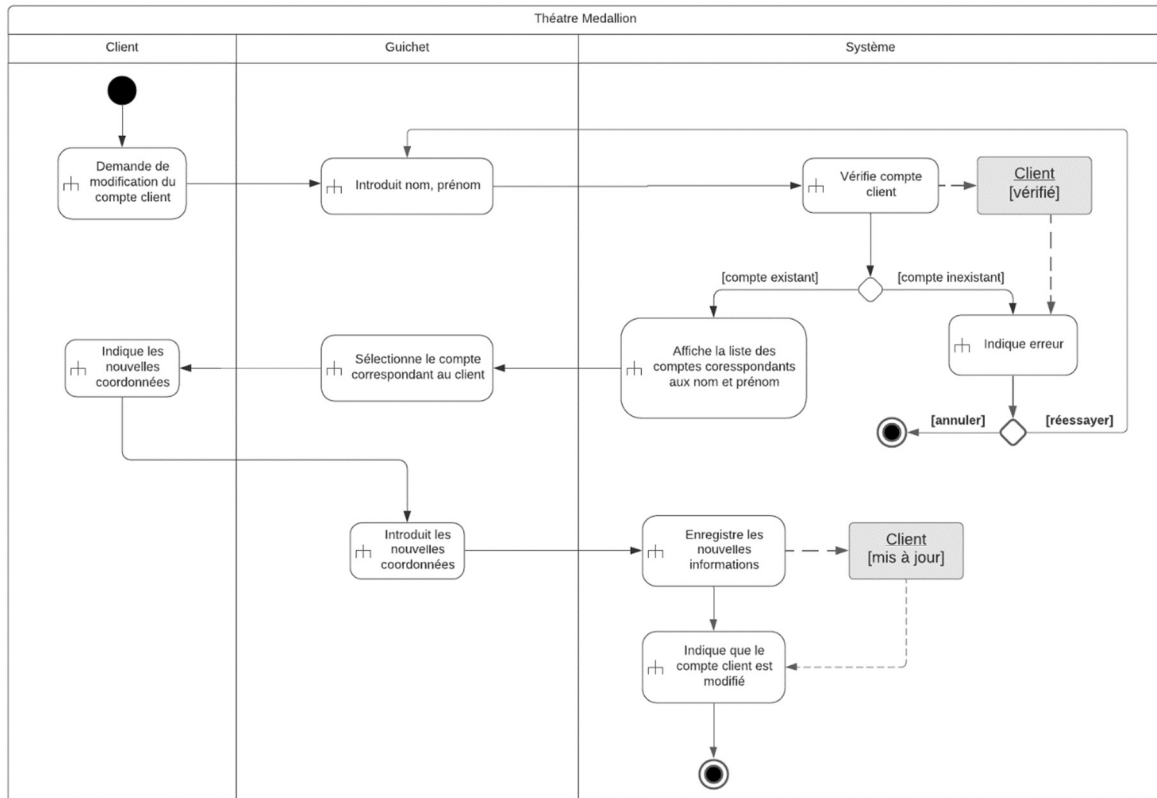
# Modifier une production



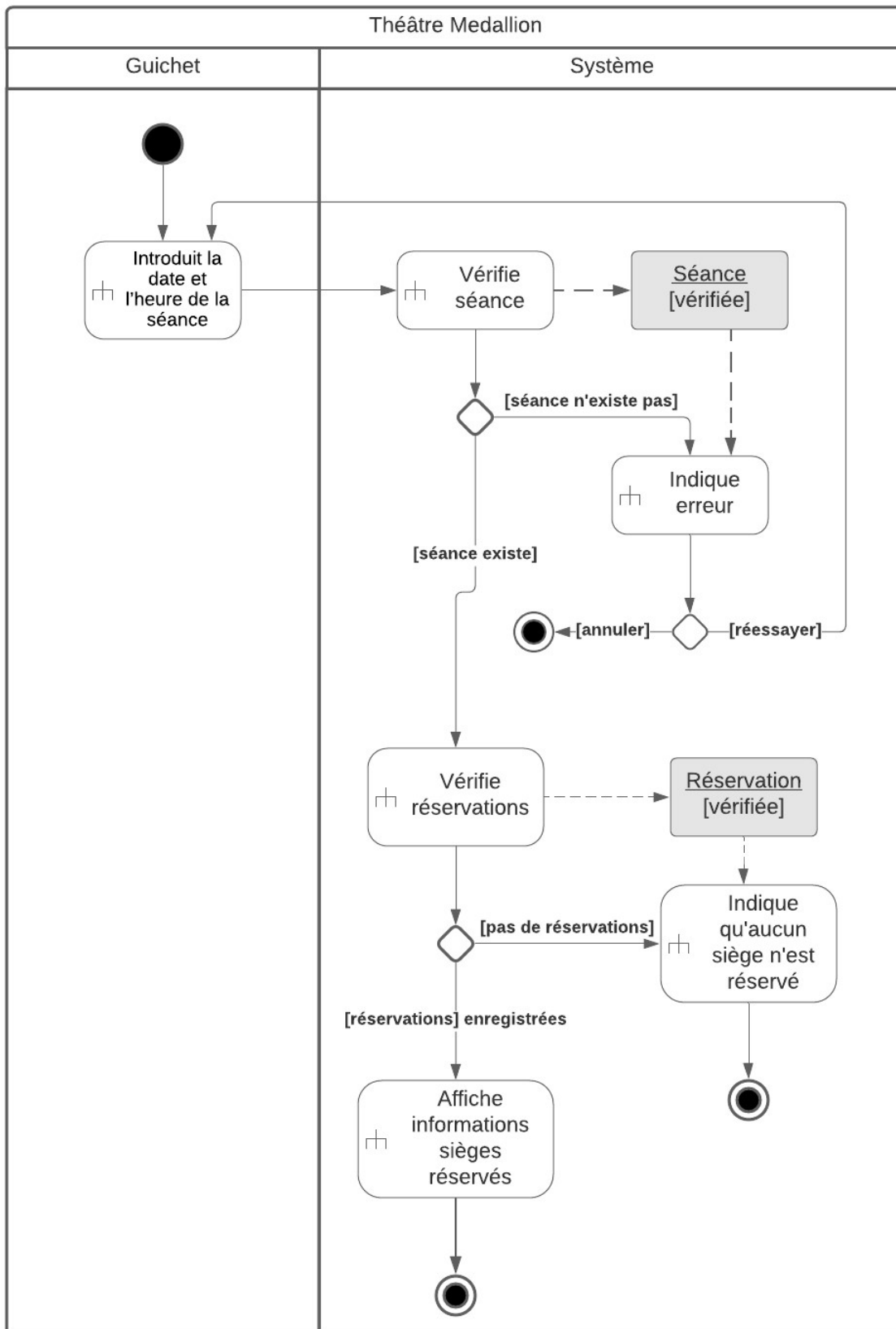
# Modifier une séance



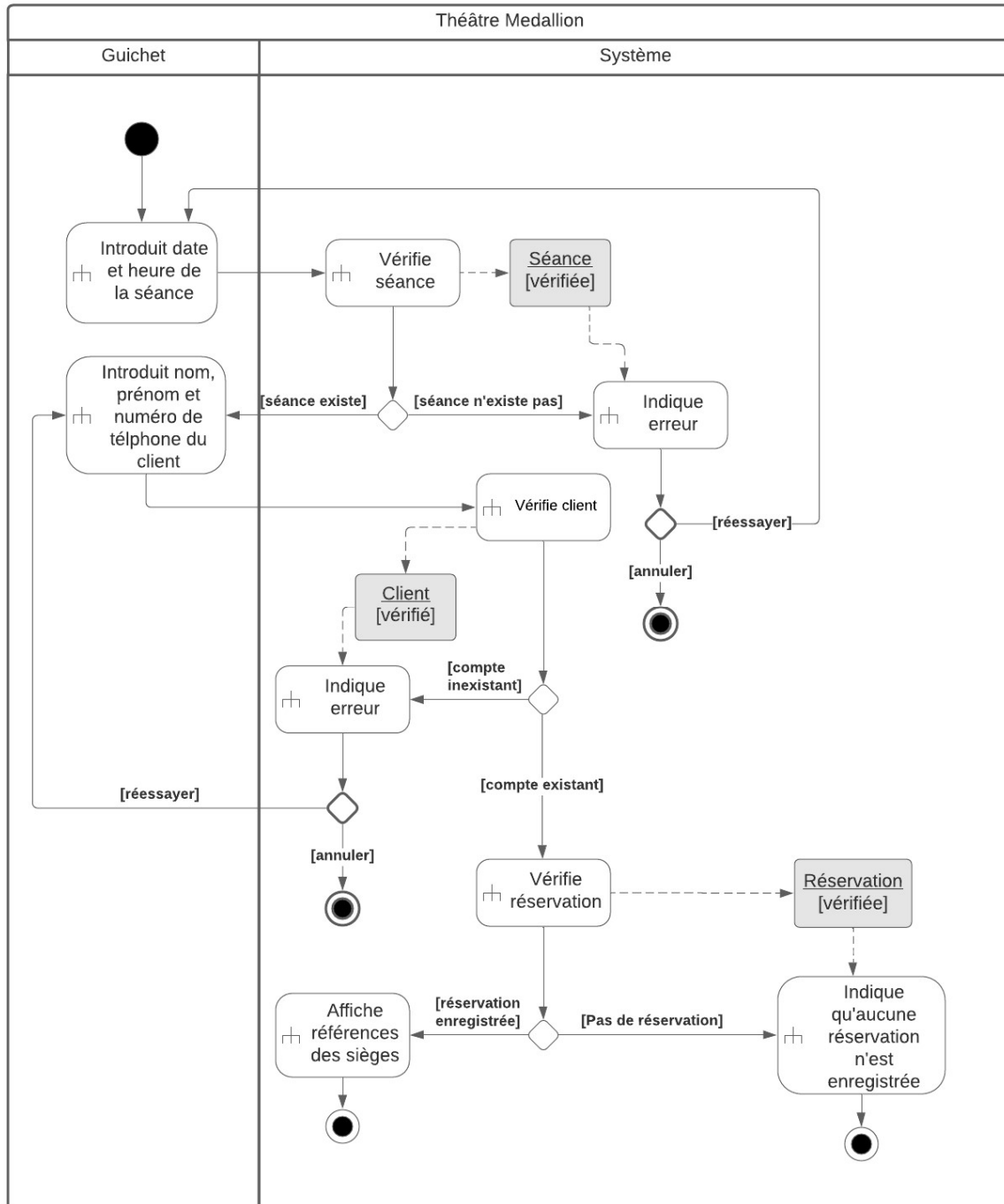
# Modifier un compte client



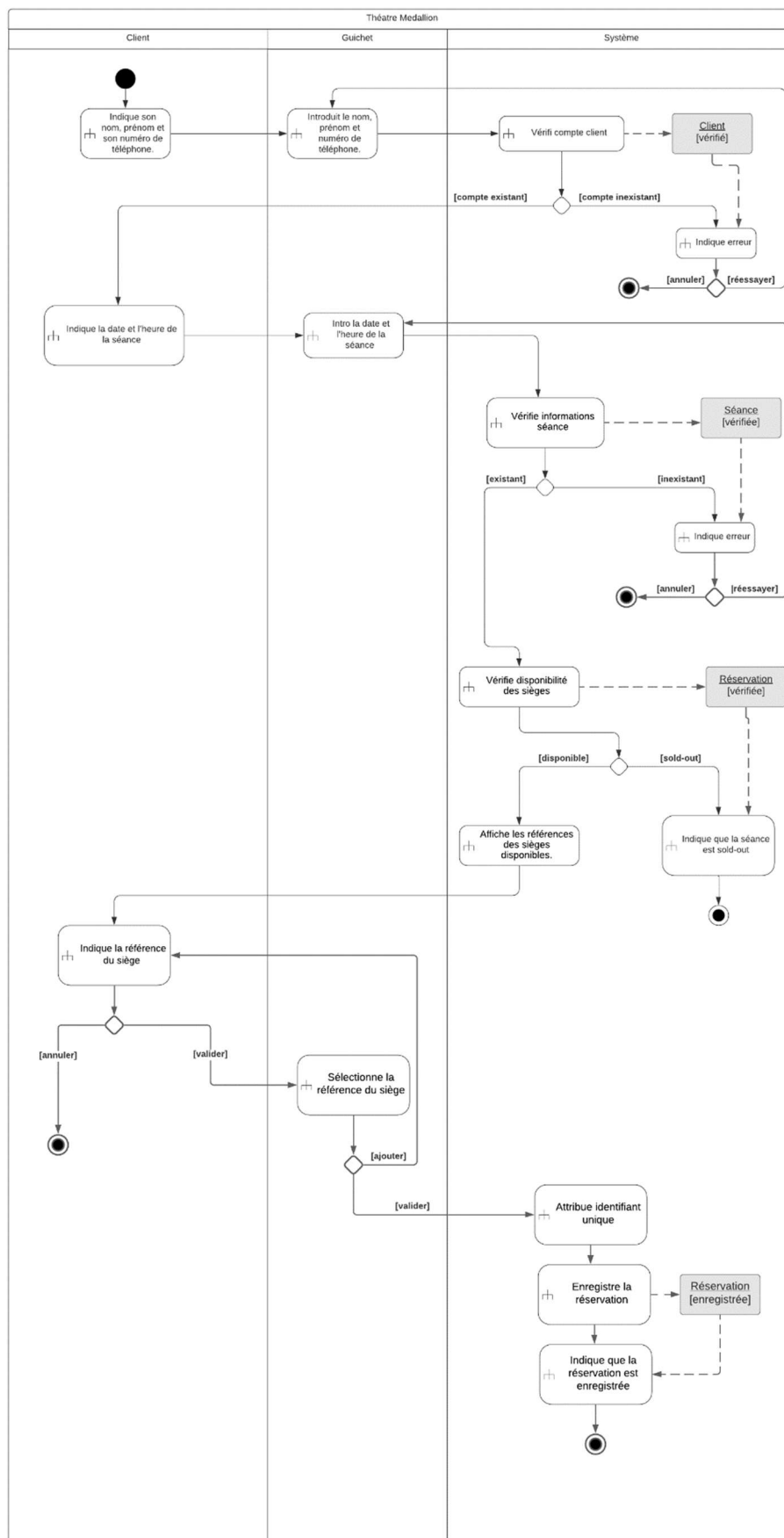
# Rapports des sièges réservés pour une séance



# Rapports des sièges réservés pour une séance pour client

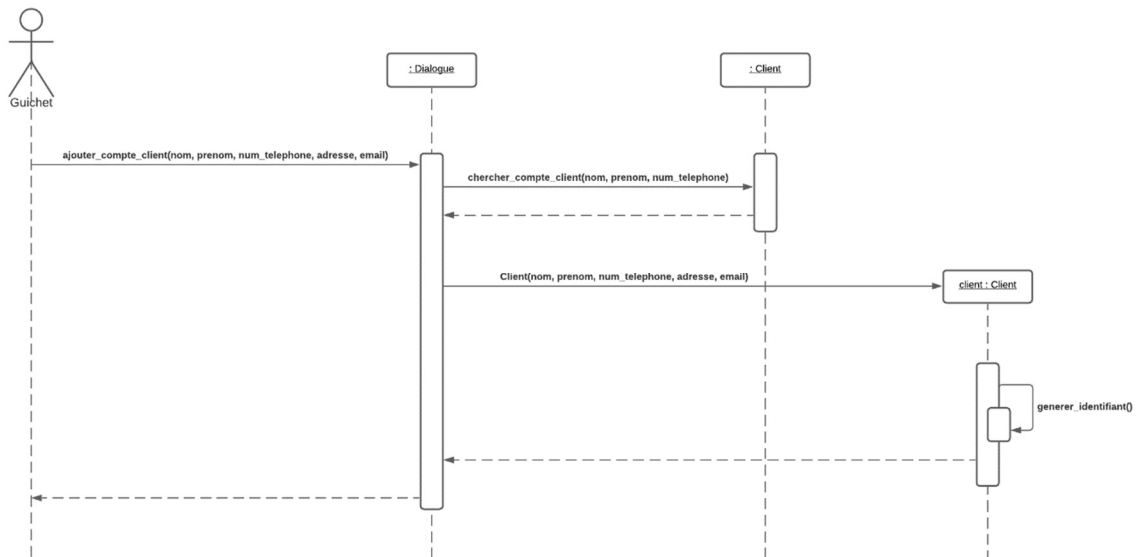


# Réserver

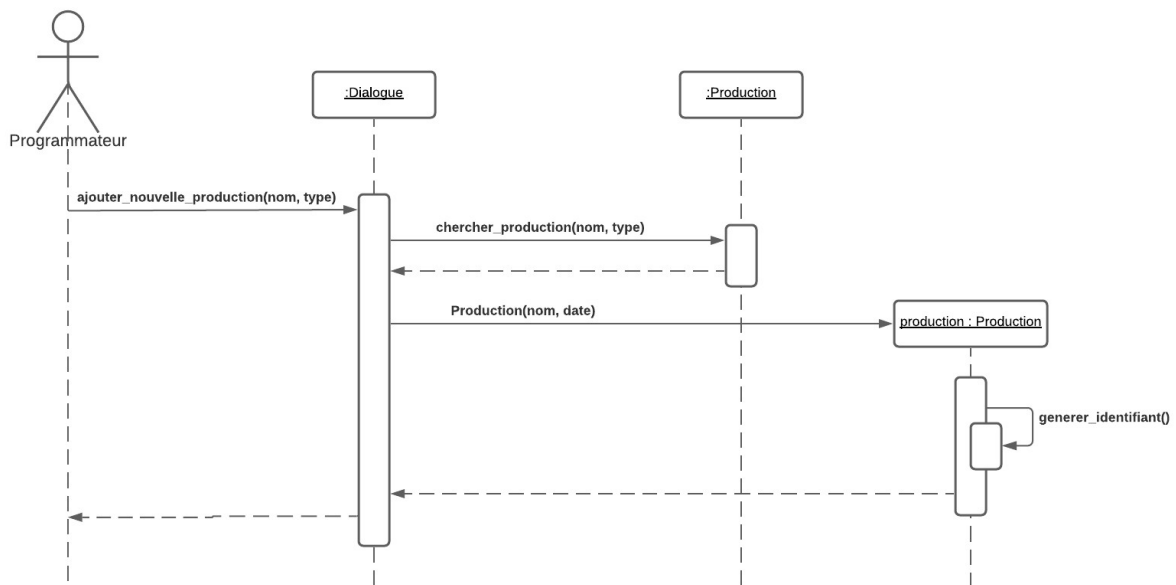


# Appendice 4 - Diagrammes de séquences fonctionnels

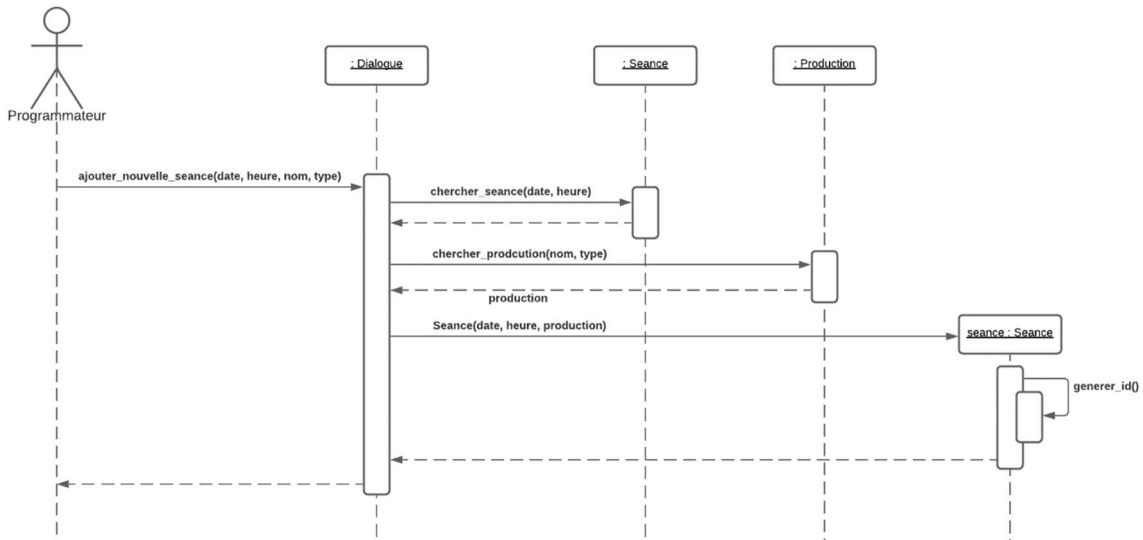
## Ajouter un compte client



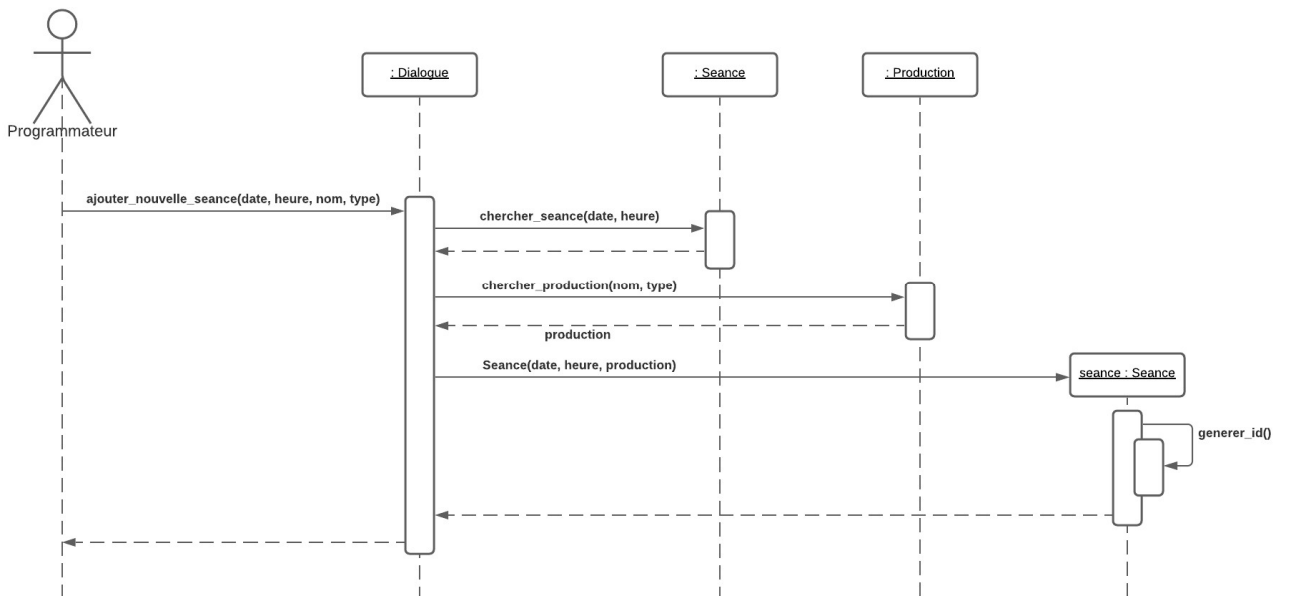
## Ajouter une production



# Ajouter une séance

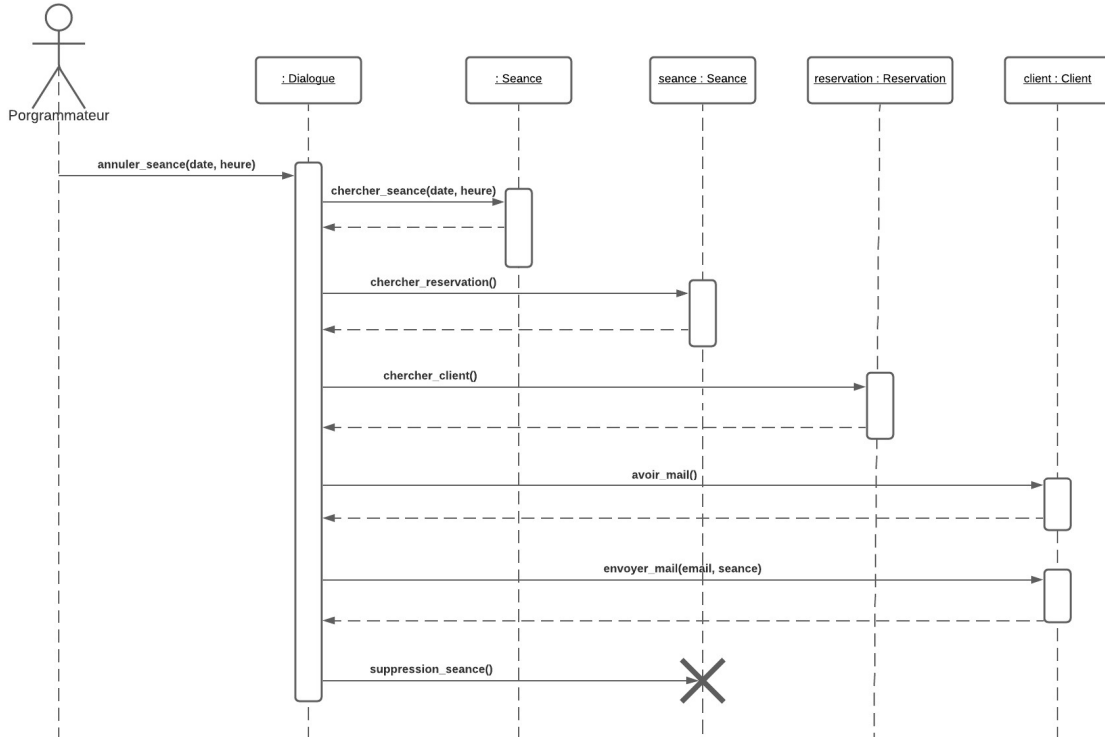


# Annuler une séance (principal)

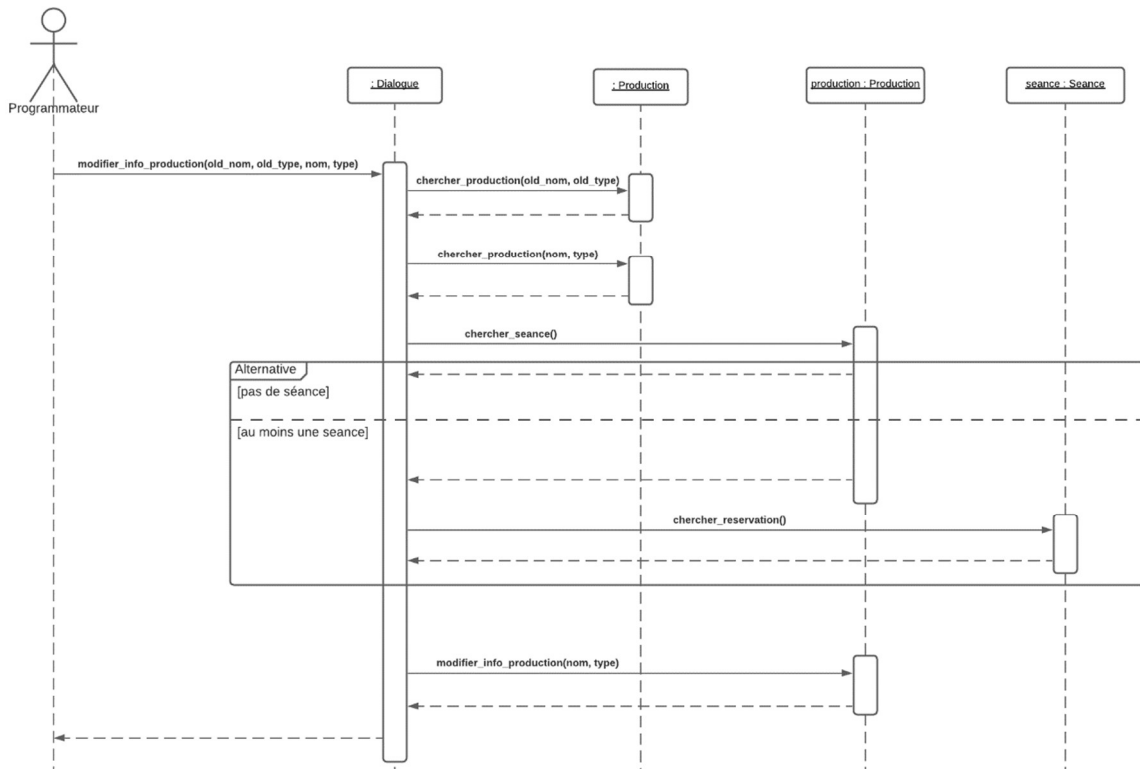




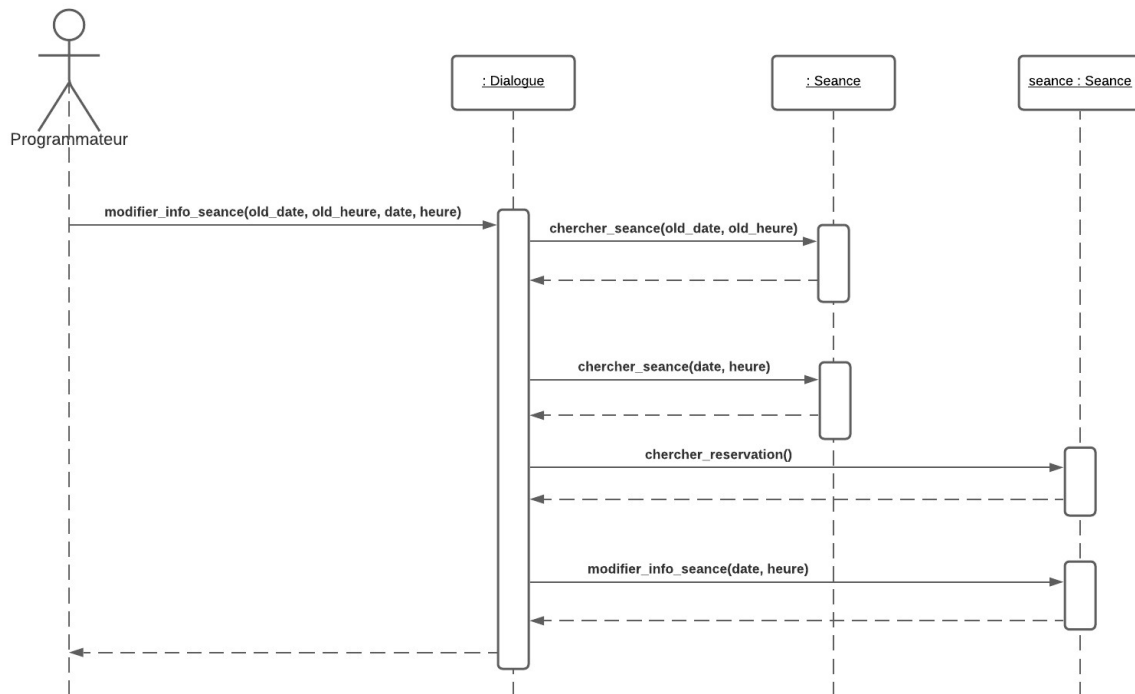
# Annuler une séance (alternatif)



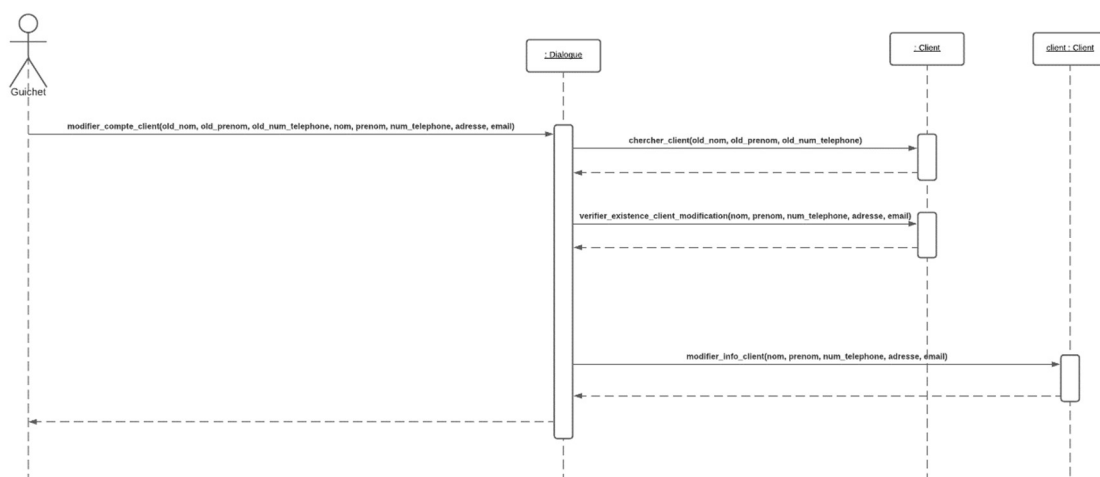
# Modifier une production



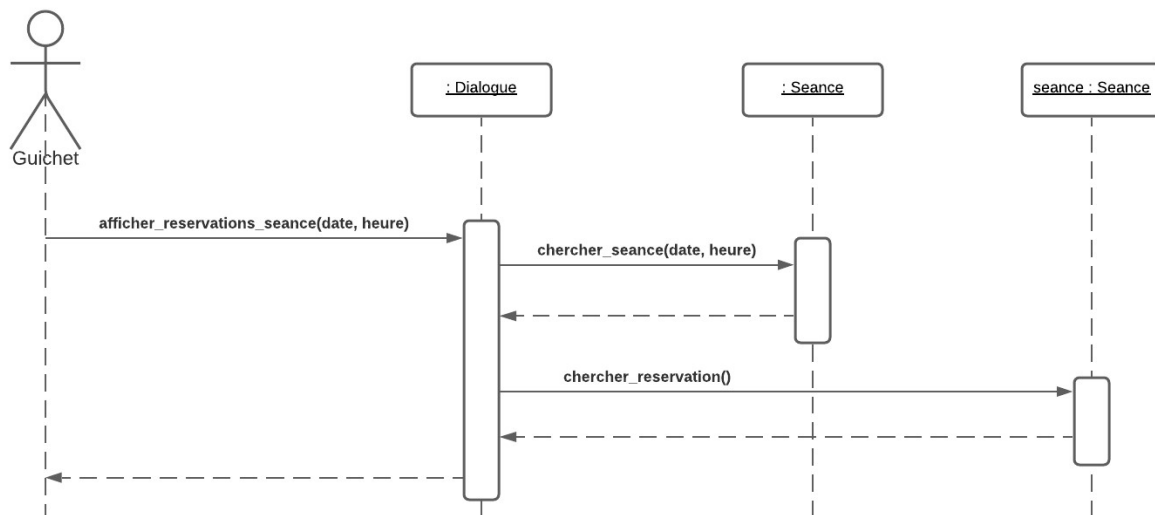
# Modifier une séance



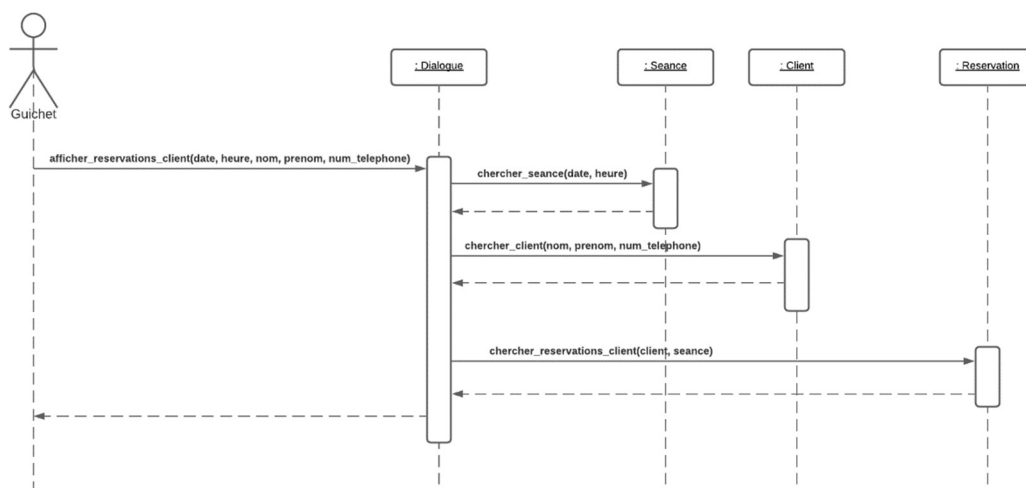
# Modifier un compte client



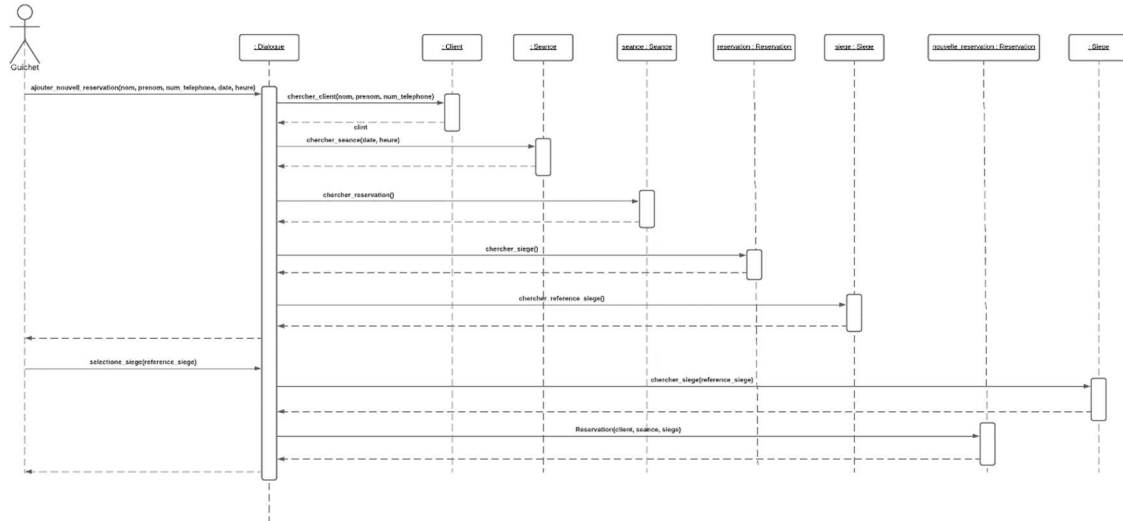
# Rapports des sièges réservés pour une séance



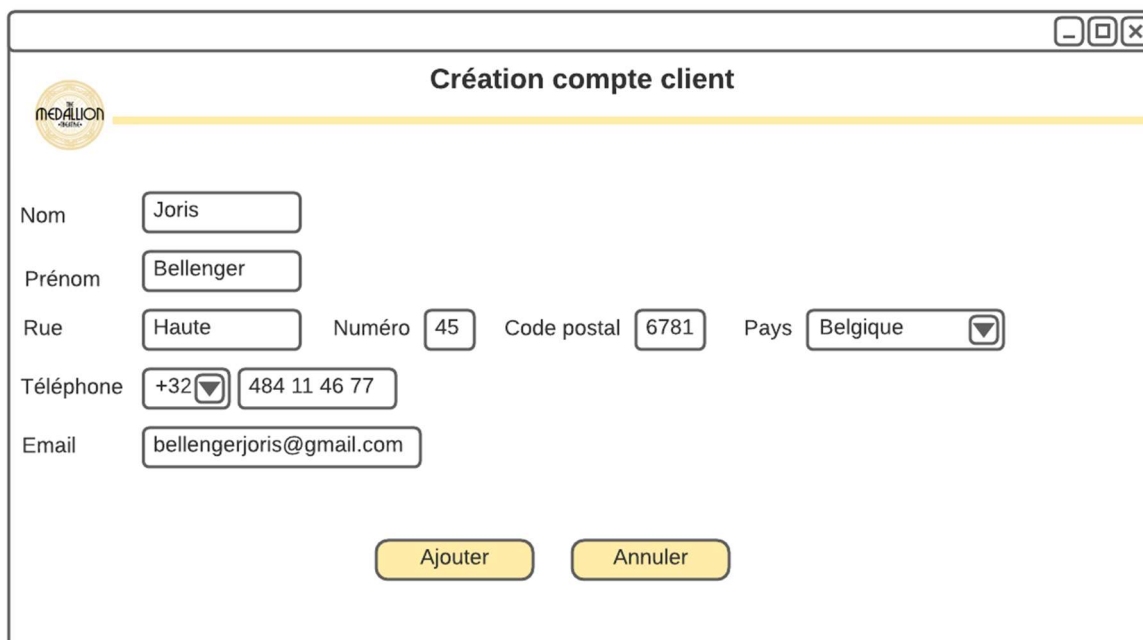
# Rapports des sièges réservés pour une séance pour client



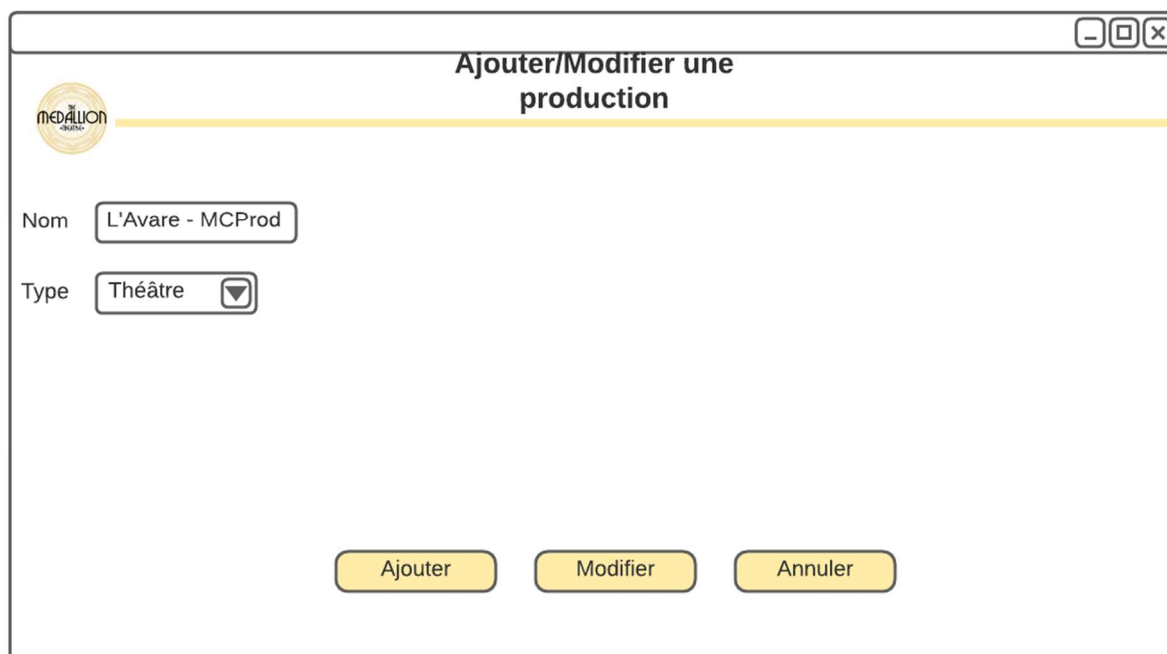
# Réserver




# Appendice 5 – Mock-up de l’interface utilisateur



The image shows a web browser window titled "Création compte client". The browser's address bar contains the URL "medallion.be". The page header features the "medallion.be" logo on the left and the title "Création compte client" in the center. Below the header, there is a form with the following fields: "Nom" (Joris), "Prénom" (Bellenger), "Rue" (Haute), "Numéro" (45), "Code postal" (6781), "Pays" (Belgique), "Téléphone" (+32 484 11 46 77), and "Email" (bellengerjoris@gmail.com). At the bottom of the form, there are two buttons: "Ajouter" and "Annuler".



The image shows a web browser window titled "Ajouter/Modifier une production". The browser's address bar contains the URL "medallion.be". The page header features the "medallion.be" logo on the left and the title "Ajouter/Modifier une production" in the center. Below the header, there is a form with the following fields: "Nom" (L'Avare - MCPProd) and "Type" (Théâtre). At the bottom of the form, there are three buttons: "Ajouter", "Modifier", and "Annuler".



## Ajouter une séance

---


Nom

Type

Date

|    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|
| S  | M  | T  | W  | Th | F  | S  |
|    | 1  | 2  | 3  | 4  | 5  | 6  |
| 7  | 8  | 9  | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 |    |    |    |

Heure



## Sièges réservés pour une séance


---

Date

Heure

Sièges réservés pour : *L'Avare - MCPProd*

| Nom       | prénom | Siège | Catégorie | Prix |
|-----------|--------|-------|-----------|------|
| Bellenger | Joris  | A1    | Orchestra | 65€  |
| Schmitz   | Laure  | K3    | Mezzanine | 55€  |
| Schmitz   | Laure  | K4    | Mezzanine | 55€  |




### Sièges réservés pour une séance pour un client

Date  
 Heure  
 Nom 
 Téléphone

Prénom

Sièges réservés pour : L'Avare - MCPProd

| Siège | Catégorie | Prix | Date de réservation |
|-------|-----------|------|---------------------|
| K3    | Mezzanine | 55€  | 15/12/2009          |
| K4    | Mezzanine | 55€  | 15/12/2009          |



### Ajouter une réservation

Date

| S  | M  | T  | W  | Th | F  | S  |
|----|----|----|----|----|----|----|
|    | 1  | 2  | 3  | 4  | 5  | 6  |
| 7  | 8  | 9  | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 |    |    |    |

Nom

Prénom

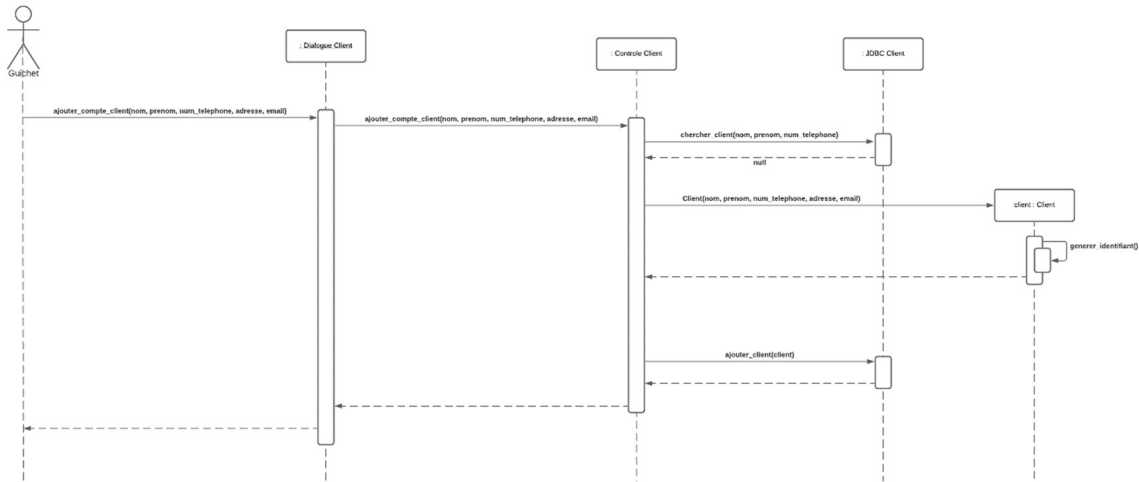
Téléphone

Heure

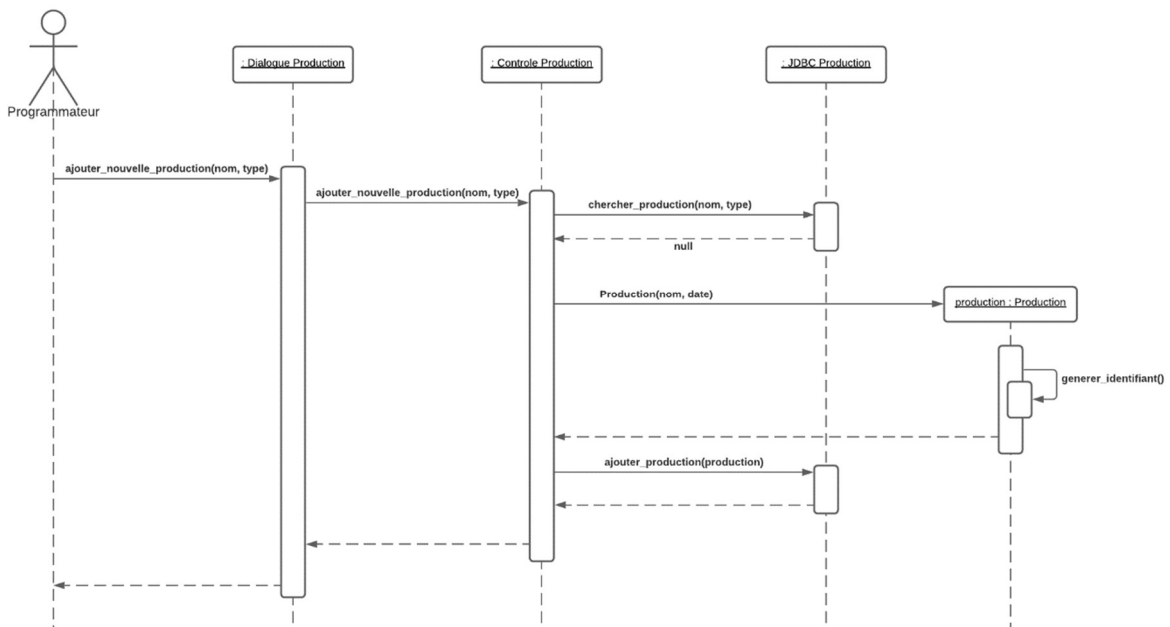
Siège

# Appendice 6 – Diagrammes de séquences de conception

## Ajouter un compte client

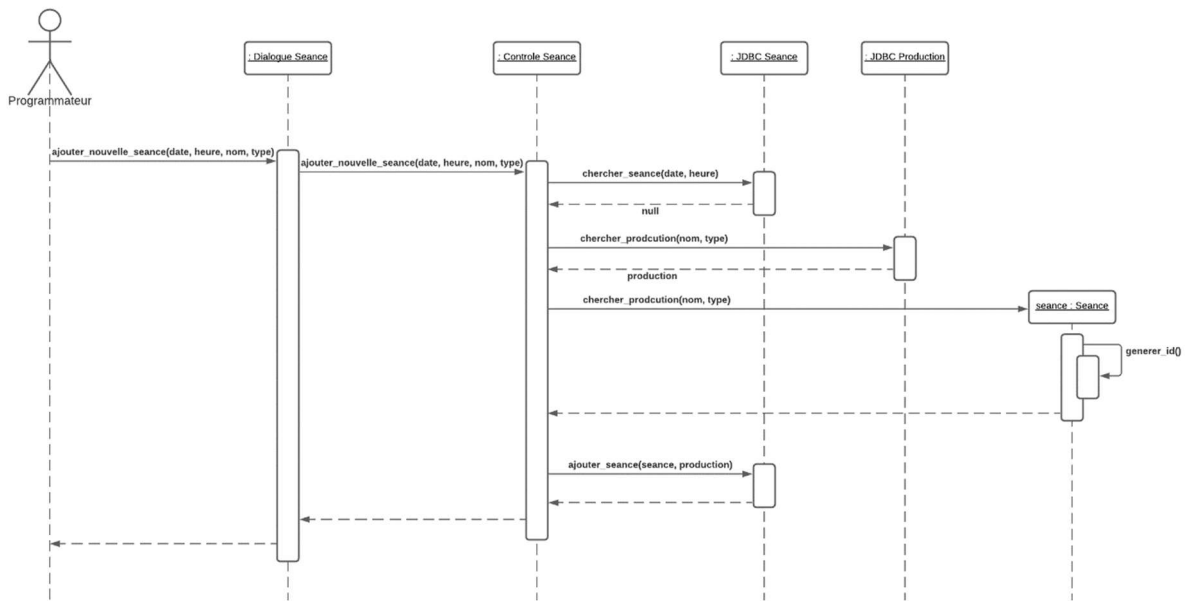


## Ajouter une production

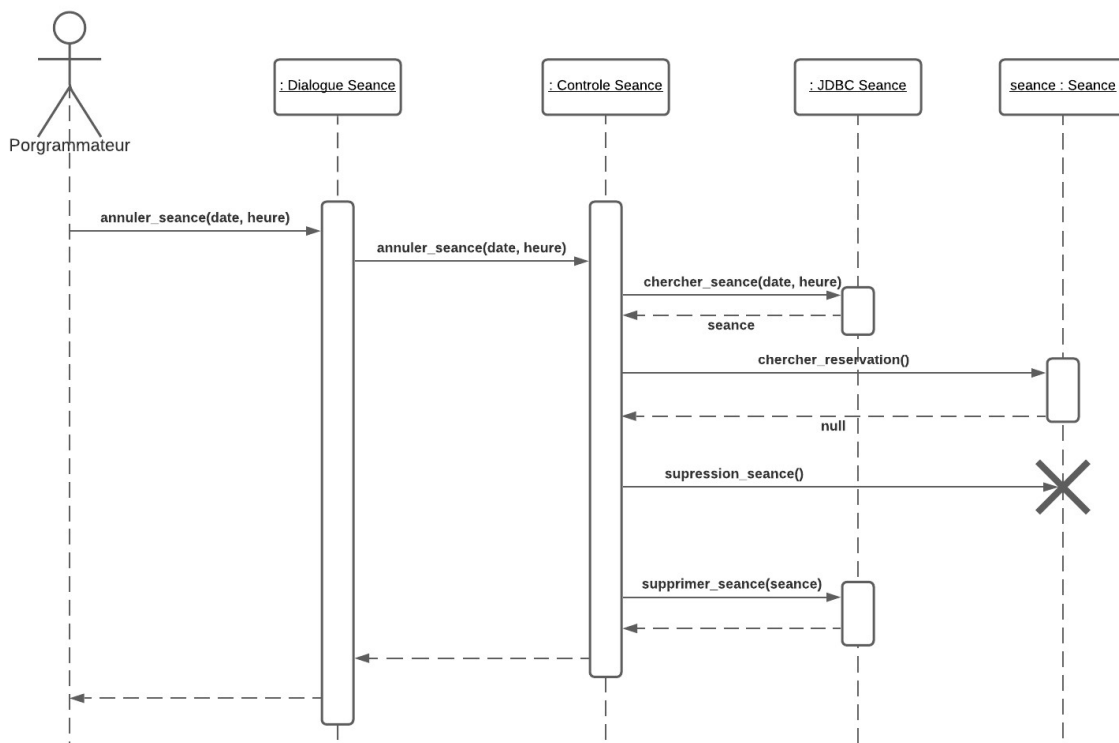




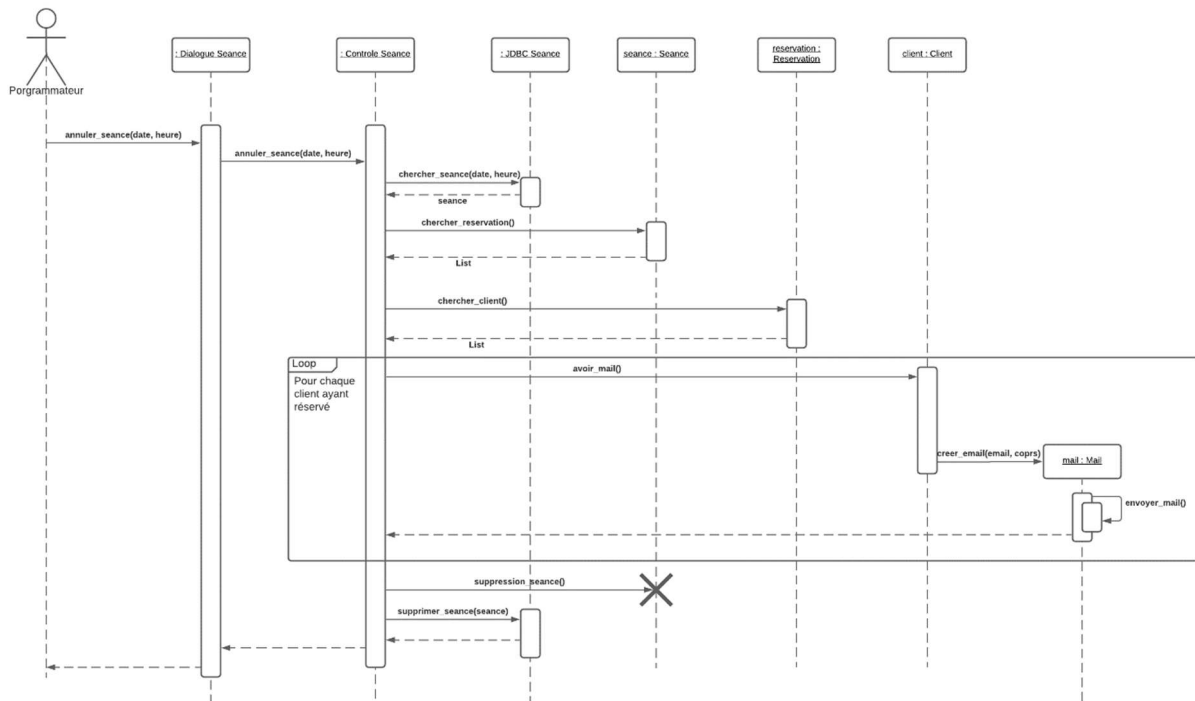
# Ajouter une séance



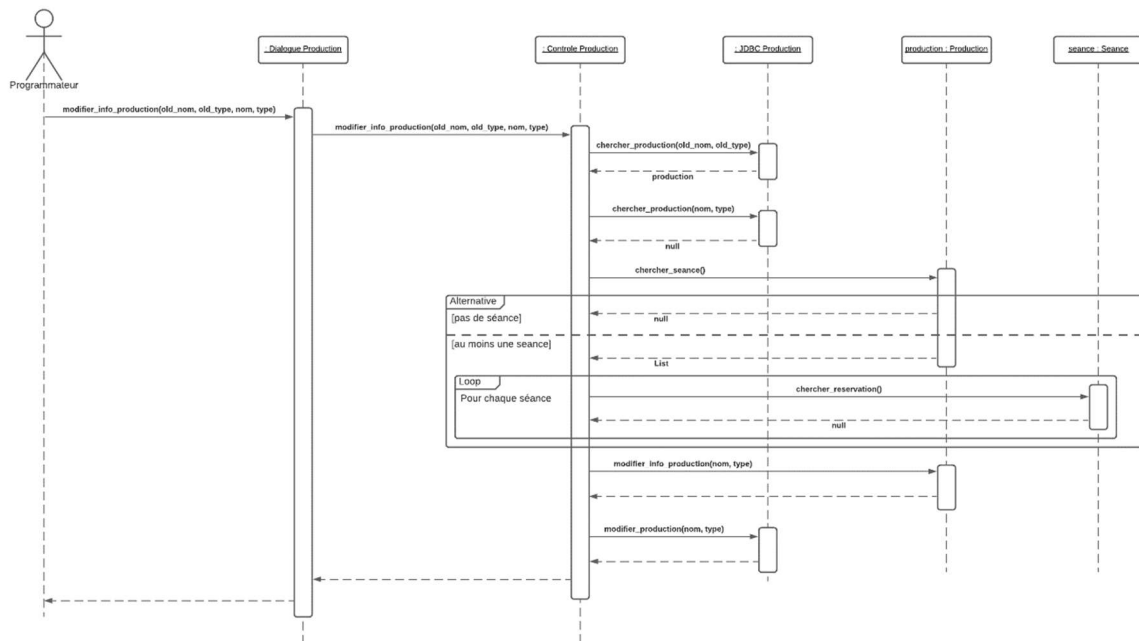
# Annuler une séance (principal)



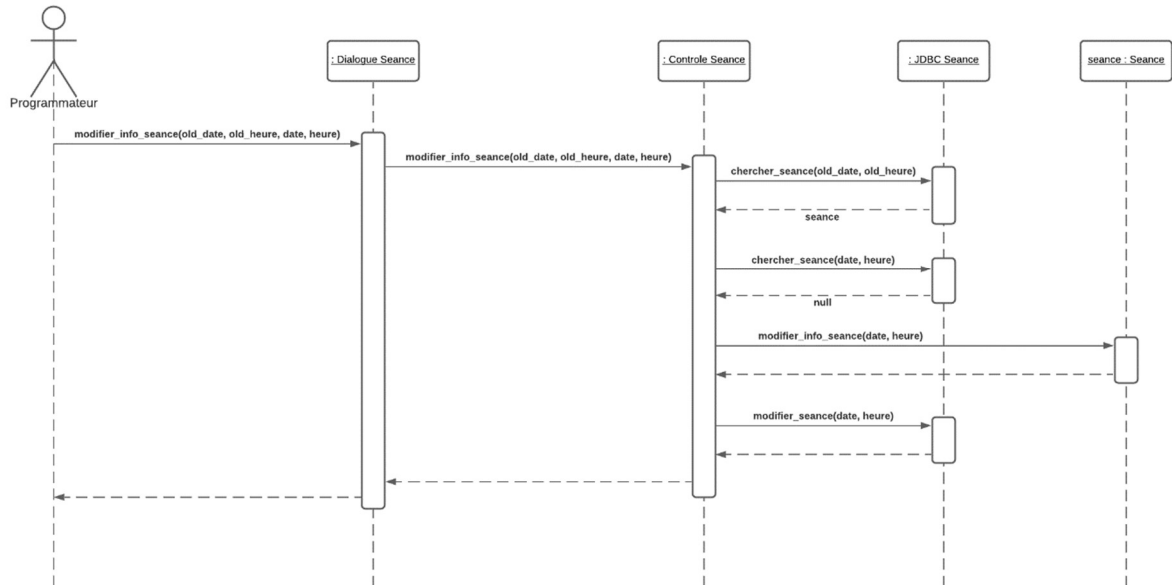
# Annuler une séance (alternatif)



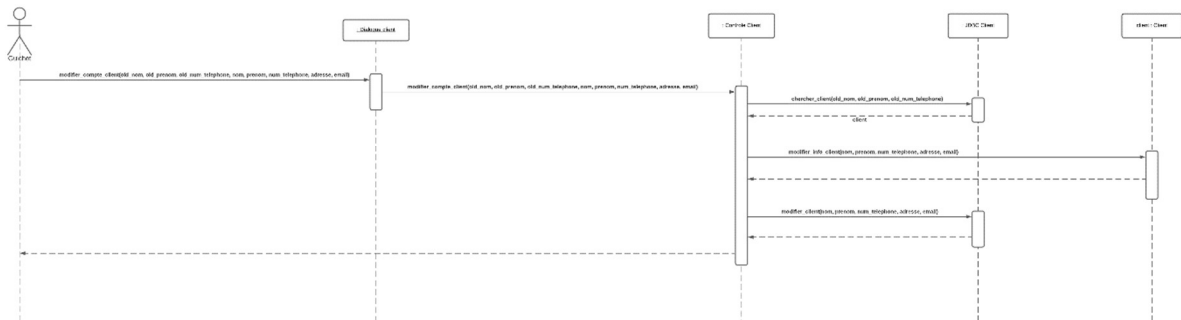
# Modifier une production



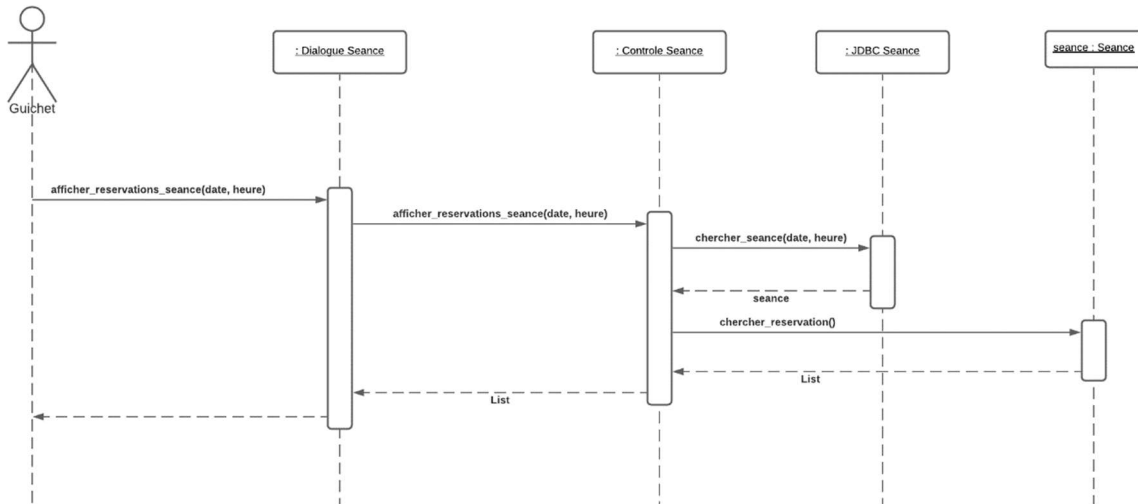
# Modifier une séance



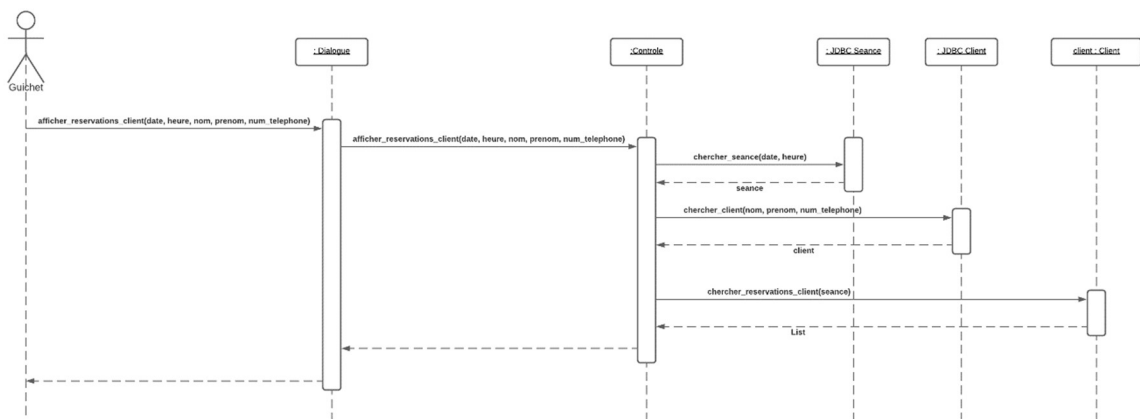
# Modifier un compte client



# Rapports des sièges réservés pour une séance



# Rapports des sièges réservés pour une séance pour client



# Réserver

